Investigating the Impact of Entry Qualifications on Student Performance in Computing Programmes at Undergraduate Level

SARAH ANNE DRUMMOND

Investigating the Impact of Entry Qualifications on Student Performance in Computing Programmes at Undergraduate Level

PhD Thesis

Sarah A Drummond

Department of Computer Science
Durham University

September 2009

Abstract

Context: This thesis investigates the impact of prior A-level study on students taking degree programmes within the Computing discipline. The focus of this work investigates opportunities to providing more-personalised learning which is based on students' existing knowledge, for example, by providing additional learning support to those students who had studied a particular topic at A-level. Although other studies have been carried out in this area, these studies have typically focused on outcomes across multiple programmes. Due to the variation of content taught, the researchers carrying out these prior studies have been unable to draw conclusions at the level of specific assignments.

Aim: The aim of this work is to investigate the impact of A-level subject selection on the performance of those studying Computing programmes at Durham University.

Method: This thesis is a detailed study, tracking Durham students, from entry until the completion of year two, over a particular three year period. This three year period of study was selected as, during these three years, Durham's entry qualifications and course content remained largely unchanged. Hence, the unintended impact of entry qualification and content change were not factors that needed to be taken into consideration.

A statistical analysis framework has been developed to investigate the impact which choosing specific A-levels has on student performance. Particularly, this work considers the impact on student performance, in course work (to the level of specific assignments) and examinations, of Maths, Computing, ICT, and Physics A-levels. The research compares the outcomes for students who have these qualifications against those who have not. Specific combinations of these A-levels are also considered.

Results: The results highlight some benefits in year one for students studying specific qualifications: largely Maths. However, the most significant result of this work is that, at the end of year two, any differences are insignificant. Therefore, while students with specific A-levels may gain benefits initially, at the point these student enter the final year of their programme, these differences no longer impact of their ability to study. The curriculum within Durham, therefore, already appears to address the needs of students, specifically by covering knowledge, or promoting individual study, of all topics necessary for successful progression. This research has, thereby, revalidated and added to the current body of knowledge in this research area.

While this work has identified students without specific A-levels are not adversely affected, what it points to is that some students with A-levels, for instance, in Computing, perceive their early University education to repeat much of their A-level work. So, although this study was not able to recommend personalisation of learning in support of those who have not studied specific A-levels, this work does highlight that, perhaps, personalised learning for those who have taken specific A-level may be necessary.

Conclusions: The outcomes of this research have clear and important consequences for Higher Education Admission Policies for the Computing discipline. As an outcome, it would seem that the requirements placed by many institutions on entrants to have specific A-level is unnecessarily restrictive and may be preventing many students entering a discipline in which they would, otherwise, have been successful.

Acknowledgements

My return to education came later in life than for most students and there have been many people from the past and in the present who have helped me to get to this academic point in my life. From my friend Tom Taylor as we journeyed together through the early days as fellow students to Professor Cornelia Boldyreff, who gave me my first job in academia, encouraged me to write my MSc and became a good friend.

In more recent years and when embarking upon this PhD I required and received the support and encouragement of many people. I would firstly like to thank my supervisor Professor Liz Burd for her support and guidance and whose supervision has been exactly right for me. Secondly I can't overstate how grateful I am to my friend and colleague Brendan Hodgson who has been a calming influence, source of common sense and a fantastic proof reader – his command of the English language and its constructs never ceased to amaze me. I will always be indebted to him for his help and the kindness and concern he has shown me.

Finally my family and friends without whom I would never be what I am today. Harry, my husband who has been a constant source of support and love and who never faltered in the belief that I would finish this work, even though I did on many occasions. My children Ellen and Harry who got used to a 'missing' mother but were proud of what I was trying to achieve and finally to my sister Julia Scott, who provided encouragement and much needed respite when things got tough.

I sincerely hope that all these people who have shared part of my life realise how they have helped in the completion of this thesis. Love to you all and thank you.

Copyright

Declaration

No part of the material provided has previously been submitted by the author for a higher degree in Durham University or in any other University. All the work presented here is the sole work of the author and no-one else.

CONTENT

TABLES

TABLE OF FIGURES

Chapter 1. Introduction

1.1. Background

It has been widely believed among academics for a number of years that A-levels are poor indicators of eventual university performance even though, in the UK (excluding Scotland), these remain the primary recruiting criterion for the vast majority of 18-year-old students (Sear 1983; Collins, White et al. 1995; Bekhradnia and Thompson 2002; Boyle, Carter et al. 2002; McManus, Powis et al. 2005). Computer Science is one of the few sciences that do not necessarily specify an entry subject in their particular discipline and, therefore, Computer Science students enter university with a diverse set of subjects which results in each cohort not starting on a "level playing field". This situation is rarely found in other disciplines, such as Mathematics, where the pre-condition for entry may well be high achievement in Maths and, frequently, in Further Maths as well. Certainly the A-level subjects Computing or Information and Communication Technologies (ICT) are not stated as compulsory for any Computer Science programmes and yet they are, at least to an outsider, our discipline's subjects.

At Computing education events, and more recently a British Computer Society workshop (February 12th 2009), a question often asked is which subjects studied at school best serve students when they embark on further studies in Computer Science at university. Whilst this question is often debated, there is never a conclusive outcome. One reason for this could be that it is often the case that each institution places different emphases on the syllabus and, hence, require students to have particular prior knowledge, which is demonstrated through their qualifications. Many Computing departments believe that Maths is the central skill and, therefore, require students to have a strong mathematical background. This may lead to a requirement on the students of having an A or B grade at A-level Maths in order to be allowed to undertake some Computer Science degree programmes which have strong theoretical emphases in their syllabus.

Certainly in Computer Science at Durham, Maths is currently seen as desirable but not as an essential qualification in order to read Computer Science but, as certain aspects of the syllabus are highly theoretical, it had been noted anecdotally that students without A-level Maths found these theoretical parts more difficult than those who did have Maths[1].

The discipline of Computer Science has strongly held but as yet unsubstantiated ideas of the effect which different pre-entry qualifications, specifically A-levels, have on student achievement in year one and year two of their study. The process of discovering whether these ideas contain any truth was the focus of this thesis with evidence being provided to establish, at least for Durham Computer Science students, what A-level subjects, if any, best prepare them. The data analysed is from three consecutive Durham Computer Science cohorts: 2004, 2005 and 2006. The course content in academic year 2003/04 underwent significant changes and, therefore, the choice of cohorts has been determined by these cohorts being exposed to a more stable post-2003/04 syllabus.

The use of consecutive cohorts, all of whom have been exposed to the same curriculum and syllabus, has the advantage that the results are more convincing and it is less likely that results are peculiar to a single cohort. Multiple cohorts allow for the replication of statistical tests and this, in turn, provides the opportunity of uncovering a significant result in one cohort, with the aim of replicating this or similar findings (literal replication) in further cohorts. Even with the cohorts being likely to differ to some extent, e.g. their students' motivation, preferred learning style, prior learning experiences, qualification etc., it is under these varied circumstances that it is hoped to arrive at a common conclusion for all cohorts. Whilst these other factors do contribute to a student's performance, they are outside the scope of this thesis; many of these factors are subjective and, therefore, difficult to measure.

[1] From October 2009 entry for CS at Durham requires A-level mathematics or equivalent.

1.2. Research objectives and study approach

This section introduces the main objectives of this research which are to determine what prior knowledge in the form of A-level qualifications is the most useful for Computer Science students before they arrive at university. Previous research in this area has looked at the relationship between entry qualification and performance at university, but often this is the relationship between entry subjects and exit degree classification or entry subjects and grades and performance in year one (Campbell and George 1984; Boyle, Carter et al. 2002; Alexander, Martyn et al. 2003). The research in this thesis places great value on the previous work and further extends it by, firstly, using consecutive cohorts of students studying the same syllabus at the same institution and, secondly, by investigating if there is any correlation between, on the one hand, exam, coursework and overall performance in year one and year two at university and, on the other hand, choice of A-level subjects. Particular emphasis is placed on the Computing and ICT A-level subjects to determine if any knowledge acquired through studying these subjects provides subject-specific knowledge which supports these students in their studies in year one. Further to this, it is of interest to determine if there is any difference between the school exam boards which offer ICT and Computing A-levels and whether this has any impact on the performance of these students in certain areas of their studies in Computer Science at university.

The approach taken for this work involved the analysis of the three cohorts of student data (A-level entry qualifications and marks achieved at university in year one and year two of study). This analysis is based on a number of 'Investigations' which are used to answer the research questions described in the following section and the statistical steps necessary to generate the results. To do this an analysis framework has been developed (section 4.7) which outlines the procedural steps taken to produce the results for each investigation.

1.3. Criteria for success

The success of this work can be measured, first, by evaluating the results which have been produced through statistical analysis and, secondly, by the analysis of a student questionnaire. The results of this work, which are based on 'Local' data, will be compared to previous studies which have been classified as 'International' (Alexander, Martyn et al. 2003), and 'National' (Boyle, Carter et al. 2002). It is through this analysis that first, the specific research questions for the 'Local' study will be answered and secondly, conclusions drawn from the three studies. The aim of this research is to contribute this knowledge to the higher education community and, more specifically, identity implications for Admissions and for Learning and Teaching Policies for Computer Science.

The quantitative and qualitative analysis of the Durham data has been divided into five distinct areas, with each area having a corresponding research question. The areas for analysis are:

1. Analysis of cohort;
2. Subject analysis;
    a. Single subject and combination subjects
3. Coursework categories;
4. Computing and ICT A-level syllabuses;
    a. Syllabus overlap achievement and exam board comparison
5. Students' perception of Computing and ICT A-levels

The research questions within these areas are as follows.

1. *Does student performance in exams and coursework and in their end-of-year performance differ between cohorts?*

2. *Does having a particular A-level (e.g. Computing) result in an improved performance in year one and year two compared with not having that A-level? The subjects of interest are:*

   i. Math
   ii. Computing
   iii. ICT
   iv. Physics

   a. *Does having a combination of A-level subjects, e.g. Maths and Computing, result in an improved performance compared with having neither subject? Combinations of interest are:*

      i. Math and Computing
      ii. Math and Physics
      iii. Physics and Computing

3. *Does having a particular A-level subject result in a better performance for certain types of coursework compared with not having that A-level?*

4. *Having identified an overlap of topics between the year one Computer Science syllabus and A-level Computing and ICT, do the students with these subjects have an advantage over non-Computing or non-ICT students in specific module assignments because of specific prior knowledge?*

   a. *Does it make any difference for students which ICT or Computing A-level exam board they used? Does one particular exam board better prepare these students than another exam board?*

5. *Research question 4 identified if there were significant differences in coursework assignments between students who do and do not have Computing/ICT A-level. Do these findings bear any relation to the ICT and Computing students' perceptions about their year one studies, in respect of what these A-level subjects provided them with?*

1.4. Structure of the thesis

This thesis contains seven chapters with this being the first.

Chapter 2 – Literature Review: provides a discussion of the literature that surrounds our understanding of how we learn. The problems of a student's acquisition of knowledge are discussed with particular reference to Computer Science education.

Chapter 3 – Computer Science Education in the UK: discusses Computer Science education and the philosophy behind it. Following this, an overview of the upper secondary education system in the UK is provided with emphasis on the A-level Computing and ICT qualification. A comparison of the syllabuses of the exam boards offering these subjects is presented.

Chapter 4 – Method: this chapter describes the sources of evidence and methods used to produce the results which are presented in Chapter 5. The statistical methods that have been used and an analysis framework which provides a step-by-step approach for this analysis are described. The set of tests (Investigations) and the questionnaire data that have been used to answer the research questions are described.

Chapter 5 – Results: the first part of this chapter presents the results generated from the statistical analysis for the 'Investigations'. Lastly, qualitative and quantitative results of the student questionnaire are presented.

Chapter 6 – Discussion: presents a discussion on the finding of this work. Comparisons between this 'Local' study and the 'International' and 'National' studies are identified and conclusions are drawn from the findings. The implications of the findings for university Admission and Learning and Teaching are discussed.

Chapter 7 – Conclusions: this concludes the research and presents a reflective outline of the contribution of the research. The success of the thesis is measured by providing answers to the research questions asked in the "Criteria for success" above. The recognised limitations of this research are also described with potential further work suggested.

Chapter 2. Literature Survey

This chapter describes the personalisation of learning and discusses the more predominant learning theories and learning styles, each of which have associated measurement tools.

An understanding of, and the effective application of learning theories (described in the following sections) in higher education can have an impact on student performance. In particular Constructivism which is seen as the dominant theory of learning (Ben-Ari 1998) is applicable to Computer Science education with a good exemplar of constructivist learning environment being Problem Based Learning (section 2.4.1). Furthermore, research into learning styles has been made to provide a better understanding of the ways students choose to learn and which formed a basis for the investigations undertaken. To teach students it is important for the student and teacher to recognise how the student learns and how best to support this learning.

Personalising the learning situation for each learner would be the ideal but can be problematic – simply because of the wide variety of learning styles found in a classroom. There is a large amount of literature describing learning theories (Skinner 1953; Martin 1999; Piaget 2000; Cassidy 2004; Fosnot 2005) and the adoption by learners, of a particular style of learning (Sadler-Smith 1997; Adey, Fairbrother et al. 1999) and how this can have an impact on performance and on the achievement of learning outcomes. Several methods, described in section 2.3, have been developed to identify an individual's learning style, many of which have been met with criticism.

Software tools are discussed as part of the 'scaffolding' that can be provided to the learners to support their different levels of understanding. Determining the level and progressing this understanding often requires the learner to cross a 'threshold' when they can move forward with a transformed way of thinking, or understanding, of a particular topic. This ‘threshold’ concept is discussed in general and then specifically applied to Computer Science education.

2.1. Personalised Learning

Personalised learning can be seen as an approach in educational policy and practice whereby every learner matters, with the ultimate intention of equalising the learning opportunities in terms of learning skills and motivation to learn (Jarvela 2006). In his paper for the Personalised Education Conference in 2004, the then UK School Standards Minister, David Miliband, presented his vision and policy agenda for personalisation of learning – explaining the rationale for personal learning being to “focus learning and teaching on the aptitudes and interests of pupils so as to tailor education to ensure that every pupil achieves the highest standard”(Miliband 2006).

However, the reality of this is problematic because of the variety of learning styles, motivation and needs of students of all ages who enter education. Whilst students do not “recite lessons in chorus as they did a century ago, we are a long way from true personalised learning in education” (Paludan 2006). Our current educational system includes ‘measurement’ and this sets limits, such as formal examinations, on the extent to which personalisation can be made available. Whilst personalised learning is clearly advantageous to learners, the reality is that it has not been widely introduced into all education. However, it should be noted that personalised learning has been used in one-to-one tutorial sessions for many years and used extensively for Special Needs learners in specialist schools where the individual’s needs have to be specifically catered for.

The ideal of tailoring education to an individual is to involve the learner in accepting responsibility for their own learning and the attainment of learning outcomes. In doing so, it is more likely that a learner may feel that the system takes their individuality into account and it can promote a better understanding of his or her learning needs (Paludan 2006). However, this does not mean mandating an educational system where every learner sits in isolation with their own personalised learning strategy.

Being responsible for one's own learning is a core principle of Higher Education. This is unlike primary and secondary education, where students generally lack the maturity for such responsibility. However Leadbetter (Leadbetter 2006) states that personalised learning would offer choices in how their education "could unfold, branching out in many ways and styles" but with the core still being the basic curriculum.

2.1.1. e-Learning and Personalised Learning

The rapid evolution of ICT provides tools which facilitate the implementation of a new paradigm in education – e-Learning (Sampson, Karagiannidis et al. 2002) – which helps to provide personalisation with, for example, on-line training programmes. These will be customised to an individual's needs and be based on some analysis of the learner's objectives and current skills or knowledge and learning style preferences. In this context, personalised learning becomes more attractive than instruction and is not restricted by time or place.

With the rapid evolution of technology, tools which facilitate the implementation of e-Learning (Sampson, Karagiannidis et al. 2002) have been developed to help provide this personalisation and move away from the more traditional mode of instruction, of one-to-many lecturing, which cannot fully accommodate the different learning styles of diverse learners. The introduction of personal learning environments help learners set their own learning goals and manages both the content and context of their learning. A number of projects (Manouselis and Sampson 2002; JISC 2006) have been funded to harness the potential of using various technologies for e-Learning to support individuals, not just for immediate learning but also to equip them for lifelong learning.

The concept of personalised learning builds mainly on the cognitive and constructivist theories of learning which are described in the following sections.

Both cognitive and constructivist theories include learners being actively involved in their learning.

2.2. Learning theories

There are many theories of how we learn and these have been developed over a number of decades (Pavlov 1941; Skinner 1953; Vygotsky 1978; Piaget 2000) each contributing to an understanding of learning.

Learning theories address how people learn and they tend to fall into one of several paradigms, including behaviourism, cognitivism and constructivism. These are now described in the sections below.

2.2.1. Behaviourism

Behavioural theories which dominated the psychology of learning in the first half of the 20th century viewed learning as the acquisition of a new behaviour or a change in behaviour in the learner. Behaviourism assumes a learner forms associations between environmental stimuli and their responses, and where behaviour can be based not only on the consequences of the stimulus but also on past behaviour through positive or negative reinforcement[2](Skinner 1964). Reinforcement is the extent to which the individual was supported in the past in performing the same or similar behaviour (Schunk 2008).

Behaviourism began with the rise of experimental work in psychology and a move away from anecdotal evidence. Behaviourism focused on observable behaviour which was easier to quantify and from which to collect data. Whilst there are a number of proponents of behaviourism, perhaps the most widely known work and a forerunner to behaviourism, was Ivan Pavlov in 1890 (Pavlov 1941). Pavlov's classical conditioning (learning by association) involved dogs being taught through a

[2] Negative Reinforcement should not be seen as punishment but that it strengthens a behaviour because a negative condition is stopped or avoided as a consequence of the behaviour.

simple stimulus-response mechanism where the sound of a bell was associated with being given food. This is also known as Pavlovian conditioning. However, the founding of behaviourism can be attributed to John Watson, known as the "father of behaviourism" whose focus was on the external behaviour of people and their reaction to a given situation.

Criticism of Watson's work, and behaviourism in general, arose because of his dismissal of the study of consciousness which he saw as "neither definite nor a useable concept" (Bloomfield 1967) and being irrelevant in predicting the behaviours of humans and animals (Martin 1999).

Behaviourism follows the premise that learning requires a low degree of processing, e.g. rote memory, stimulus-response and reinforcement, but it places little importance on the vast differences between learners and between the different types of learning. This is where it differs from the cognitive school of thought.

2.2.2. Cognitivism

Cognitivism is about how we gain knowledge and use that knowledge to guide decisions. This theory stresses the need for the acquisition of knowledge and the practising of skills, the formation of mental structures (memory networks) and the processing of this information to promote learning (Schunk 2008). Psychologist Jean Piaget's assertion is that a learner's interaction with their peers, and the learning environment supporting the activities of the learner, are important in increasing cognitive development (Piaget 2000).

In the process of learning, actions and active participation, not just responding to environmental stimuli, result in thinking. Vygotsky (Vygotsky 1978) adds to this with the view that culture – including the family environment – is a prime determinant of an individual's development. Downing's (Downing, Ho et al. 2007) work on the impact of social and cultural factors on the development of meta-cognition in first year university students found significant differences between students living in their home environment and those who moved away from their family and, in some cases, cultures. Living away from home was found to provide a new learning environment which fostered the development of meta-cognition and provided students with more opportunity to "become successful problem-solvers and lifelong learners".

Whilst the term meta-cognition (thinking about thinking) is relatively young, the concept is centuries old. It is about knowing how to reflect and analyse thought and put what has been learned into practice. To be good problem-solvers, learners need to understand how their mind functions e.g. in cognitive tasks such as remembering things learnt earlier that might help with the current task or problem. Meta-cognition focuses on the process of problem-solving while cognition focuses on solving the problem (Downing, Ho et al. 2007).

2.2.3. Constructivism

Constructivism is a theory about knowledge and learning and it describes both what "knowing" is and how one "comes to know" (Fosnot 2005). Constructivism sees learning as a process in which the learner actively constructs or builds new ideas or concepts from their current and past experiences; it links new knowledge to prior knowledge and incorporates the new experience into an already existing framework without changing that framework. In some respects, constructivist theories are similar to cognitive belief in that learners take the information from the environment and combine it with their present knowledge.

Formalisation of the theory of constructivism is generally attributed to a number of people, including Vygotsky and Piaget, and has increasingly been applied to learning and teaching. Teachers focus on making connections between facts to foster new understandings in learners and to encourage students to analyze, interpret and predict information.

The constructivism approach to learning deals with ill-defined problems which can be dealt with through reflection in action. An example of this is the problem-based learning approach (section 2.4.1) where knowledge is constructed, not transmitted, and prior knowledge impacts the learning process.

2.2.4. Summary of learning theories

There are differences and similarities between these learning theories. For example, within behaviourism there is a difference between Pavlov’s early work on classical conditioning, which is a simple form of learning using stimulus-response mechanism, and Skinner’s work, which is where new behaviour can be based not only on the consequences of the stimulus but also on past behaviour through reinforcement (Atherton 2005). This aspect of Skinner’s work can be likened to a facet of constructivism which promotes the learner’s use of prior knowledge.

In an educational setting, behaviourism implies that the teacher dominates the learning. This is not only by providing more instruction and immediate corrective feedback but also by lesson planning, ensuring an orderly classroom, providing clear learning objectives for progression from simple to more complex concepts, the use of practice and repetition thus strengthening learner motivation and ensuring learners are aware of the significance of the subject matter.

In constructivism, the emphasis is on hands-on learning, with the teacher helping to make connections between the facts and fostering a new understanding in the student. It is the learner rather than the teacher who is at the centre in the constructivist approach.

2.3. Learning Styles

The ways in which we learn are known as ‘learning styles’ each contributing to the way we define how individuals learn. Individuals perceive and process information in very different ways and a learning style is the underlying preference an individual has for a particular type of learning. It is the way in which "each person absorbs and retains information and/or skills regardless of how that process is described" (Dunn 1984).

Learning style(s) can be dramatically different for each person. Some people prefer visual learning – looking at images, mind maps etc. (imagers), whilst some prefer auditory learning (verbalisers) (Riding and Rayner 1998). Learners can, however, be multi-modal and have more than one strong learning preference. If a learner understands how they learn best, it can increase the learning motivation. Learners should be encouraged to diversify their style preferences (Friedman and Alley 1984), where one learning style is neither preferable nor inferior to another, but is simply different with different characteristic strengths and weaknesses.

We learn better when someone is teaching us in our most comfortable style, even though we are capable of doing it in other ways. It is, however, important that learners are aware of, and develop, other learning styles. Having an understanding of where your own strengths and weaknesses lie, and the learning strategies available which can help in dealing with these, can also help foster independent learning. Styles are part of the individual's make-up and strategies can be learned and called upon by the learner as needed.

An individual's learning style can be complex and not easily reduced to simple categories. Many learning style instruments or tools have been developed (Honey and Mumford 1992; Tait, Entwistle et al. 1998; Kolb and Kolb 2005) to determine an individual's actual learning style. However, an individual's learning style can be complex and not easily reduced into simple categories, a deficiency for which many of these tools have been criticised (Coffield, Moseley et al. 2004).

2.3.1. Models of learning styles and learning style tools

Learning styles have been the focus of much research and practitioner-based studies which has resulted in a diversity of theories, definitions, models, measures and interpretations (Cassidy 2004). The outcome of this large amount of research and empirical investigation is ambiguity and debate in selecting one of the many tools/instruments developed. It is also widely agreed that many of the tools/instruments are derivations and adaptations of other models (Coffield, Moseley et al. 2004).

The terms ‘tools’ and ‘instruments’ are often used but in fact these are, in the main, questionnaires, surveys or inventories to measure the learning preference of learners. However, this in itself is problematic as the measurements are derived from subjective judgements which learners make about themselves and as such the treatment of scores is “on shaky and insecure foundations” (Coffield, Moseley et al. 2004a).

Learning style models can be categorised into four groupings (Riding and Rayner 1998). Those that are:

1. focused on the learning process (Honey and Mumford 1992; Kolb, Boyatziz et al. 2000),
2. grounded in orientation to study (Biggs 1987; Tait, Entwistle et al. 1998),
3. based on instructional preference (Dunn 1984),
4. based on cognitive skills development (Gregorc in (Coffield, Moseley et al. 2004) pp13, (Allinson and Hayes 1996)

Six examples of the more prominent learning style models which are categorised above are:

1. Kolb’s Learning Style Inventory
2. Honey and Mumford’s Learning Style Questionnaire
3. Tait and Entwistle’s Approaches and Study Skills Inventory for Students
4. Dunn and Dunn Learning Style Inventory
5. Gregorc’s Style Delineator
6. Allison and Hayes Cognitive Style Index

Outlines of each of these learning style models and their associated tools are described in the following sections.

2.3.2. Kolb's Learning Style Inventory

The Learning Style Inventory (LSI) was first developed in 1969 for adults in management training to help an individuals understanding of the process of experiential learning. Since then, another four versions have been developed with the latest adaptation being in 2005 (Kolb and Kolb 2005).

Kolb suggests that individuals' learning styles can be identified by assessing their position on two axes using the LSI, with many people having learning styles which are in-between the extremes. The LSI classifies an individual's preferences along these axes:

1. Concrete experience (CE) or abstract concept (AC)
2. Active experiment (AE) or reflective observations (RO)

The first dimension is concerned with whether an individual is more comfortable with concrete or with abstract ideas. The second dimension relates to the extent to which an individual would rather think and reflect on something than get involved. Kolb's research on the LSI identified four basic learning styles associated with different approaches to learning. These styles and approaches are:

1. **Divergers**: good at brainstorming, interested in people, prefer to work in groups,
2. **Accommodators**: learn from hands on; rely more on others for information,
3. **Assimilators**: more interested in ideas and abstract concepts; like to think things through,
4. **Convergers**: can solve problems and made decisions; best at finding practical uses for ideas and theories (Lashley and Barron 2006).

Scores attained on the inventory are intended to be interpreted not as definitive but as a starting point for discovering how the learner best learns and to help in selecting the learning approaches that work best for them in different learning situations (Lashley and Barron 2006).

2.3.3. Honey and Mumford's Learning Style Questionnaire

The Honey and Mumford Learning Style Questionnaire (LSQ) is the most widely used approach to determining learning styles in the UK. The Kolb model was used as a basis for the development of the Honey and Mumford's LSQ (Sadler-Smith 1997).

Honey and Mumford (Honey and Mumford 1992) argue that people learn most usefully from experience. However, simply having experiences does not guarantee effective learning. The experience should be reviewed, conclusions drawn from the review and action taken to build upon the conclusions drawn. If all parts of the cycle are not followed then effective learning does not occur.

The Honey and Mumford LSQ identifies individual strengths in each of four learning styles which are:

1. **Activists**: learn through experience in concrete situations.
2. **Reflectors:** like to process information by deliberating over experiences and observe them from different perspectives.
3. **Theorists:** process information by assimilating it into coherent theories and models.
4. **Pragmatists:** learn by relating new information to practical situations and problems.

2.3.4. Tait and Entwistle's Approaches and Study Skills Inventory for Students

The Approaches and Study Skills Inventory for Students (ASSIST) tool has evolved over a period of 30 years from the first in 1981 (Approaches to Study Inventory ASI) and its subsequent revision in the mid-'90s called the Revised ASI (RASI) (Tait, Entwistle et al. 1998). From evaluations of ASI and RASI, ASSIST was derived to capture, quantitatively and qualitatively, students' approaches and perceptions of learning (Coffield, Moseley et al. 2004a). This inventory is one of the few that specifically address learning in higher education.

The RASI uses a Likert scale response format and identifies five approaches: (Sadler-Smith 1997; Cassidy 2004):

1. **Deep approach**: try to work out the meaning of the information themselves.
2. **Surface approach**: rely on a rote-learning of material, acceptance of ideas without necessarily understanding it.
3. **Strategic approach**: learner has clear goals related to their studies and ensures they have all the resources for success and are generally well organised.
4. **Lack of direction**: reflects subject's lack of clear academic and career direction and goals.
5. **Academic self-confidence**: scoring high on this, the learner perceives themselves as able, intelligent and able to cope with the intellectual and academic demands of their studies.

2.3.5. Dunn and Dunn Learning Style Inventory

The Dunn and Dunn Learning Style Inventory (LSI) model is based on students' preferences and learning outcomes and related is not just to intelligence, but to factors such as environment, opportunities presented etc. The Duns' model is often referred to as the "VAK" approach because it focuses on visual, auditory and kinaesthetic learning styles. Dunn and Dunn (Dunn, Dunn et al. 1975) have developed a number of tools including the Learning Style Questionnaire introduced in 1979 with the LSI introduced in 1992. The LSI is widely used within education on a world-wide basis.

Dunn and Dunn divide learning styles into five major strands called stimuli. These stimuli are:

1. **Environmental**: room, seating, light, heat
2. **Emotional**: motivation, persistence, need for structure, responsibility
3. **Sociological**: learning groups or alone, help and support available from parents and teachers
4. **Psychological**: time of day, mobility, preference for visual, auditory, kinaesthetic or tactile
5. **Physiological:** elements that influence how individuals learn.

2.3.6. Gregorc's Style Delineator

The Gregorc Style Delineator is a self-administered and self-analysis tool which involves completing the rank-ordering of a set of words that are the most and least descriptive of the individual (Cassidy 2004; Coffield, Moseley et al. 2004). Gregorc identifies four learning styles which consist of distinctive behaviours which indicate how a person learns from and adapts to his environment. These learning styles and behaviours are:

1. **Concrete sequential**: direct step-by-step, orderly, sensory based learning
2. **Concrete random**: trial and error, intuitive and independent approach to learning
3. **Abstract sequential**: analytical and logical approach, prefers verbal instruction
4. **Abstract random**: preference for holistic, visual, experiential learning style

2.3.7. Allison and Hayes Cognitive Style Index

The Cognitive Style Index (CSI) developed in 1995 (Allinson and Hayes 1996), is based on a three point rating scale in order to measure a single dimension with intuition, at one extreme, and analysis, at the other. 'Intuition' is seen as a characteristic of right-brain orientation with judgement based on feeling. 'Analysis' is characteristic of left-brain orientation with judgement being based on mental reasoning and focus on detail (Coffield, Moseley et al. 2004; Coffield, Moseley et al. 2004a). Completion of the questionnaire would provide a score indicative of either an intuitive or analytical nature. Once the CSI has determined how far individuals are intuitive or analytical in their cognitive style they consider whether or not it is possible to integrate these two styles into a 'whole brain' approach (Sadler-Smith 1997).

2.3.8. Can Learning styles change?

Learning styles are biologically-determined functions of individuals and can be considered stable over time (Riding and Rayner 1998; Cassidy 2004). Learning styles can be difficult to change but that is not to say that people cannot change or cannot learn to make something out of all the learning experiences they encounter. People can adapt their style – sometimes temporarily to suit a particular learning demand. Ideally, as individuals, it would be sensible to know our preferred learning style so we can understand why some things are easier to cope with and make adjustments.

If a student is told they have a particular learning style, they become familiar with the characteristics of this style and can assume that is the only way of learning. The student needs to become aware of the different styles which can be adopted and are appropriate to them in different situations. The problem a teacher faces is to accommodate the different styles present in a classroom to stop some students becoming bored by a particular approach which is not suitable for them. This type of situation could be served well by the use of e-Learning tools providing support through offering additional challenging material.

2.3.9. Criticisms of learning styles and tools

Educators in the field recognise the importance of understanding how individuals learn and any attempts to integrate learning styles into educational programmes should be made from an informed position (Cassidy 2004). However, the literature on learning styles is extensive and confusing. Researchers over the years have worked in isolation and developed their own instruments of assessment and given their own labels to the styles they were studying.

Learning-style theories have been criticized by many with Sadler-Smith, and Coffield and others discussing this in detail (Sadler-Smith 1997; Coffield, Moseley et al. 2004; Felder and Brent 2005; Kratzig and Arbuthnott 2006). Some psychologists and neuroscientists have questioned the scientific basis for these models and the theories which often rest on dubious theoretical grounds.

A systematic and critical review on learning styles used in post-16 learning was undertaken in 2004 for the Learning and Skills Research Centre (LSRC) (Coffield, Moseley et al. 2004) which looked at a large number of learning-style models and, from these, selected for detailed study thirteen of the most influential models of learning styles.

The outcome of the LSRC report was highly critical of the tools and instruments used to identify an individual's learning style. They examined the theory and terms used behind each model and the instrument/tool that was used to assess types of learning style defined by the model. They analysed the claims made by the authors' external studies and independent empirical evidence of the relationship between the 'learning style' identified by the instrument and students' actual learning. Coffield's team found that none of the most popular learning-style theories had been adequately validated through independent research, leading to the conclusion that the value of matching learners and their learning styles was all "highly questionable".

The LSRC report concluded that there is:

- Consistent psychometric failings in models and measures of cognitive learning styles,
- Replication between research,
- Commercial conflicts of interest which undermine reliability in reported research,
- No consensus or coherent theory.

These conclusions were generally supported by DEMOS (DEMOS 2005) whose conclusion included “the research evidence for these styles is highly variable ….various exponents were not by any means frank about the evidence of their work”. With many schools adopting these tools to determine the learning style of their students, this is a worrying situation (Revell 2005).

Coffield's review of learning styles in the context of further education provides a useful description of a wide range of learning-style models. However, Rayner (Rayner 2007) feels that it is difficult not to be affected by the “tone and nature of evaluation and perspective” presented by the review team. Rayner argues that learning styles are an important concept but rather than try and apply this to a one-size-fits-all solution what is required is “developing approaches to diversity and individual needs in the classroom”.

2.3.10. Summary of learning styles

When learning something new or difficult, there is a natural tendency to use a preferred learning style. If the material presented is not in the style preferred, having knowledge of other learning styles and strategies which can be adopted is very important in learning and studying. Individuals process information differently and, in order to make the learning more efficient, a learner needs to understand how they as individuals do this. Determining an individual’s learning style(s) has resulted in many tools being developed with variability in quality and, in many cases, weak independent empirical evidence to support the claims of these ‘tools’.

2.4. Enquiry-based learning

Enquiry[3]-based learning (CEEBL) is “a broad umbrella term used to describe approaches to learning that is driven by a process of enquiry” (Kahn and O'Rourke 2005). EBL is self-directed learning where the learner takes control of their learning process and takes an active role in the acquisition of relevant knowledge. EBL does, however, incorporate structures and forms of support to help and encourage learners to create and conduct their own enquiries through the teacher establishing a general theme, an issue or triggers (such as a picture, a quotation, a current event) and then facilitating the process (CEEBL 2005). The learners pursue their own lines of enquiry, draw on their existing knowledge and identify their learning needs in the subject area. They seek out relevant evidence and take responsibility for analysing and presenting this appropriately, either as part of a group or as an individual supported by others.

On an individual basis, an example of EBL would be final year undergraduate projects where individuals must plan, research and write up a project. However, EBL is fundamentally about collaboration with learners working together to pool their collective knowledge and understanding and acting together to create new knowledge for a particular purpose. This approach seems to be as much about formulating questions to understand the complexity of problems and their contexts, as it is about seeking answers and possible solutions.

EBL follows four basic stages which define self-directed learning (CEEBL 2005). Learners take more responsibility for:

1. Determining what they need to learn,

[3] The Oxford Concise Dictionary definition for Enquiry is that it is simply used as another term for Inquiry.

2. Identifying resources and how best to learn from them,
3. Using resources and reporting their learning and
4. Assessing their progress in learning.

2.4.1. Problem-based learning

Problem-based learning (PBL) is a type of EBL and it allows learners to learn actively and it also encourages team-working skills. PBL, under the EBL umbrella, helps deliver technical theory in a practical way, encouraging active participation where learners encounter real-world, complex problems which requires them to define their own learning needs within broad goals set by the teacher.

In PBL learners are not expected to acquire a predetermined series of right answers. Instead, they are expected to engage with the problem and decide what information they need to learn and what skills they need to gain in order to complete the task (Kenny 2005). Once these gaps are identified, the learners will undertake self-directed study, and learn to use the appropriate information resources (Jarvela 2006). This process will then help ensure its recall and application to future problems.

Students can enter higher education conditioned by their previous educational experience, often in a traditional instructivist way, to be passive recipients of what they are taught. Allowing students to take control and responsibility for their learning can significantly increase their ability to learn (Savery and Duffy 1995). Importantly, the PBL learning process involves trying to use any prior knowledge or experience the learners may already have to help solve these real-world problems. This is important as it allows them to appreciate and use what they may already know.

Savery's definition of PBL is "an instructional (and curricular) learner-centered *(Shepherd)* approach that empowers learners to conduct research, integrate theory and practice, and apply knowledge and skills to develop a viable solution to a defined problem" (Savery 2006). This learner-centred educational approach makes the learner begin to become less dependent on the teacher. As learners become more proficient in the PBL learning process, the teacher becomes less active. PBL contrasts with the traditional subject-based approach where learners are first taught a body of knowledge and then may have the opportunity to apply it when they are presented with sample problems.

Whilst the notion of learning through solving, or managing, problems is not new, the emergence of PBL began in the mid-sixties at McMaster University in the medical education community. Barrows (Barrows and Tamblyn 1980) discovered, through his research into medical education, that medical students for the most part didn't think – they gathered data ritualistically and then tried to make sense of this afterwards – and that some students came up with a diagnosis based on some symptom or sign but never considered other possible alternatives. To overcome this, Barrows set out to develop a PBL curriculum build around small-group, student-centred learning. PBL has now also been implemented at different educational levels from primary schools to university.

2.4.2. Issues in PBL

Whilst problem-solving is an important part of the student learning experience, there is some confusion about the difference between problem-solving and PBL (Savin-Baden 2000; Butler, Inman et al. 2005). Problem-solving learning is an approach which teaching staff have used for many years. The focus in problem-solving learning is that learners are given, for example, an article to read and then they are given a set of questions based on this information. Learners are expected to find the solutions to these questions which are rooted in the information supplied and then to bring them to the seminar as a focus for discussion. The problem scenarios are bounded within a specific subject or disciplinary area (Marton and Saljo 1976). .

Butler (Butler, Inman et al. 2005) states that an issue with PBL is the word 'problem' because PBL means quite different things to different people. Some problems have one solution whilst others may have a number of solutions or none. The crucial point here is that it is possible to find a solution, but not necessarily an understanding, whereas PBL leads to an understanding of the problem but not necessarily to the one, definitive solution (Savin-Baden 2000).

Resistance to PBL from students can manifest itself in the form that, with workload pressures, students want simply to be taught/given information as they believe there is inadequate time in which adequately to explore issues (Greening, Kay et al. 1997; Ryan 1997; Albanese 2000)

2.4.3. Assessment in PBL

Curricula and assessments are often tightly specified and regulations can constrain assessment because they make it difficult to use the type of assessment that would be aligned to the learning intentions for PBL. There are a number of issues which currently stand in the way of effective and efficient adoption of PBL, one of which is student assessment. Assessments in conventional courses are designed to test recall of information and the ability to apply it. It is difficult to apply these forms of conventional assessment to PBL curricula since they are based on different conceptions of knowledge. Exponents of PBL state that students will acquire the information as and when they need it, which is the case, but their capacity to do this outside a controlled curriculum tends to be based on motivation and experience (Drinan 1997).

Longitudinal studies which compared the academic performance of students using PBL and those following the standard curriculum concluded that students using PBL performed at least as well as the other cohort but that no significant effects on knowledge levels were found for PBL schools versus non-PBL schools (Greening, Kay et al. 1997; Distlehorst and Robbs 1998; Prince, van Mameren et al. 2003). However, others (Barrows and Tamblyn 1980; Albanese 2000; Norman and Schmidt 2000) argue that this PBL approach sets out the conditions needed for effective and deep learning of disciplinary knowledge and problem solving. Further studies by Martensens and Eisenstaedt in (Norman and Schmidt 1991), and Strobel (Strobel and van-Barneveld 2008) on short- and long-term recall whilst learning from PBL, concluded that the PBL approach may initially reduce levels of learning but it leads to increased retention over a period of time and also enhances self-directed learning in comparison with a conventional curriculum.

2.4.4. Summary of Enquiry Based Learning

The main characteristic that distinguishes PBL from EBL is that, in PBL, the problem is presented first at the start of the learning process before the curriculum has begun. In addition, students take responsibility for defining their own learning issues, what they need to research and how to apply it to the problem (Barrett, Labhrainn et al. 2005). In contrast, in EBL it is often the case that learners are guided by some initial instruction, or material with pre-set learning outcomes, which the student studies independently and then, through group discussions, devises a new set of questions to examine the problem in more detail, before final discussions and consolidation of the research (CEEBL 2005).

PBL starts with problems rather than an explanation of the necessary knowledge. It moves learners towards acquiring knowledge and skills through a sequence of problems presented in context, together with learning materials and support from the teacher. PBL takes account of how students learn, it is more effective and this active learning expands their knowledge of the discipline. It is impossible to include all knowledge and so students need to be able to learn quickly, effectively and independently. Important components of PBL are that learning is cumulative, i.e. a topic is not studied in-depth but is ongoing with different levels of sophistication. Subjects are not presented separately but are available for study as they relate to the problem.

Monitoring the knowledge students acquire can be problematic which has led to research being conducted to improve the delivery and effectiveness of PBL by using intervention. This intervention has been referred to as ‘scaffolding’ (see Section 2.5). Scaffolding is a process in which students are given support until they can apply new skills and strategies independently (Rosenshine and Meister 1992). Scaffolding should be seen as providing a framework for a teacher to access different levels of a student's understanding.

2.5. Scaffolded Learning

Scaffolding is the process by which a teacher or "expert" assists a learner to complete a complex task that would be beyond their reach without support and guidance. This concept is based on work of Lev Vygotsky (Vygotsky 1978), an influential figure in developmental psychology who proposed that, with an adult's assistance, children could accomplish tasks that they ordinarily could not perform independently.

The actual term, "scaffolding", was first used in the mid-seventies in a discussion on the role of teaching in the problem-solving process (Wood, Bruner et al. 1976) and was expressed in terms of a particular task; that of a teacher teaching young children to build a 3D structure that required a degree of skill that was initially beyond the child. Wood saw the process as similar to problem-solving in which mastery of lower level elements of the problem/task influenced subsequent levels. Controlling the elements of the problem/task allows the student to concentrate upon completing only those elements that are within his/her range of competence before moving on (Simons and Klein 2007). Pea (Pea 2004) calls this "channelling and focusing" with the intention that this element of control keeps the learner focused on the desired part of the task in hand.

Traditional views of scaffolding have focused primarily on interactions between teacher and peers, with the teacher providing support through features such as:

- guidance and help in setting appropriate goals,
- decomposing the task to appropriate levels which allow the students to recognise if they have achieved a task (Wood, Bruner et al. 1976),
- accentuating features of the task which are the most important and relevant. Students may lack the knowledge which experts have on the task and so

may not know what actions/information are most relevant (Quintana, Reiser et al. 2004),

- providing curriculum material which would assist the students in gaining higher levels of understanding.

Where the difficulty can lie is in determining what to scaffold, when to scaffold and how to scaffold. This is in part determined by the domain context, the tasks to be performed, what the achievements of the learners are to be and if there are any individual student differences which need addressing (Lajoie 2005). The earlier work in this area by both Vygotsky and Wood (Wood, Bruner et al. 1976) applied to one-to-one human interaction and did not consider scaffolding being applied more broadly to incorporate peer-interaction and, more recently, scaffold support through the use of technology tools.

The last two decades of research on learning science has focused on technology and ways in which it may provide types of scaffolding functions (Reiser 2004). Puntambeka (Puntambeka and Hubscher 2005) and (Pea 2004) are concerned that the generalisation of scaffolding within education has become so broad that it is becoming unclear of its significance. Puntambeka does acknowledge that these software tools provide new techniques to support student-learning but they generally provide "static and non adaptive…" scaffolding. However, Quintana (Quintana, Reiser et al. 2004) describes software tools which provide scaffolding in the form of guiding the learners with support features such as helping to keep track of key components and support for planning and performance. Quintana's review of the use of scaffolding software tools also presents a scaffolding design framework that defines rationales and approaches for how software tools can be used to scaffold.

In an educational context, the intention is that these tools reduce the overall complexity of the task which allows the learner to focus on the more important aspects of the problem. For example, a calculator undertakes the task of simple calculation, thus allowing the learner to concentrate of what combinations of calculations are required to solve the problem. The support provided not only assists learning by allowing completion of the task but also ensures that the student learns from the experience. The scaffolding tools need to provide support and to continue actively to encourage and engage the student in the process (Reiser 2004).

The scaffolding construct, whether human or technological, should therefore be dynamic not static. Features need to change, depending on the needs of the student, and are removed when there is evidence that the student no longer requires them (Lajoie 2005).

Simons (Simons and Klein 2007) classifies scaffolds as either 'hard' or 'soft':

- Soft scaffolds can be seen as being dynamic, timely interactions where the teacher circulates amongst the students asking questions and providing immediate feedback.
- Hard scaffolds are static scaffolds and can be developed in advance, based on typical or anticipated difficulties students may face (Brush and Saye 2002) but which can be eliminated through discussion on how to deal with them. Hannafin (Hannafin, Land et al. 1999) describes these types of hard scaffolds as 'conceptual' scaffolding which can help learners reason through complex or unclear problems as well as deal with concepts where known misconceptions are prevalent.

2.5.1. Scaffold Fading

Scaffolding is useful within what Vygotsky called the "zones of proximal development," (ZPD) (Vygotsky 1978). In the context of adult education, ZPD can be seen as the distance between the level of understanding from independent problem-solving and the level of understanding reached through problem-solving with the intervention of teachers and peers.

The principle of teaching within the students' ZPD implies that students will need further explanation and other forms of assistance from their teachers, but also that this scaffolding will diminish as the students' expertise develops. Eventually, students should become able to use what they are learning autonomously and to regulate their own engagement with the task.

Traditional scaffolding was discussed by Wood (Wood, Bruner et al. 1976) in terms of one-to-one interactions, where the teacher reacts to the current situation and modifies the scaffold. With experience, this may be possible on a one-to-one basis but little research has been done on large classroom settings where the teacher would be confronted with many ZPD. One solution is having students work in groups and then scaffold the groups (McNeill, Lizotte et al. 2006). However, McNeill points out that this is still problematic because of the potentially large number of groups and this is where scaffolds such as using technology, or written materials, can be given to the students.

Scaffolding should be seen as providing a framework for a teacher to access different levels of a student's understanding. Students are given support until they can independently demonstrate and articulate their new skills and strategies and have reached higher levels of understanding (Rosenshine and Meister 1992). Once a student's level of knowledge has been demonstrated, the scaffolding can be removed or faded gradually (Lajoie 2005).

2.5.2. Scaffolding and PBL

A widely accepted claim, especially in science education, is the constructivist idea that discovery learning, as opposed to direct instruction, is the best way to get deep and lasting understanding (Klahr and Nigam 2004). The students work in an environment with little guidance and are allowed to discover new rules and ideas rather than being required to memorise what the teacher says (Mayer 2004). The advocates of discovery learning agree with Piaget's (Piaget 2000) assertion that "each time someone prematurely teaches a child something he could have discovered for himself, the child is kept from inventing it and consequently does not understand it completely".

However, over the past half century there has been disagreement over intervention models in learning. On one hand, there are those who state that people work best in an unguided, or minimally guided, environment where students are expected to discover the fundamental principles of the discipline (Kirschner, Sweller et al. 2006). This minimally guided approach, according to Kirschner, can be called discovery learning, PBL, EBL, experiential learning or constructivist learning. The other side of the argument suggests that novice learners should be given strong direct instructional guidance which fully explains to the student the fundamental principles of the discipline.

There is general agreement that, over the years, there is little empirical evidence to support the claim that unguided and experiential methods foster learning (Kirschner, Sweller et al. 2006; Hmelo-Silver, Duncan et al. 2007). Discovery learning in a classroom has been seen to have a negative impact on the students' learning due to minimal feedback resulting in students becoming confused and frustrated, which can lead to misconceptions (Brown and Campione 1994). Hmelo-Silver disagrees with Kirscher's view that PBL and EBL are in the category of minimally-guided instruction and argues that both PBL (whose origins lie in medical education) and EBL (whose origins are in the practice of scientific enquiry) emphasise collaborative learning with the teacher facilitating the learning process and providing guidance and content management on a just-in-time basis e.g. a mini-lecture presenting key information at an appropriate time therefore providing support or 'scaffolding' for the learner.

2.6. Threshold Concept

Teachers in higher education have always had concerns about why some students have great difficulty at certain points in the curriculum whilst others do not. Meyer (Meyer and Land 2006) suggests that when a student is "fixed in a space", he is in a state of "liminality". This state of liminality is the stage of development in relation to the student's existing thinking, where he cannot go backwards, or unlearn, but cannot go forwards without acquiring the new knowledge.

Within each discipline, certain concepts or new knowledge appear to be particularly difficult and troublesome to students and it is of concern to educators to identify why it is that these concepts cause problems for some and not others and how students can be helped to cope with these problems, to resolve them and then to move on from them; to "cross a threshold" (Meyer and Land 2006) of understanding.

The concept of crossing a threshold can mean different things to different people. A threshold concept can be considered as "akin to a portal, opening up a new and previously inaccessible way of thinking about something" (Meyer and Land 2006). It represents a transformed way of understanding, or interpreting, or viewing something without which the learner cannot progress (Meyer and Land 2006). These threshold concepts act as critical portals in the development of a learner's understanding of a subject (Davies and Mangan 2006). Threshold concepts can be seen by educators as the core concepts or building blocks that must be understood before progression can be made in the subject. This understanding, or 'crossing the threshold', may happen in an instant when a student has a tangible, 'Eureka' moment or it could be a gradual process that the student is not consciously aware is happening (Drummond and Jamieson 2005).

The difficulty lies in identifying threshold concepts within a specific discipline because some students may experience a given concept as difficult to grasp whilst some do not (Eckerdal, McCartney et al. 2006). If teaching progresses on the teacher's incorrect assumption that the student has an understanding of a threshold concept, it can invite the student to adopt a surface-learning approach in which the retention of material does not promote understanding but will, the student hopes, be enough for them pass the course (Davies 2003). Alternatively this incorrect assumption may lead to students developing misconceptions which are hard to forget and can distort further new knowledge (Eckerdal, McCartney et al. 2006).

2.6.1. Characteristics of Threshold Concepts

A threshold concept would typically be described as a core learning outcome that progresses an understanding of the subject about which there is general agreement within the discipline. After discussions across a range of disciplines, Meyer (Meyer and Land 2006) describes the general characteristics of threshold concepts as being:

- **Transformative**: once the concept is understood, the way a student looks at things in the discipline is changed.
- **Irreversible**: this change in perception is unlikely to be forgotten or unlearned.
- **Integrative:** it ties together the concepts in ways that were previously hidden or unknown to the student.
- **Bounded:** (but not always) – a boundary or demarcation can define ‘academic territories’ indicating the limits of a conceptual area of the discipline itself.

These characteristics are inter-related. A concept that integrates prior understanding with newly acquired knowledge is transformative because it changes a learner’s perception of their existing understanding of the concept and is, therefore, more likely to be irreversible. Once this knowledge or understanding is acquired, it is almost impossible to unlearn it and a new way of thinking has been established. Using the term ‘irreversibility’ implies that further change is not possible. On the contrary, acquisition of further concepts can again change the way of thinking but not by retracing steps.

As threshold concepts can define the theories and knowledge required within a discipline, they can also help set the boundary limits of a subject. The stronger the integration of concepts, the sharper the boundaries of a subject will appear. Meyer goes on to create a link between these characteristics and what Perkins (Perkins 1999) described in earlier years as “troublesome knowledge”.

This troublesome knowledge which Perkins defines as "that which appears counterintuitive, alien or incoherent" and suggests that one challenge is in recognising the different kinds of knowledge so that a teacher can "create appropriate, targeted constructivist responses to the learner's difficulties". Perkins describes these different kinds of knowledge as:

- **Inert knowledge**: knowledge that a student simply learns and holds in the back of their mind and only recalls when necessary. Students can learn ideas or concepts in various subjects, e.g. sciences, but make no connections to the world around them.
- **Ritual knowledge**: memorization of names, dates, routines in mathematics etc.
- **Conceptually difficult knowledge**: this is most common in mathematics and science, e.g. a heavier object falls at the same rate as a lighter one in a vacuum. Students can learn this to be true but their intuition tells them otherwise and it is conceptually hard for them to grasp. In Computer Science, the use of pointers in programming is, perhaps, a similarly difficult concept although students do not even have an intuitive view of how they operate.
- **Foreign or alien knowledge**: students do not always recognize knowledge as foreign, e.g. in history students tend to view past events through present knowledge and values. Computer Science students similarly can be baffled by the idea of a computer before the introduction of a mouse or a screen or a disk drive. Students need to be aware that there are always alternative perspectives to consider. They may find it difficult to imagine writing software for a computer that had a few hundred bytes of memory, for instance.

The process of understanding something that is troublesome may require a mental transformation (Eckerdal, McCartney et al. 2006). A mental model is a person's internal view of how something works. Ben Ari (Ben-Ari 1998) argues that the lack of mental models plays an important part in why students find it difficult, for example, to learn to program, "A (beginning) Computer Science student has no effective model of a computer". However Eckerdal adds that there are similarities between mental models and threshold concepts in that they both can be transformative. Threshold concepts are accepted concepts within a discipline and can be troublesome to learn while mental models are subjective and individual and can be learned without much effort. It is, for example, relatively easy to learn how a computer works.

Meyer has used the term ‘liminality’ to describe the in-between transition state before passing over the threshold. This needs to be preceded by ‘pre-liminal variation’ (Land, Cousins et al. 2005; Meyer and Land 2006), which is where teachers need to understand what a student already knows. Whilst pre-liminal variation is used in terms of how students approach or come to terms with threshold concepts, it is not possible to guess what previous knowledge or experiences a student brings with them.

2.6.2. Prior knowledge

Entwistle (Entwistle 2003) states that the quality of learning achieved by the student depends on “the knowledge and understanding which they bring with them, along with the associated abilities, motives, conceptions and styles of learning”. It should also be recognised that prior knowledge can be detrimental to a student’s learning if this knowledge is incorrect or inaccurate. Whilst accurate prior knowledge can aid learning, inaccurate prior knowledge can interfere with learning and having to change or correct this knowledge can be more onerous than learning unfamiliar information (Shapiro 2004). However Ausubel (Ausubel, Novak et al. 1968) does not seem to differentiate between accurate and inaccurate knowledge but that “… the most important single factor influencing learning is what the learner already knows. Ascertain this and teach him accordingly”. In Computer Science education many students arrive with knowledge of using a computer but little applicable knowledge or experience of Computer Science as a discipline and so there is a greater need to determine their prior knowledge and ‘teach them accordingly’.

Subject-specific prior knowledge can be attributable to what students already know about a particular subject, e.g. they have already studied the subject at school and this prior knowledge can clearly be demonstrated as a learning outcome, such as, a qualification. Prior knowledge also encompasses their 'learning history' and 'conceptions of learning'. Figure 2.1 illustrates the learning outcome in the context of the mastery of a subject and what contributing factors can be taken into account in achieving this. For example, learning to program requires a form of understanding which is not simply the memorising of factual knowledge even if that is an approach which can be successful with some other traditional A-level material (Drummond and Jamieson 2005). The 'learning history' is the variety of approaches to learning that the student brings with them. Meyer (Meyer and Land 2003) suggests that a history can show "patterns of learning engagement" and, in some cases, these patterns can highlight higher-risk approaches adopted by students. It is a student's approach to learning, their conception of what learning is, their prior knowledge – both general and subject-specific – and their motivation which may explain the variation in their engagement with the subject and the resultant learning outcome.

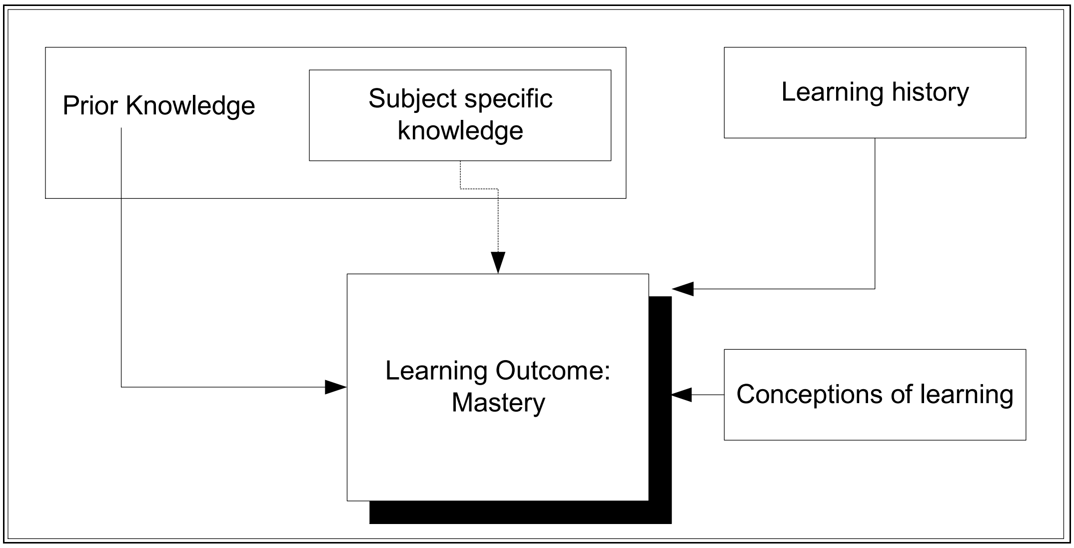

Figure 2.1: Possible factors in achieving mastery

2.6.3. Threshold concepts in Computer Science

Research surrounding the identification of threshold concepts has previously been within disciplines such as economics, maths and physics (Meyer and Shanahan 2001). Computer Science, which is a relatively young discipline, is highly technical and more importantly a subject that is constantly evolving, with software development and technological advancements growing significantly.

Threshold concepts research in computing currently is concerned with issues related to learning to program. Programming is typically an area that causes many students a great deal of trouble. Based on informal interviews and discussions – with input from both students and teachers – and ongoing empirical investigations, a number of candidate concepts have been identified which are seen as ‘difficult to learn’ and, therefore, as recurring problem areas (Eckerdal, McCartney et al. 2006; Boustedt, Eckerdal et al. 2007). These concepts include OOP (Object Oriented Programming), pointers, abstraction and recursion (Sheppard 2007).

If a group of experts are asked to determine threshold concepts in Computer Science, they are likely to identify concepts, such as those above, which are already seen as fundamental within the discipline. Moström (Moström, J.Boustedt et al. 2009) examined how student ways of thinking had been transformed for computing concepts such as abstraction. The study found no general agreement on any one particular concept which suggested that the transformations and thresholds were highly individual. Drummond (Drummond and Jamieson 2005) proposed that crossing a threshold in the context of Computer Science education and, in particular, programming is not one composed purely of computing concepts but motivation was a prime determinant.

2.7. Summary of the literature survey

This chapter has discussed and described some of the classic learning theories and learning styles, both of which are the subject of many publications. Within the literature, there is mixed opinion, certainly in the context of determining a student’s learning style, on the use of what researchers call ‘tools’ or ‘instruments’. The validity of the outcomes generated by these tools to determine a student’s learning style is met with some scepticism. However, it is important that students and teachers alike recognise the different learning styles and recognise which approach is appropriate to use in different situations in order that the learning can be personalised for each student.

Problem-based learning which is seen as a constructivist approach encourages active participation from the student and places them in a situation which requires them to define their own learning needs. The Problem-based learning approach encourages the use of prior knowledge students already possess and to assimilate this with new knowledge. Prior knowledge can be a major contributory factor in the mastery of a subject or concept.

Research directly related to threshold concepts in Computer Science have focused on learning to program and the difficulties this presents to many students. Many students arrive at university without this prior knowledge and find learning to program difficult. However, student mastery of programming, or any other computing topic, can come from their own motivation to learn. This motivation can be encouraged by the teacher and by the use of software tools which provide a personalised learning environment where students are challenged and as a result, become more engaged in their own learning.

Learners, software tools and teachers work together within an education system and it is an over-simplification to consider how tools can scaffold learners without also considering the other aspects of the system. Tools should not replace the teacher but they can provide support that, in the right context, can influence the practices of the learner.

Chapter 3. Computer Science Education in the UK

This thesis is concerned with the relationship between qualifications gained through the secondary education system in England and their effect on a student's performance in computing degree programmes at Durham University. To understand the context in which the students study it is important to consider a number of factors including computing education and the philosophy behind it. In addition this chapter gives an overview of the upper secondary education system in the UK with emphasis on the A-level qualification which is used as the primary recruitment method by UK universities. Finally the Computing and ICT A-level subjects are described and comparisons made between the exam boards' coverage of topics.

Durham offers degree programmes in Computer Science and also in Software Engineering, both of three years' duration. The following sections will briefly describe the domains of Computer Science and Software Engineering and more specifically how education within these domains has evolved.

3.1. Computer Science

Computing itself is a broad discipline that takes important competencies from mathematics, science, engineering and business and, as such, has in the past typically been integrated into one of those departments within universities. Whilst programmers in the 60s typically came from liberal arts, music and mathematics (Tomayko 1998) today's programmers come from a variety of degree programmes which include Computer Science and Software Engineering.

For some years there had been controversy over whether Computer Science was a legitimate academic discipline or, indeed, if it was even a science (The Joint Task Force 2001; Denning 2005). Some viewed it as a vocational specialty for technicians, whilst others saw it as a research platform for mathematicians. “Computer Science is not a science. I was once a scientist and I know" (Maurice Wilkes quoted by Herb Grosch 04/28/2007 in comments requested by the ACM on the review of the Computer Science 2001 curriculum).

However, by the 1990s, Computer Science had accumulated a considerable body of knowledge, research and innovation which spanned both theory and practice so the debate about legitimacy began to diminish. Computer Science spans the range from theory through to programming, thereby offering a comprehensive foundation that permits graduates to adapt to new technologies and ideas – rather than having immediate job-related skills.

Evolutionary change brought about by advances in technology affects Computer Science. With the increase in computing power it is possible to solve problems that would not have been possible before (The Joint Task Force 2001). This evolutionary change affects the body of knowledge for Computer Science and a consequence of this, how it is taught.

3.1.1. The Evolution of Computing Science Education

The computing education curricula began as early at 1946 with Grosch and Eckert at the Watson Scientific Computing Lab at Columbia beginning the world's first computing courses (Brennan 2003). However, the main growth of the Computer Science curriculum began in earnest in the mid 1960s.

“*Computing has dramatically influenced progress in science, engineering, business, and many other areas of human endeavor. In today’s world, nearly everyone needs to use computers, and many will want to study computing in some form. Computing will continue to present challenging career opportunities, and those who work in computing will have a crucial role in shaping the future*”.

(The Joint Task Force 2005)

The emphasis clearly showed the importance of how current and future students are to be educated in the computing disciplines and, over the last 40 years, a number of organisations have developed and published curriculum guidelines (Table 3.1) with regard to various types of degree programmes that were evolving.

Year	Organisation	Discipline
1968	ACM	Computer Science
1972	ACM	Information Systems
1978	IEEE-CS	Computer Science
1985	AITP	Information Systems
1989	BCS/IEE	Computer Science (Software Engineering)
1990	SEI	Software Engineering
1997	AIS (ACM & AITP)	Information Systems
2000	QAA for Higher Educ.	Computing (UK based only)
2001	IEEE/ACM	Computer Science
2004	IEEE/ACM	Software Engineering
2008	IEEE/ACM	Interim review of Computer Science 2001

Table 3.1: Organisations in the evolution of the computing curriculum

In the early 90's, the ACM and IEEE-CS produced a joint curriculum report for computing 'Computing Curriculum 91' (CC91) (The Joint Task Force 1991), which was followed by other reports from organisations detailing different degree programs such as 'Computer Support Services', 'Computer Engineering Technology' etc. By the end of the 1990's' the proliferation of a variety of degree programmes on offer to students both at undergraduate- and postgraduate-levels emphasized the considerable growth in the field of computing.

As a result of this growth, the Joint Task force came together in the late 90's to produce an up-to-date curriculum report (CC2001) (The Joint Task Force 2001). However, they quickly realised that computing had grown so quickly, and in so many dimensions, that no single view of the field seemed adequate. The depth and breadth provided by various computing disciplines called for a new way of defining what the computing curriculum should be. It was, therefore, decided that a curriculum report for each of the major computing disciplines of Computer Science, Software Engineering, Information Systems and Information Technology was to be developed, with an overview report to serve as a guide to these reports (Figure 3.2). It was also an important factor that this series of computing curricula would accommodate emerging computing disciplines.

The IEEE/ACM computing curriculum of 2001 is currently under review (The Review Taskforce 2008) in order to identify changes needed to the Computer Science curriculum and also to identify the fundamental skills and knowledge that all computing students should now possess.

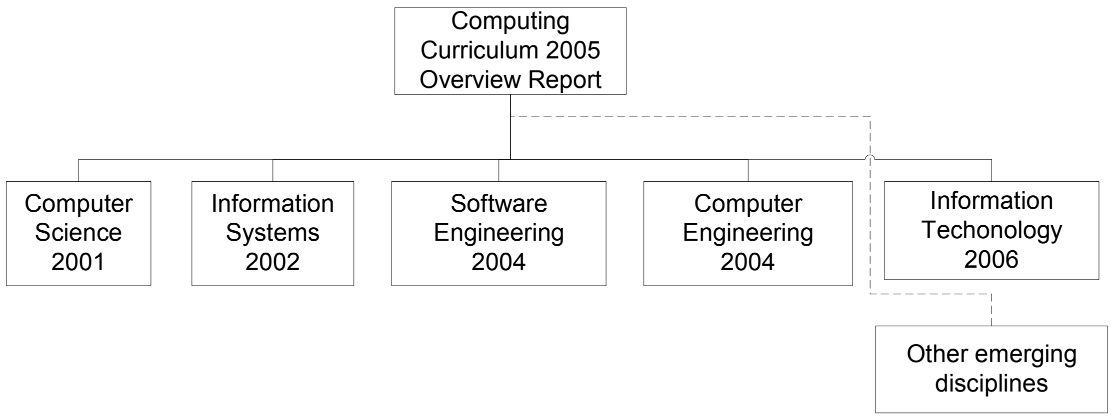

Figure 3.2: Adapted from the Structure of the Computing Curricula Series (The Joint Task Force 2005)

In the UK, the Quality Assurance Agency for Higher Education (QAA) have subject benchmark statements for bachelor level degree with honours for Computing (QAA 2000).

These statements make explicit the general academic characteristics and standards of an honours degree in Computing in the UK and provide general guidance for articulating the learning outcomes associated with the programme. Whilst these benchmarks are not a specification of a detailed curriculum they do provide a set of knowledge areas which list topics seen as indicative within Computing. These knowledge areas are very similar to the Body of Knowledge (BOK) described by the IEEE/ACM computing curricular. This BOK is described in more detail in Chapter 4 section 4.2.1.

The curriculum reports for Computer Science and Software Engineering shown in Figure 3.2, are pertinent to Durham as the two degree programmes offered to students are within these domains. The Software Engineering and Computer Science degrees share the same syllabus in year one and year two but are differentiated in year three by degree specific modules.

3.1.2. Software Engineering

Software engineering had emerged as an area within Computer Science because creating reliable software was becoming more complex and difficult. Similar to other sciences, Computer Science focuses on creating new knowledge. Software Engineering, like other branches of engineering, focuses on rigorous methods for designing and building things that do what they are supposed to do and do them reliably, with the outcome being to see a design converted into a successfully functioning system.

3.1.2.1. Software Engineering Education

Typically in 1988, Software Engineering was being treated primarily as a topic for specialist postgraduate courses or for inclusion as one component in the later stages of undergraduate computing degree programmes. Software Engineering became a more significant part of the Computer Science curriculum in the UK and Australia in the 1980's and the US in the 1990s (The Joint Task Force 2004). In the last ten to fifteen years, the status of Software Engineering as an academic subject has progressed considerably, partly in response to demands from employers and partly as a reflection of progress in the discipline itself.

As Software Engineering draws its foundations from a wide variety of disciplines such as mathematics, engineering and project management, all software engineering students must learn to integrate theory and practice. They must also be able to acquire special domain knowledge, beyond the computing discipline, to support software development in different domains and to appreciate the value of good design. Thus, students need to be able to understand concepts and how to apply them to real problems. Software engineers need to be adaptable and to be able to deal with constantly changing technologies; therefore, these students need to be able to assimilate technology quickly and effectively and use their knowledge in different contexts.

3.1.3. Difference between Computer Science and Software Engineering education

Parnas (Parnas 1998) differentiates between Computer Science education and Software Engineering education in that the former teaches the science and scientific methods needed to extend the science, whilst being heavily focused on the area and continually keeping knowledge up-to-date. The Software Engineering education also requires students to learn the science and the methods needed to apply the science. Software Engineering students, however, may do this in relatively broad terms by being aware of the scientific knowledge and the technology that has already proven reliable (Figure 3.3).

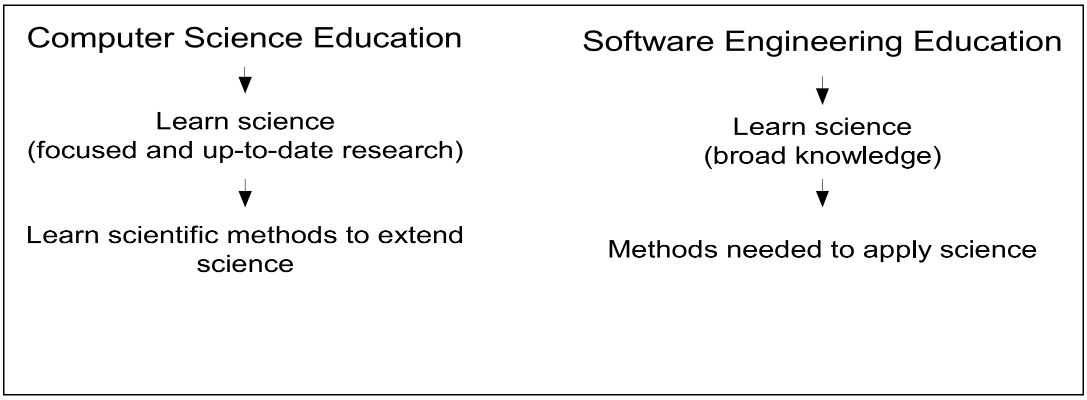

Figure 3.3: Differences between Computer Science and Software Engineering Education - Adapted from Parnas (Parnas 1998)

Both Computer Science and Software Engineering curricula typically require a foundation in programming fundamentals and basic Computer Science theory. The Software Engineering curriculum specifically is involved in building software that is useful and reliable for a customer and satisfies the requirements. The divergence between Computer Science and Software Engineering can be seen as the former's looking at systems, networks, databases, AI and theoretical concepts while the latter looks more towards modelling and analysis, design, verification and validation, quality, the software process, management etc. Parnas (Parnas 1998) advocates that there is a need for degree programmes that follow the traditional engineering approach and educate engineers whose speciality, within engineering, is software construction. In contrast, Shaw (Shaw 2000) argues that there are not as yet enough independent curricula to justify new programmes and that the Computer Science programmes would benefit from adding a stronger engineering sense through most of the curriculum.

3.2. Philosophy of Computer Science Teaching

One of the approaches to education in the higher education sector remains the instructivist method which is more linear and didactic, with knowledge being constructed by the student from books and lectures. This instructivist approach is still useful in that it provides an understanding of fundamental concepts. For example, the Central Processing Unit (CPU), memory etc. could be explicitly taught and thus provide a basic mental model in the learner's mind. However, Computer Science[4] education is not primarily theoretical and, whilst still using this instructivist approach, it is also heavily based on laboratory work (Barrett, Labhrainn et al. 2005). Lab work is integral to the teaching of the subject and follows more closely the constructivist approach (section 2.2.3), with the focus on both independent learning and problem-based learning.

Independent learning can be seen in final year projects where students are expected to demonstrate their knowledge, their problem-solving skills and technical competences acquired from different parts of the degree programme by integrating them all into a single project.

Software programming, unlike other subjects, is not amenable to a 'just learn it' approach (Drummond and Jamieson 2005). Learning to program typically causes many students some difficulty. Novice programmers often come to some impasse where encountering some aspect they don't understand, e.g. recursion, and, rather than stop and reflect on the problem, they simple push ahead using a trial and error technique until the software at least compiles; Shneiderman (Shneiderman 1998)

[4] References to Computer Science, will now be used to mean both Computer Science and Software Engineering.

calls this 'bricolage' or endless debugging. However, the inability to construct knowledge, or understand the core building blocks, results in the learner not being able to progress satisfactorily or to cross a threshold of understanding. Students who are unable to adapt their learning approach are often unable to cope with the demands of learning to program in the absence of considerable support intervention. It is in these programming labs that this support or scaffolding can be provided.

Programming labs are not only an environment for educational intervention from the teacher but also provide social interaction which is important in helping students to construct knowledge with pair-programming being a common activity. Closed labs (timetabled slots in a supervised setting) which provide this intervention and interaction are advocated over open labs (assignments to be worked on whenever convenient) (Ben-Ari 1998; Greening 2000). From a constructivist point of view, the actual type of problem assigned to the student is also important. Some tasks can be highly structured, leaving little room for students to determine the technique that should be used. For some students, creating a program from 'scratch' is very satisfying. However, for others, this prospect is quite daunting and they need scaffolding, such as partly completed programs for them to complete.

Shneiderman (Shneiderman 1998) has a three component philosophy called Relate-Create-Donate, which works well with the constructivist approach and is very pertinent to Computer Science education. This philosophy is about working in collaborative teams (relate) on ambitious projects (create) which have real meaning in the 'outside world' by using real customers to produce something that is of value to others (donate).

Team projects and final year projects fall into Shneiderman's philosophy with the learners having ownership of the learning as well as having ownership of the problem itself. It is recognised however that, because of university regulations in regard to assessment (e.g. examinations being mandated), whilst teachers give students ownership of the problem, in many instances there is a need to dictate the process the students should follow so as to satisfy assessment criteria.

Computer Science team projects are good examples of problem-based learning. Challenging and interesting projects are outlined to students and students are expected to discuss the problem and generate solutions which are based on whatever experience or knowledge they collectively have at that point. The identification of learning issues results in individuals being required to undertake self-directed learning such as learning some new technology, and to then report back to the team (Jarvela 2006; Reason, Terenzini et al. 2006). The problem is then re-examined in the light of the new collective knowledge.

The scaffolding provided to the team can take the form of hard or soft scaffolding. For example, in a student, software development cross-site project, communication between sites has always caused problems for students from both a technical and a social perspective (Burd, Drummond et al. 2003; Drummond and Devlin 2006). In this instance, soft scaffolding can be interpreted as dynamic, timely intervention, or advice, from the teacher who responds to student inexperience. This may, for instance, be in dealing with the more social or communication problems, such as the other site not responding to their requests. Hard scaffolding can be provided in the form of expert advice on known recurring problems through media such as online expert tips or podcasts on using technologies such as version control systems or how to effectively use video-conferencing software and hardware.

Student prior knowledge (discussed in section 2.6.2) can be attributed, in part, to students having studied the subject as school. However, Computing is different from many of the traditional sciences as it is not a core curriculum subject hence Computing students arrive at university with a wide range of abilities and skills. As well as this prior knowledge or subject-specific knowledge, included is a learning history which is the approach(s) to learning that the students bring with them.

One of the research aims of this thesis asks if students arrive in higher education adequately prepared for study in Computer Science. To do this, it is necessary to look at the university admissions system for A-level qualifications and, in particular, the core content of the Computing and ICT A-level subjects studied at school. These are discussed in the following sections.

3.3. 'Sixth Form' Education in the UK

Secondary education in the UK (excluding Scotland) covers compulsory schooling – from the age of eleven to the minimum school leaving age of sixteen – and then the 'Sixth Form', the traditional name for the 16-18 period. Pupils follow a common curriculum leading to qualifications such as GCSE at sixteen. At some schools, pupils may stay on at the school for the sixth form (others attend Further Education colleges) for a further two years, during and after which they may sit the General Certificate of Education Advanced Level (GCE A-levels) or the General Certificate of Education Advanced Subsidiary examinations (GCE AS examinations) or vocational courses or similar qualifications.

Examinations.	
General Certificate of Secondary Education (GCSEs)	Age 16. A wide range of subjects are available with some normally being compulsory.
General Certificate of Education Advanced Supplementary (AS)	AS-levels are now intended to be of the same standard as GCE A-level but cover less content.
Advanced Level GCE A-level (A2)	Assessed mainly by an examination at the end of the course and usually taken by those who are 18 years or over.
General National Vocational Qualifications (GNVQs)	Based on the skills required by employers, combined with the development and understanding of skills needed in vocational areas. Vocational areas covered include business, health and social care and engineering.
Higher National Certificates (HNCs) and Higher National Diplomas (HNDs)	Modular courses of vocational study mostly taken at college or school.
National Vocational Qualifications (NVQs)	Based on skills, knowledge and competencies required by specific occupations and set out by industry-defined standards.

Table 3.2: Types of qualifications in England, Wales and Northern Ireland (adapted)(TeacherNet 2009)

Table 3.2 provides an overview of the examinations commonly offered in the Sixth Form. However the primary, and more common, qualification for entry to many universities including Durham is the GCE A-level (A-levels), with HNDs being acceptable for some programmes. A-level examinations are subject-based and students attempt three or four subjects with some disciplines asking for a specific subject, e.g. a degree programme in Maths will in general require A-level Maths. Computer Science at Durham has typically asked for A-level grades AAB to BBB with Maths being preferred but not compulsory[5].

Examination boards (awarding bodies) are responsible for setting the syllabus and assessment methods for secondary-level qualifications such as GCSE and A-levels. These exam boards are independent of government but do have a regulatory body, the Qualifications and Curriculum Authority (QCA), who ensure that, for each subject, the syllabuses are broadly comparable across exam boards (Clark and Boyle 2006). Schools and colleges have a free choice between exam boards and many schools offer a mixture of boards for their GCSE and A-level subjects.

In England, Wales and Northern Ireland there are currently five exam boards, all of which offer a broad range of qualifications.

1. Assessment and Qualifications Alliance (AQA)
2. Edexcel
3. Oxford, Cambridge and RSA Examinations (OCR)
4. Council for the Curriculum, Examinations & Assessment - not active outside of Northern Ireland (CCEA)
5. Welsh Joint Education Committee (WJEC)

3.4. A-levels as indicators of success at university

A-level qualifications are the primary recruiting criterion for the traditional 18 year old entering university. However, there is debate as to whether the A-level qualification and the grades achieved are good predictors of how a student will perform academically at university. For example, for entry to medical school A-

[5] From 2009, the entry grades for the Computer Science (G400) programme are AAB which must include Mathematics.

levels are seen as "indirect measures of intelligence" and provide the student with "a broad array of facts, ideas and theories about disciplines such as Biology and Chemistry which underpin medicine" (McManus, Powis et al. 2005) and as such are valuable but should not be the only means of assessing the suitability of a candidate.

Many articles which claim that there is no relationship between A-level grades and final degree outcomes have stemmed from work by Sear (Sear 1983) who concluded that the correlation between A-levels and degree results is "generally statistically significant but relatively weak". Work by Bekhradnia (Bekhradnia and Thompson 2002) did, however, find that students with lower A-level grades – on average – progressed with more difficulty. McManus (McManus, Powis et al. 2005) and Alexander (Alexander, Martyn et al. 2003) do, however, acknowledge that "better grades" are often the result of student commitment and motivation, both being personality traits which are desirable for success at university. Lack of these personality traits has been cited as a major reason for non-completion of studies, especially in the first year of study (Ozga and Sukhnandan 1998).

In the Computer Science education community, a question often debated is whether there are particular A-levels which prepare the students for study in Computer Science. More specifically, are the foundational concepts provided by Maths A-level or the prior knowledge and experience gained in, for example Computing A-level, beneficial for academic success, perhaps in year one of study? Studies by Boyle (Boyle, Carter et al. 2002) found there was no correlation between A-level entry and student performance, and, in particular, graduation performance. Boyle also concluded that Maths A-level, or the lack of it, had no impact on student performance overall. Conversely Wilson in (Rountree, Vilner et al. 2004) found there was a positive correlation between having a Maths qualification and success in year one of a computing degree although this was in the USA.

3.5. Computing and ICT A-level

An overarching research question in this thesis is whether particular A-level subjects prepare students for studying Computer Science. The focus here is on the small set of A-level subjects which are identified in section 5.1 Table 5.6, and which include A-level Computing and ICT. Subject specific prior knowledge gained through Computing and ICT should aid learning, certainly in year one, in Computer Science and provide these students with an advantage over those students who have no formal education in the discipline. The following sub-sections provide information on the exam boards offering these subjects and the types of assessment and specific topics covered within A-level Computing and ICT.

3.5.1. Computing and ICT A-level exam board assessment

The exam boards specific to England and the subject qualifications of particular interest to this research are AQA, Edexcel and OCR who offer both A-level Computing and ICT programmes of study[6]. Table 3.3 provides an overview of the assessment structure for each exam board for these subjects. Study for A-level qualifications is over a two year period, with the first year resulting in AS qualifications and the second year in the A2 qualification (A-level).

From Table 3.3 it can be seen that the length of exams for both subjects for the three exam boards appear to be relatively similar except for Edexcel ICT which is exam-assessed at AS and project-assessed at A2 (percentage breakdown was not available). Coursework for both A-level subjects is project-based and further investigation shows that students are required to demonstrate knowledge of analysis, design, implementation, testing and evaluation. For Edexcel there is a large amount of project management included in the final project.

For AQA and OCR Computing, the AS-level forms 50% of the assessment weighting of the full A-level (A2) qualifications. Generally this is 60/40 exam coursework ratio (Table 3.3).

[6] The syllabuses for each exam board are pre-September 2008.

Certainly in A-level Computing, students are expected to be able to demonstrate some practical skill in programming and testing in a programming paradigm such as imperative, procedural or OO with Visual Basic commonly being used. In ICT the programming element would appear to be making use of propriety software rather than actual implementation.

Subject	**Assessment**	**AQA**		**OCR**		**Edexcel**	
		AS	**A2**	**AS**	**A2**	**AS**	**A2**
Computing	Exam	4.5hrs* (50%)	3 hrs (30%)	3hrs (30%)	3hrs (30%)	3hrs (33.35%)	3hrs (33.35%)
	Coursework	0	Proj (20%)	Proj (20%)	Proj (20%)	Proj (16.65%)	Proj (16.65%)
ICT	Exam	3 hrs (30%)	4 hrs (30%)	3hrs (30%)	3hrs (30%)	2 hrs	0
	Coursework	Proj (20%)	Proj (20%)	Proj (20%)	Proj (20%)	e-Portfolio	2 Projs**

Table 3.3: Assessment structure for Computing and ICT A-level subjects

**One exam is a 1.5 hour externally set practical (15%)*

*** One project (Databases) is externally assessed*

Data has been gathered from the syllabus of A-level Computing and A-level ICT for each exam board. A comparison has been made between the subject exam boards on the levels of coverage of topics common to them all and to see if in fact some topics only appear in one exam board. Using a simple ranking system, the level of coverage for comparable topics between the subject exam boards has been ranked as ‘full’, ‘mostly’, ‘partial’ or ‘no coverage’. Completion of this ranking provided an immediate view of the emphasis placed by each exam board on each topic. This ranking document was used as the basis for the mapping of the Computing and ICT syllabuses to the Computer Science year one syllabus. This mapping is described in more detail in section 4.2.1.

The following two sub-sections provide a summary description of the topics covered in Computing and ICT A-levels and where any main differences occur between the exam boards.

3.5.2. Computing

The A-level Computing syllabuses for all boards have an equal amount of 'full' coverage in the topics:

- computer architecture;
- number representation;
- high-level programming languages;
- hardware devices;
- internal components;
- operating systems;
- file types;
- communication and networks;
- Internet and users;
- general purpose applications, e.g. databases, spreadsheets;
- system development.

The topic which has the largest presence is related to computer architecture and, whilst there are other topics covered, but not listed, the coverage is 'partial' or very minimal. AQA is the only exam board, however, to cover representation of images, sound, bit-maps, vectors etc. and only OCR has reasonable coverage of simulation and real-time processing, both in theory and practice (robots and sensors). Edexcel specifically covers the topic of the 'Internet and users' but only at a high-level.

3.5.3. ICT

The ICT syllabus for all exam boards has 'full' coverage on topic areas such as:

- hardware devices;
- social and ethical issues;

- importance of information and data entry/capture;
- information systems within organisations including user support and training;
- evaluation of software and evaluation criteria;
- communication skills.

There are a small number of other topics but coverage is ‘minimal’.

There is ‘partial’ coverage of the more low-level computing topics by OCR and Edexcel but these are covered to simply give an appreciation of the concepts, e.g. what does an operating system do, rather than what they are or how they work. It is only Edexcel ICT which has any coverage of web development, web applications and e-commerce. ICT appears to be more generic and focuses on common applications, investigation and evaluation, information systems strategy & information and information processing.

3.6. Summary

The debate on whether A-levels are a good predictor of success at university and whether Computing and ICT A-level subjects prepare these students for study in Computer Science still meets with mixed reactions from academics, both anecdotally and in research papers. However, for university admission, the A-levels are, and will remain, the primary method of selection for a number of years to come. Computing and ICT are not pre-requisites for entry to Computing programmes and it is debateable as to whether they are ‘useful’ subjects and whether the prior knowledge gained from these will help the students achieve high marks in year one. With the choice of three exam boards seeming to offer comparable syllabuses, does it matter therefore which exam board a student studies under or is there a difference between them which provides some students with a ‘better’, more comprehensive, syllabus than another? The following chapters look at all of these issues and draw conclusions from current research and the statistical analysis undertaken.

Chapter 4. Method

This chapter presents the evidence sources and the methods employed in the analysis of the data. A description is given of the cleaning of the data prior to analysis and the statistical approaches/techniques and investigations used in the analysis itself. An analysis framework is provided which describes how, and in which circumstances, these statistical approaches are to be used.

4.1. Sources of the evidence

Evidence used in this thesis is from documentation, from archived student records and from a questionnaire. Multiple sources of evidence are used to triangulate the research outcomes derived from the research questions, the data collected and the conclusions drawn.

Table 4.4 below [adapted from (Yin 2003)] presents the sources of evidence and the associated strengths and weaknesses of such sources. For example, while the questionnaire had a targeted audience, the responses – whether qualitative or quantitative – were subject to problems such as inaccuracies in recall. Students completing the questionnaire ranged from First Year students who had recently completed A-level Computing or ICT to students who had completed such a qualification over three years earlier.

Source of evidence	Strengths	Weaknesses
Archived Student Records: Student prior qualifications (A-levels) and departmental assessment records	Stable over time. Precise and quantitative.	Accessibility issues due to DPA (Data Protection Acts) regulations – e.g. deletion of student records after 5 years.
Questionnaire:	Targeted – focused directly on case study topics and individual units of analysis (ICT/Computing students). Qualitative and quantitative data captured.	Poorly constructed questions. Inaccuracies in recall. Target subjects limited to those currently at university. Response rates can be low.
Documentation: • IEEE/ACM Curriculum 2001	Stable since 2001	Currently under international review – draft review document released 2008.
• Durham Computer Science curriculum	Full coverage	Often under review/restructuring.
• Pre-university qualifications (English school exam boards)	Relatively stable and accepted and used nationally. Exact content with full coverage.	Currently under national review – new educational initiatives are being introduced.

Table 4.4: Sources of Evidence

The following sub-sections provide further information on these sources of evidence in Table 4.4 and how the data will be used.

4.1.1. Archived Student Records

Archived student records include all individual student details and assessment marks over a period of the first two years of study for each cohort. Year 3 assessment marks have been excluded from this research as the student choice of modules is much wider than in Year 1 or Year 2 of study, with some modules being less popular than others, thus making statistical comparisons problematic. Archived student marks are a major resource and their accuracy and relevance is important.

a. Student records are held centrally within departmental databases. This data includes:

   - student identification number,
   - entry year,
   - pre-entry qualification subjects and marks achieved, and
   - exam boards for each subject.

b. Student assessment marks are held centrally within various department spreadsheets and databases. This data held includes:

   - student identification number,
   - marks achieved for all coursework and
   - exam marks.

4.1.2. Questionnaire

In order to support the outcome of the statistical analysis (Chapter 5), an on-line questionnaire was developed to help capture students' perceptions of how A-level Computing or ICT had impacted on their studies at university. The questionnaire was, therefore, restricted to only those students entering Computer Science with ICT or Computing A-level qualifications. The questionnaire was targeted at Year 1, Year 2 and Year 3 students currently studying within the Department (2005 entry onwards), however the questions focused at modules studied only in year one and two of the degree programme.

The development and deployment of the questionnaire was through Bristol Online Surveys (Bristol-Online-Surveys 2008). The questionnaire consisted of seven questions in total therefore keeping it as short as possible so as to encourage the students to complete it. A copy of the questionnaire can be found in Appendix A. For each module students were asked if their prior knowledge of computing gained through either Computing or ICT A-level subjects was helpful for these modules. Quantitative responses were captured based on a Likert scale which ranged from "Not at all" to "Quite a lot". A N/A response was available. Students were encouraged, through the provision of open response comment boxes, to provide qualitative information to support their quantitative response. Question 6 was designed to capture if a student had any prior experience of computing other than from school. The final question determined if the student perceived prior knowledge of the subject from either school or other had an impact of their studies at University.

4.1.3. Documentation

Documentation gathered has included the following:

a. *IEEE/ACM Computing Curriculum 2001(The Joint Task Force 2001)* This document provides a syllabus of mandatory and optional topics which are recommended to be present within a Computing programme.

b. *Durham Computer Science undergraduate syllabus (Year 1 and Year 2)* Results of a department internal survey resulted in a comprehensive list of topics covered on the Durham Computer Science programmes.

c. *English school exam board national syllabuses and assessment methods.* This was restricted to those subjects which were related to Computing or ICT A-levels.

4.2. Units of Analysis

One of the overarching research questions being addressed by this work is whether students' academic performance is affected by their choices of A-level subjects. To answer this main study question requires the use of *units of analysis*, whether main or embedded, as well as the contexts of these units (Yin 2003). For example, the mapping of the A-level Computing syllabus to the Year 1 Computer Science syllabus (section 4.2.1) required a survey of Computer Science department's staff and an investigation of the various exam boards' A-level syllabuses. The survey is an *embedded* rather than a *main* unit of analysis. In addition to this, specific time boundaries are required (i.e. which cohorts are to be used to determine the limits of the data collection and analysis). The evidence and students contributing to this thesis are listed below:

a. **Student A-level profile**: this profile includes, for each student, all subjects taken at A-level, the grades achieved and the exam boards awarding the qualification. The identified student cohorts are single honours Computer Science and Software Engineering students entering in 2004, 2005, 2006 and who all complete the same core modules in Year 1 and Year 2. Analysis has been restricted to full-time students on degree programmes and has not included those students with non-traditional entry qualifications such as BTEC as they are too few in number. In addition, while the number of International Baccalaureate (Miliband) students is on the increase, numbers for statistical analysis are too low.

b. **Student assessment marks (university level)**: these include the summative assessment marks for each exam and piece of coursework within the core modules in Year 1 and Year 2 for these cohorts of students. The core modules used in the analysis are:

   i. Year 1:
      - Programming and Data Structures (40 credits),
      - Computer Systems (20 credits),
      - Formal Aspects of Computer Science (20 credits).

In addition to the Year 1 core Computer Science modules students must gain credits from a further two modules which are usually taken within another department in the university.

ii. Year 2:
   - Computer Systems II (20 credits),
   - Programming and Reasoning (20 credits),
   - Software Engineering (40 credits),
   - Theory of Computation (20 credits),
   - Software Applications (20 credits).

c. **ICT and Computing A-levels**: For each of the exam boards offering these subjects, the following analysis has been carried out.

   i. A review of the syllabus content for each of the exam boards (Edexcel, OCR, AQA) for A-level ICT and Computing (and derivatives of these names).

   ii. A comparison of the three Computing exam board syllabuses to identify common topics, the coverage and assessment methods used e.g. project, exam.

d. **Computer Science Curriculum**: The following initial analysis was carried out prior to the statistical analysis. This involved both the Durham Computer Science syllabus and the syllabuses for ICT and Computing A-levels. Further details of the mapping exercise are described in section 4.2.1

   i. A survey of the Durham Computer Science syllabus providing a comprehensive list of topics taught.

   ii. Mapping of both ICT and Computing A-level topics to the Durham Computer Science syllabus to identify where overlap or prior knowledge of topics exist.

4.2.1. Mapping of A-level ICT and Computing Syllabuses to Durham Computer Science syllabus

It would be expected that Computing or ICT A-level qualifications should provide a fundamental understanding of Computing and, thus, prepare students for study in Year 1 of a Computer Science programme. To determine if this is the case and to find any topic overlap, a mapping between the Computing and ICT A-level exam boards' syllabuses and the Durham Computer Science Year 1 syllabus has been completed.

This process of mapping began with the completion by Durham Computer Science teaching staff of a paper-based survey which listed the areas, units and, more specifically, the topics which they covered. This survey was based on the Computer Science curriculum specified by the IEEE/ACM Computing Curricula 2001 (The Joint Task Force 2001) and represented the Computer Science Body of Knowledge (BOK). This BOK was organised in a hierarchical way from Areas (disciplinary sub-field), Units (which represent thematic content of the Area) and finally each Unit is subdivided into a set of Topics. The result of the initial staff survey provided a comprehensive list of specific topics and components being taught within the Durham Computer Science department.

The topics taught in both Computing and ICT A-levels for each exam board were mapped to the Durham teaching staff survey results. The topics identified in each exam board's A-level syllabuses (sections 3.5.2 and 3.5.3) were mapped to the topics in the Computer Science syllabus and this highlighted where overlap of these topics between A-level and university Computer Science occurred. While the Computer Science staff survey collated data from all three teaching years, it is the Year 1 topics which are of interest as the mapping showed that any topic overlap applied only to Year 1 of the Computer Science programme.

The result of the mapping of A-level Computing syllabuses (for all exam boards) to the Durham Computer Science Year 1 syllabus has resulted in a concentration of the following topic areas in order of most coverage:

a. **Architecture and organisation**

- Number representation, fetch-execute cycle, I/O and interrupts, instruction sets, registers, addressing, main memory organisation and operations, virtual memory

b. **Programming fundamentals**

- Pointers, lists, stacks, queues, arrays, binary search trees, variables, types, linked structures, stack- and heap-allocation

c. **Information Management**

- Data modelling – ERD, relational databases, DBMS functions, DB architecture and data independence

d. **Netcentric computing**

- Network architectures and protocols, packets and circuit switching

e. **Operating Systems**

- Role and purpose, functionality, interrupts, paging and segmentation

While Computing A-level should not be expected to duplicate the Year 1 Computer Science syllabus, a missing element from all exam boards is an area which is core to Computer Science. This is discrete structures, covering topics such as propositional logic and truth tables.

The A-level ICT syllabus mapping resulted in mainly partial, or no coverage of many Computer Science topics. Areas of highest concentration are within architecture and organisation but to a far lesser extent compared to Computing A-level. Database systems, operating systems and network fundamentals have partial cover for all exam boards.

4.3. Data Cleaning

The previous two sections considered specific aspects, the *Sources of Evidence* and the *Units of Analysis*. This section starts the consideration of the archived student records data which is to be analysed. Before valid analysis can be carried out, a certain amount of data cleaning is required. The main purposes of data cleaning in general are to clean the sources of erroneous or irrelevant data, to identify missing data values and to detect and to remove duplicate records. Various techniques can be used to carry out data cleansing or "scrubbing" (Rahm and Do 2000) before valid analysis is carried out.
To find optimal clusters of data as described in section 4.1.1 requires confidence in the data and that it has been "cleaned". For example, duplicate records have been detected where a student's name – e.g. "Thomas" and "Tom" – has been entered differently in different places or where inconsistent abbreviations – e.g. "A-level", "GCE A level" – are used for the same qualification. The student records database and student assessment spreadsheets have both been cleaned. The size of data in this research is tractable and this has allowed for a final manual cleaning after the various initial automated cleanings. This manual cleaning has also been made easier by the fact that the students and their individual circumstances have been known to the author. The following two sections describe issues with the data which have been addressed.

4.3.1. Student records

Student records held in a central database required the removal of personal information such as name, address, telephone number, school etc. while data such as the student unique identifier, subjects taken, the marks achieved and the names of the exam boards were retained.

A-levels are the predominant qualification for entry to university and are the qualifications under investigation in this thesis. Therefore, all other qualifications, such as Scottish Highers (SQA), International Baccalaureate (Miliband), Foundation level programmes, AS level, City and Guilds, IELTS (English language proficiency), have been removed from the source data.

There was a variation in the naming and content of subjects by the exam boards and, for the purpose of this analysis, A-level subjects covering essentially the same academic field have been grouped under a single title. While each subject listed has its own unique national identification number, there are similarities at the core of, for example the mathematical subjects, and it is appropriate to group them under *Mathematics*. The "groupings" used are set out below:

a. **Mathematics (Maths)**:
   - Applied Mathematics,
   - Pure Mathematics,
   - Pure Mathematics 2,
   - MEI Mathematics,
   - Mathematics (I), (II), (III), (V).
b. **Computing**:
   - Computing,
   - Computer Studies,
   - Computer Science
c. **ICT**:
   - Information and Communications Technology,
   - Information Technology,
   - Information Studies,
   - Information Systems,
   - Using Information Technology

The exam board associated with each subject is also recorded. Once again there was a variation within the university records of the names of the exam board names and these have been standardised as below:

a. **Edexcel**:

- Edexcel (London Examinations),
- Edexcel Vocational Results

b. **AQA**:

- AQA,
- AQA Vocational Results

c. **OCR**:

- Oxford and OCR,
- OCR Vocational

4.3.2. Student assessment

For the purpose of analysis within this thesis, the student marks for all levels of study were transferred, from the databases and spreadsheets where they were originally recorded, to a single spreadsheet. For Year 1, coursework and exams marks for the four core modules (section 4.2.b) have been used, because all Computer Science students take these. In Year 2, all modules are compulsory and, therefore, the marks for all the modules are included.

The cleaning of this data has included many processes and can be illustrated by the following:

- Where a student has completed Year 1 coursework/exams but not Year 2 (e.g. a student withdrew from the course at the end of Year 1) the marks for Year 1 have been removed.

- Where a student has repeated a year the previous marks have been replaced by the marks awarded when the year was retaken.

Potential problems presented by a single source can be aggravated when multiple sources need to be integrated (Rahm and Do 2000) since these sources are often developed independently and for different purposes. However, the merging of the cleaned student subject data (exported from the database) with the marks for coursework and exams (exported from the marks database and spreadsheets) resulted in a completed set of relevant data in a consistent form ready for analysis.

4.4. Approaches to Data Analysis

The investigations and statistical tests used to generate the results in Chapter 5 are described in detail in this section. In addition an analysis framework (section 4.7) has been developed to show the step by step statistical procedures which are used for each investigation and the subsequent results.

Many of the more widely used statistical procedures are parametric tests. These are based on normal distribution of data. The assumption behind hypothesis testing relies on having normally distributed populations and, if this “normality assumption” is not met, the logic behind hypothesis testing is flawed (Field 2005). While it is now felt that the consequence of such violations of assumptions is less severe than previously thought (StatSoft 2007), analysis of data in this research will follow the normality assumption being met. Therefore, it is necessary to determine if the data to be analysed is normally distributed or not and it is the outcome of this that determines the statistical test to be used. If data are normally distributed then parametric tests can be used. If not, a number of options, including data transformation or non-parametric tests, are available.

The statistical software package SPSS (Statistical Package for the Social Sciences) has been used for the analysis of the data.

4.4.1. Distribution of data

To determine which statistical test is to be used in the data analysis, it is useful to plot the frequency distribution of the arithmetic means using a histogram with distribution curve. Histograms can also show cases of outlier data. These outliers in a data set are cases which are far removed in value from the others in the data set. If an outlier is a genuine result, it may indicate an extreme of behaviour and should not routinely be removed without further justification. It is important to determine what to do with outliers at the onset. A number of outliers were identified from Year 1 and Year 2 overall marks in the Results chapter section 5.4.7 for cohorts 2004 and 2006. In particular, there were two outliers with low overall marks. Both were identified as students who did eventually complete Year 1 and Year 2 after resits in both years. One student did, however, eventually fail at the end of Year 2. The other student was given the opportunity to retake the year but chose not to do so. Removing these students from the data set resulted in no significant change in the overall (arithmetic) means and, therefore, they have been retained. Both students had, after all, completed two years of study.

Quantifying the distribution curve of a histogram is provided by statistics such as skewness and kurtosis which indicate if there is a deviation or how much the deviation is from normal. The result of this determines which statistical test to use.

4.4.1.1. Skew and Kurtosis

For a normal distribution curve, skew should be zero while kurtosis (the degree of peakedness of a distribution) should have a value of 3. As skew departs further from zero, a positive value indicates the possibility of a positively skewed distribution – that is, the scores are bunched at the low end of the score scale. A negative value can indicate that scores are bunched together at the high end of the scale (Williams 2008). In order for a curve to be normally distributed, there must be no significant skew and no significant kurtosis. Values of skew and kurtosis in SPSS are not assigned a significance value and, therefore, statistical tables are provided to determine how far away from the normal value of zero a distribution can be before ceasing to be normal.

To overcome any distribution abnormality shown by the skew and kurtosis, further analysis requires either the use of non-parametric tests (section 4.5.2) (i.e. those which do not require the underlying data to be normally distributed) or to transform the data to remove skew or kurtosis. Transformation entails the manipulation of the data in some way. Two methods to deal with positive or negative skew are:

i **Log transformation**: ($\log(X_i)$) takes the logarithm of a set of numbers and squashes the right tail of the distribution and reduces positive skew

ii **Square root transformation**: ($\sqrt{X_i}$) takes the square root of large values – which has more effect than taking the square root of small values – and so brings the larger scores closer to the centre and can, thus, reduce positive skew.

The same process is carried out for negatively skewed data but the scores are reversed (i.e. each score is subtracted from the highest score) (Field 2005). If the transformation corrects the problem of distribution then parametric tests can be run on the transformed scores. If not, non-parametric must be used.

4.4.1.2. Shapiro-Wilks test

Histograms, skew and kurtosis provide visual representation and can show deviation from the normal. However, another useful test to complement these is the Shapiro-Wilks test. This test compares the scores in the sample to a normally distributed set of scores with the same mean and standard deviation. If the test statistic result is:

i **Non-significant** ***(p > .05)***: this means that the distribution of the sample is not significantly different from a normal distribution and parametric tests such as *t*-tests and ANOVA can be used.

ii **Significant** ***(p < .05)***: this means that the distribution of the sample is significantly different from a normal distribution and non-parametric tests such as Kruskal-Wallis test and Mann-Whitney tests need to be used.

4.5. Statistical Tests

Outcomes from the distribution tests described above determine which statistical test must be used. The following sub-sections describe the suite of statistical tests used in this work.

4.5.1. Parametric tests

Parametric tests used in the data analysis are:

i **One-Way ANOVA** – which is an overall test of whether group means differ, and

ii **Independent *t*-test** – which establishes whether or not two means, collected from independent samples, differ significantly.

A further assumption required for parametric tests (other than normal distribution) is Homogeneity of Variance. This is where, even if the arithmetic means for each cohort are different; the variance of the marks – for instance the difference between the highest and lowest grade – is the same for each cohort. Levene's test, described in the following section, is used to test for homogeneity of variances.

4.5.1.1. Levene's test

Levene's test will test the null hypothesis that the variances, in this case between the cohorts, are equal. If the test is non-significant, $p > .05$, then there is no difference between variances. If the test is significant, $p < .05$, then the variances are significantly different and homogeneity of variances has been violated. One cause of this can be inequality in sample size. SPSS does, however, provide two alternative F-ratios (a test for the overall differences between cohort means) which are *Brown & Forsythe*, and *Welch*. *Brown & Forsythe* weights the group variances, not by sample size, but by the inverse of sample size, thereby reducing the impact of larger sample sizes (Field 2005).

4.5.1.2. Independent t-test

The independent *t*-test compares two means when those means have come from two different experimental conditions, e.g. those students who have Maths A-level and those who don't have Maths A-level. Results from the *t*-test provide two tables:

- Group statistics which provide summary statistics for the two experimental conditions, and
- Independent Samples Test which provides the main test statistics. This test produces the *t*-statistic which determines if the difference between two means is significant.

The assumptions behind independent *t*-tests are that data are from a normally distributed population and that homogeneity of variances is maintained. If these assumptions are not met then non-parametric tests must be used.

4.5.1.3. One-way ANOVA

While t-tests compare two means, an ANOVA (analysis of variance) compares several means when these means have come from different groups, e.g. the students for each of the three cohorts who have Computing A-level.

Tables produced from this test are:

- Descriptives,
- Test of Homogeneity of Variances (Levene's test) ,
- ANOVA,
- Post Hoc tests (multiple comparisons) and
- Robust tests of Equality of Means (see section 4.5.1.1.).

The test of whether the group means are the same is represented by the *F*-statistic.

4.5.2. Non Parametric tests

Non-parametric tests must be used if the data are not normally distributed. Non-parametric tests do not make assumptions about the type of data (e.g. that it is normally distributed). Most work by ranking the data and the analysis is then carried out on the ranks rather than the actual data (Field 2005). The two non-parametric tests used in this thesis are the Kruskal-Wallis test and Mann-Whitney test and together these can show the existence of differences between groups and where these differences lie.

4.5.2.1. Kruskal-Wallis test

The Kruskal-Wallis test will test for equality of means across several independent groups, e.g. three cohorts. It is equivalent to the one-way ANOVA (used for normally distributed data). This test indicates if differences exist between groups but not where the differences lie. Follow-up, post hoc, tests such as Mann-Whitney tests are used to find where these differences lie.

4.5.2.2. Mann-Whitney test

The Mann-Witney test determines where the differences are between two independent groups and involves comparing the means of all combinations of pairs of the independent groups. This test is equivalent to performing an ordinary parametric two-sample *t*-test.

If Mann-Whitney tests are used extensively, they can inflate the Type I error rate (α) (belief in there being a genuine effect in populations when in fact there is not) and, therefore, an adjustment is made to ensure that the Type I error rate does not build up to more then .05 (Field 2005). When interpreting the results of a test, instead of using .05 as the critical value for significance for each test, the critical value of .05 is divided by the number of tests to be run. This is known as the Bonferroni correction and it is this critical value that is used.

4.5.3. Effect size

For many of the statistical procedures used, the interpretation of results starts with looking at the significance result, *p*. However, if a test statistic is significant, it does not necessarily mean that the effect it measures is meaningful or important. It is important to use an objective and standardised measure of the size of the observed effect and Pearson's correlation coefficient, *r,* is used to measure the effect. A correlation coefficient of 0 means there is no effect and a value of 1 means there is a perfect effect (Field 2005). Results reported in Chapter 5 show effect sizes (correlation). Guidelines to assess the importance of effects regardless of the significance of the test statistic are:

- $r = .10$ small effect
- $r = .30$ moderate effect
- $r = .50$ large effect
- $r = \geq .70$ very large to perfect

Negative values for effect sizes are useful as they show the direction of the relationship between two variables.

4.6. Investigations for the analysis of data

In the first chapter (section 1.3) a number of research questions have been posed and the investigations outlined in Table 4.5 will be used to answer most of these. Each of the investigations is statistically analysed and the results presented in Chapter 5.

The seven investigations all involve the comparison of different sets of data and the table below assigns a name to each of these numbered investigations and outlines what it is that is being compared:

Investigation	Investigation Name	Investigation outline
1	Cohort analysis	Comparison of means between cohorts for exams, coursework and overall module.
2	(A-level) Single subject analysis	Comparison of means between individual cohorts for identified A-level subjects.
3	(A-level) Single Subject v Non-Subject analysis	Comparison of the combined cohort means for students who have an identified A-level subject versus those that do not.
4	(A-level) Combinations of subjects analysis	Comparison of means between individual cohorts for identified common A-level subject combinations.
5	Coursework category v (A-level) subject analysis	Comparison of the combined cohort means for coursework categories for students who have a specific A-level subject versus those who do not.
6	Year 1 modules v (Computing and ICT A-level) exam board analysis	Comparison of the combined cohort means for the Year 1 overall module mark for students who have Computing and ICT A-level from different exam boards.
7	ICT/Comp v Non-ICT/Comp analysis for coursework	Comparison of the combined cohort means for specific coursework for students who have Computing and ICT A-level versus those who do not.

Table 4.5: Investigations for the analysis of data

The seven investigations have been placed into the following areas: cohort analysis; subject analysis; category of coursework; Computing and ICT A-level syllabus and exam boards. A more detailed description of each area and its associated investigations is provided in the following sub-sections.

4.6.1. Cohort Analysis

Cohort analysis provides a high level overview of how each of the cohorts has performed in relation to each other. This investigation compares cohort means for exams, coursework and overall module marks to determine if there are any statistically significant differences between the cohorts. This analysis can indicate if there are anomalies within the cohorts which could affect the data results. This comparison is independent of A-level subjects taken.

4.6.1.1. Investigation 1: Cohort analysis

Statistical comparisons between (entry) cohorts 2004, 2005, 2006 have made for each of the following variables:

- Year 1 Module (i.e. the mean of all core modules taken in year one of study which is comprised of coursework and exam means)
- Year 1 Exam (i.e. the mean for all core module timed exams taken at the end of year one of study)
- Year 1 Coursework (i.e. the mean for all core module summative coursework completed during year one of study)
- Year 2 Module (i.e. the mean of all modules taken in year two of study which is comprised of coursework and exam means)
- Year 2 Exam (i.e. the mean for all timed exams taken at the end of year two of study)
- Year 2 Coursework (i.e. the mean for all core summative coursework completed during year two of study)
- Year 1 and Year 2 combined (i.e. the mean of the combined Yr1 Module and Yr2 Module means)

4.6.2. Subject Analysis – A-levels

This set of investigations will determine if there is a relationship between specific A-level subjects taken and students' mean performance in Year 1 Modules and Year 2 Modules. Only four A-level subjects were taken by a sufficiently large number of students and, therefore, the analysis has focused specifically on the following A-level subjects (or subject "groupings" as described in Section 5.1):

- Maths,
- Computing,
- ICT and
- Physics

Three specific investigations have been used in this section:

- Investigation 2 takes each of these A-level subjects and compares the means – as in part of Investigation 1 – between the three cohorts but is restricted to those students who have passed the particular subject at A-level.
- Investigation 3 determines if the presence or absence of a particular A-level subject affects the means for the combined cohorts (i.e. all three cohorts combined).
- Investigation 4 compares means between cohorts for the more common combinations of A-level subjects.

4.6.2.1. Investigation 2: Single subject analysis

Single subject analysis will, for each A-level subject listed in Table 5.1, determine whether there is a statistically significant difference between the cohorts for those students who have a specific A-level subject e.g. Maths. The variables used in this analysis are:

- Year 1 Module means
- Year 2 Module means

4.6.2.2. Investigation 3: Single Subject v Non-Subject analysis

"Single Subject v. Non-subject" Analysis will compare the means of the group of students who have a particular subject, e.g. Maths, at A-level with the group of students who do not have this subject. This comparison will show if there is a significant effect on the mean corresponding to the possession or otherwise of an A-level subject. Because of the relatively small numbers, this comparison uses combined cohort means rather than individual cohort means. The analysis at this point has not taken into account what other subjects have been studied with the subject under investigation. Subjects for comparison are:

- Maths (68) vs. Non-Maths (54)
- Comp (62) vs. Non-Comp (60)
- ICT (32) vs Non-ICT (90)
- Physics (62) vs Non-Physics (60)

4.6.2.3. Investigation 4: Combinations of subjects analysis

This investigation is similar to Investigation 2 (section 4.6.2.1) and will, for each pairing of A-level subjects listed, determine whether there is a statistically significant difference between the cohorts for those students who have a specific A-level subject pairing, e.g. Maths & Comp. A-level common combinations to be investigated (identified in 5.1) are:

- Mathematics and Computing (29)
- Mathematics and Physics (43)
- Physics and Computing (25)

Results for the pairing of these subjects do not take into account any third (or other) subject studied. The variables used in this analysis are again:

- Year 1 Module means
- Year 2 Module means

4.6.3. Coursework categories

Coursework for Year 1 and Year 2 for all cohorts have been clustered into category types. The categories are determined by the type of work that is expected from the coursework For example, the *Theoretical* category contains coursework such as logic and algorithms and complexity, which requires the students to demonstrate mathematical skills. The *Theory and Reporting* category not only requires the demonstration of this skill but also for the student to be able to describe and discuss their work. This combines not only a student's mathematical skill but also a student's ability to articulate their work. In some instances, a student may have an incorrect answer to the problem but can demonstrate their understanding of the correct process. In other cases a student may have the right answer but be unable to write a clear explanation of it.

There are no 'hard and fast' categories of Computer Science work and certain choices had to be made. In the end, the chosen categories of coursework were:

- Theory (Mathematics/logic),
- Theory (Mathematics/logic) and Reporting,
- Programming,
- Programming and Reporting,
- Essay and/or Report writing.

4.6.3.1. Investigation 5: Coursework category v subject analysis

This investigation will determine if there is an effect on the combined cohort mean for each of the coursework categories (in section 4.6.3). For a particular A-level subject, the combined cohort means for each coursework category are compared between those who have that particular subject and those who do not, e.g. Maths vs. Non-Maths comparison for the Theory coursework.

4.6.4. Computing and ICT Exam Boards and Syllabuses

Two of the research questions in this thesis ask, firstly:

- whether the choice of school exam board for Computing and ICT A-levels makes a difference to a student's eventual academic achievement at university and secondly
- whether A-level Computing or ICT provide a good preparation for studying Computer Science at university.

Investigation 6 and 7 will help answer these questions and, in both instances, only Year 1 variables will be investigated as the mapping exercise (3.5.2 and 3.5.3) showed that any overlap in topics occurred in the Year 1.

4.6.4.1. Investigation 6: Year 1 modules v exam board analysis

This investigation will determine if there is a statistical difference in performance for students who studied (Computing or ICT A-levels) with a particular exam board; AQA, Edexcel and OCR. Combined cohorts are used. The variables under investigation are:

- Year 1 Module

4.6.4.2. Investigation 7: ICT/Comp v Non-ICT/Comp analysis for coursework

Investigation 7 will show whether Computing and ICT students perform better in certain types of work in Year 1 perhaps because they have already encountered some of this work during their A-level study. The results from the mapping exercise (A-level –> Computer Science) described in (3.5.2 and 3.5.3), identified the topics which were covered in all the A-level syllabuses under investigation and which were then covered again in the University's Computer Science programme. It was also seen that these were restricted to two Year 1 modules, Programming and Data Structures (PDS) and Computer Systems (CSys). In these two Computer Science modules, specific coursework assignments have been identified and it is these assignments which are used in the analysis. In the CSys module for example, all students undertake an operating systems assignment. The students can be split into two groups, "Comp" (i.e. those students in all three cohorts combined who have A-level ICT or Computing) and "Non-Comp" (i.e. the other students). A comparison will be made between the means for this assignment of the two groups of students. Variables under investigation are:

- Year 1 coursework (see Section 4.6.1.1) and its component parts.

4.7. Analysis Framework

For each investigation listed in Table 4.5, the results are presented in Chapter 5. These results are generated by following a number of statistical procedural steps, all of which have been described in section 4.5. Which statistical test to use is determined by whether the data being analysed is from a normally distributed population. The analysis framework below has been developed to step through the process of firstly determining if the distribution of data meets the "normality assumption" with the outcome of this then determining which statistical test must be used. A series of tests are undertaken for each of the investigations in Chapter 5. For each of these tests there is an associated table provided which clearly sets out which steps from the analysis framework have been used and in which order. Not all steps are used for each investigation and these steps are not necessarily used in numerical order.

STEP 1: Test for normal distribution of data

a. Produce descriptive statistics and frequency distribution using a histogram.

b. Quantify shape of distribution using descriptive statistics for skew and kurtosis. Use statistic tables available to determine the value of skew using the skew statistic and the sample size. In this work, the sample size is 122 and a value of skew > .35 indicates that the level of skew is high and that skew is significant.

c. Identify outliers which can cause skew. Use box plots to show clearly which case(s) are the outlier values. If values are valid, decide if they are to be left in. If they are not valid, they should be removed.

d. Use the Shapiro-Wilks test of normality: if the significant value $p > .05$ then the distribution of the sample is not significantly different from a normal distribution and Step 3 should be performed. If $p < .05$ then perform Step 2.

STEP 2: If data are not normally distributed:

a. Transform the data: use a mathematical function (logarithm or square root) to try to correct the distribution abnormality. If the transformation corrects the distribution problem then run the parametric tests, Step 4, on the transformed scores. If the transformation does not correct the problem then perform Step 5.

STEP 3: If data are normally distributed, check homogeneity of variance.

Test for homogeneity of variance using Levene's test:

a. If $p < .05$ then the test is violated and variances are significantly different. Attempt to transform the data – Step 2.
b. If $p > .05$ homogeneity of variance holds and satisfies the assumption of parametric tests. Use parametric tests – Step 4.

STEP 4: Parametric test

a. An ANOVA produces the five tables:
   - Descriptives,
   - Test of Homogeneity of Variances (Levene's test) ,
   - ANOVA,
   - Post Hoc tests (multiple comparisons) and
   - Robust tests of Equality of Means (see section 4.5.1.1).

i If Levene's test gives $p < .05$ read Robust test of equality means table
ii If Levene's test gives $p > .05$ read the ANOVA table.
iii Read Scheffe test in Multiple comparison table which shows where any differences in cohorts are.

b Independent *t*-tests produce the two tables:
   - Group statistics providing a summary for the two experimental conditions and
   - Independent Samples Test which provides the main test statistics. This test produces the *t*-statistic which determines if the difference between two means is significant.

i. If Levene's test gives $p < .05$ then variances are different and the statistics row *Equal variances not assumed* is used
ii. If Levene's test gives $p > .05$ then variances are similar and the statistics row *Equal variances assumed* is used

STEP 5: Non-parametric test

a. For each Kruskall-Wallis test, two tables are produced:

   - a Ranks table and
   - a Test Statistics table.

i. Read the *Monte Carlo Sig* in the test statistic table for the significant value. Please note that the *Test Statistic, H,* appears in SPSS output as "Chi-Square".

ii. Follow this up with Mann-Whitney tests between pairs of conditions

b. For each Mann-Whitney test, the two tables are also produced:

- a Ranks table and
- a Test Statistic table.

If many tests are run, a Bonferroni correction should be applied to the significance level $\frac{0.5}{n}$ (where n is the number of tests) and all effects are reported at that level of significance.

STEP 6: Report results

Standard reporting syntax for each statistical test is used. For:

a. One-way ANOVA tests: report the *F*-ratio, degrees of freedom, *df*, and the effect size, *r*.

b. Independent *t*-tests: report the *t*-statistic, *df,* and the significance (sig) value, *p*, and effect size *r*.

c. Kruskal-Wallis tests: report the *H* statistic, the *df* and the sig value, *p*.

d. Mann-Whitney tests: report the *U* statistic, the sig value, *p*, and the effect size *r*.

The degree of freedom, *df,* reported in test results is the number of values in the final calculation of a statistic that are free to vary while the calculated statistic is unchanged. Suppose, for example, four numbers have a mean of 10. If the mean is to remain at 10 and three numbers take random values then the fourth number is forced to have a specific value. Its value is constrained by the three varying numbers. The *degree of freedom* is, therefore, said to be three (i.e. 4-1) (Field 2005).

4.8. Student A-level profile

The A-level academic profile of the students prior to university is provided in Table 5.1 and Table 5.2. This profile includes the frequency of A-level subjects appearing across all cohorts and the most common combinations of A-levels that Computer Science students have taken. It is the most common subjects and combinations of subjects that are the basis for the data analysis in these investigations.

4.9. Student Questionnaire

This thesis has concentrated on the statistical analysis of student marks for exam and coursework. However, to supplement those findings a web-based on-line questionnaire (Bristol-Online-Surveys 2008) was undertaken with those students who had studied ICT or Computing at A-level prior to entry to higher education. The purpose of this questionnaire was to collect student perceptions of how the study of either of these subjects had helped prepare them for First Year studies at university.

The cohorts asked to complete the questionnaire were entry years 2005/06, 2006/07, 2007/8 and 2008/09. While these cohorts do not exactly match the cohorts which are part of the main statistical analysis, cohorts 2007/08 and 2008/09 can offer a valuable contribution in the collection of questionnaire data.

The questionnaire consisted of seven questions, each of which used a Likert scale plus an optional free text response area for each question. Each module in Year 1 and Year 2 was part of the questionnaire and students were asked to rate (on a scale ranging from “Not at all” to “Quite a lot”) how they believed their Computing or ICT A-level had helped in their study not only of that particular module but also overall.

Results reported were restricted to the four compulsory modules (80 credits) from Year 1 because respondents for 2007/08 and 2008/09 only had knowledge of the first year of study at the time of their completing the questionnaire. The modules are:
:

- Programming & Data Structures (PDS) – 40 credits,
- Formal Aspects of Computer Science (FA) – 20 credits, and
- Computer Systems (CSys) – 20 credits

Chapter 5. Results

This chapter presents a profile of the A-level subjects, and to a lesser extent grades, of Durham Computer Science students. The quantitative analysis involves a series of investigations which address the research questions and which fall into five main areas of interest. A summary is provided for each of these areas. Qualitative analysis and the results of a student questionnaire are presented. A discussion of the results concludes this chapter

5.1. Student Profile by A-level subject

Analysis of the cohort data had identified a wide range of A-level subjects taken by the student cohorts ranging from Art through to Sociology. However, across all three cohorts, the subjects that occur most frequently are shown in Table 5.6:

Subject	Freq: all cohorts	%: all cohorts	2004 entry (46 students)	2005 entry (34 students)	2006 entry (42 students)
Computing	62	(50.8)	21 (45.7%)	19 (55.9%)	22 (52%)
Mathematics	68	(55.7)	29 (63%)	14 (41.2%)	25 (59.5%)
ICT	32	(26.2)	13 (28.3%)	9 (26.5%)	10 (23.8%)
Physics	59	(48.4)	20 (43.5%)	14 (41.2%)	25 (59.5%)

Table 5.6 : Frequency of A-level subjects for all cohorts

For detailed statistical analysis, the number of students studying a specific A-level should be no fewer than 30. Therefore, all subjects that have been studied by fewer than 30 students have been removed.

There are only two students within the sample population who do not have at least one of the subjects in Table 5.6. The subject combination at A-level for these two students is

- Student 1 had Economics, History, English, and
- Student 2 had Language, Music and Psychology.

Whilst two cases are not sufficient for statistical analysis, for completeness a comparison has been made and details of these students' outcomes are given.

1. Student 1 (cohort 2006) – A-level grades ABE. This student performed well below the overall first year mean of 56.27. The student marks for both exam and coursework were in the 3rd classification degree range.

2. Student 2 (cohort 2005) – A-level grades AAB. This student performed above the overall first year mean of 58.28. The student marks for both exam and coursework were in the 2.i classification range.

The more frequent combinations of the subjects in Table 5.6 are shown in Table 5.7. This table shows that Maths-Physics is the most popular combination.

Subject Combinations	**Freq: all cohorts**	**%: all cohorts**	**2004** (46 students)	**2005** (34 students)	**2006** (42 students)
Maths and Comp	29	(35.38)	12	6	11
Maths and ICT	9	(10.98)	5	2	2
Maths and Physics	43	(52.46)	15	9	19
Maths, Comp, Physics	15	(18.3)	4	4	7
Comp and Physics	25	(30.5)	6	7	12
Physics and ICT	13	(15.86)	5	4	4
Maths, Phys, ICT	7	(8.54)	3	2	2
Maths, fMaths, Phys	6	(7.32)	3	2	1
Maths, FMaths, Comp	2	(1.22)	1	0	1

Table 5.7: Frequency of combination of A-level subjects for all cohorts

Table 5.8 outlines for each cohort the grades students achieved at A-level independent of A-level subject. The A-level entry points have been calculated for each A-level student. The scale used is:

- 10 points for an A grade at A-level,
- 8 points for B,
- 6 points for C,
- 4 points for D,
- 2 points for E.

The best three A-level grades (if more than 3 grades acheived) are used in the department to determine if university entry requirements have been satisfied.

The points average for all A-level students in this analysis is 26.51 (which can equate to roughly ABB (26pts)) and is determined by the A, B and C grades only. Grades D and E are provided for information only.

A-level Grades by cohort						Grade point average
	A	B	C	D	E	
2004 (46)	59	60	20	5	3	25.86
2005 (34)	60	37	11	1	0	28.29
2006 (42)	52	62	8	1	1	25.40

Table 5.8 A-Level Grades achieved by Entry Year

From Table 5.8 it can be seen that there is a slight difference in the number of A or B grades between cohorts, although 2005, which is the small cohort, has the greatest number of A grades and therefore the highest grade point average.

5.2. Representation of the population

One of the simplest models used in statistics is the mean and, as such, it is widely used to test the data presented in this chapter. The (arithmetic) mean is the "average" of a set of values, or distribution and is calculated by dividing the sum of the values by the number of values. However, for skewed distributions (section 4.4.1.1.), the mean is not necessarily the same as the median (the "middle" value). Figure 5.4, below provides a high-level view of the means for the student population under investigation. Each individual cohort-mean represents the combination of Year 1 and Year 2 of study. There is no statistically significant difference between the cohort-means.

The combined-cohort size (i.e. the size of all three cohorts taken as a whole) used for analysis is 122 students, made up of 46 students in 2004; 34 students in 2005 and 42 students in 2006.

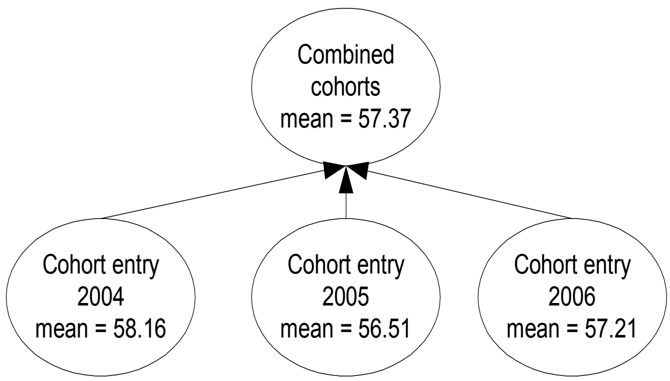

Figure 5.4: Means for each cohort

The combined-cohorts mean shown in Figure 5.4 represents the mean for the group made up of all three cohorts of students taken as one. General descriptive statistics for all variables, irrespective of cohort, are presented in Table 5.9. Associated with each variable are the results for the different statistical models, for example mean and median.

Statistics

		Yr1 Modules	Yr2 Modules	Yr1 Exams	Yr2 Exams	Yr1 Coursework	Yr2 Coursework	Y1andY2
N	Valid	122	122	122	122	122	122	122
	Missing	0	0	0	0	0	0	0
Mean		59.8634	54.8934	56.0847	52.6607	65.0874	55.7770	56.7572
Std. Error of Mean		1.01757	.94722	1.15862	1.05014	.93733	.96931	.88591
Median		61.1667	55.3000	56.6667	52.8000	66.8333	57.6000	57.1875
Mode		65.67[a]	55.60	62.33	48.60[a]	74.00	56.40[a]	44.25
Std. Deviation		11.23942	10.46234	12.79732	11.5992	10.35316	10.70640	9.78514
Skewness		-.585	-.466	-.376	-.260	-1.242	-.703	-.591
Std. Error of Skewness		.219	.219	.219	.219	.219	.219	.219
Kurtosis		.186	.396	-.162	.127	2.445	.462	.569
Std. Error of Kurtosis		.435	.435	.435	.435	.435	.435	.435

a. Multiple modes exist. The smallest value is shown

Table 5.9: General statistics overview for all variables – combined-cohorts

Measures to note from Table 5.9 are 'Skewness' and 'Kurtosis' (section 4.4.1.1), each with an associated standard error ('Std. Error of ….'). As all Skewness scores are negative, this indicates that there is significant skew of scores on the right of normal distribution. This deviation from a normal distribution of scores is particularly noticeable for coursework in Year 1 with a skew value of -1.242. Figure 5.5, as an example, illustrates that the largest values are not at the centre of the bell curve and, moving right from the centre, the bars do not decrease.

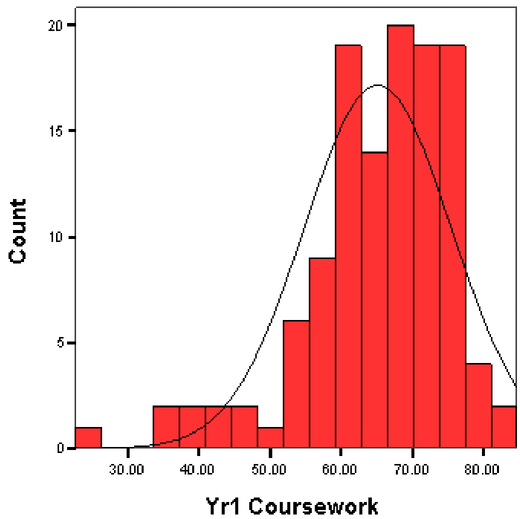

Figure 5.5: Year 1 Coursework – an example of skewed distribution

The combined mean for each variable is provided in Table 5.9. The variables represent:

- Year 1 Module: the mean of all core-modules (exams and coursework together) taken in year one of study.
- Year 1 Exam: the mean for all core-module timed exams taken at the end of year one of study.
- Year 1 Coursework: the mean for all core-module summative coursework completed during year one of study.
- Year 2 Module: the mean of all core-modules (exams and coursework together) taken in year two of study.
- Year 2 Exam: the mean for all core-module timed exams taken at the end of year two of study.
- Year 2 Coursework: the mean for all core-module summative coursework completed during year two of study.
- Year 1 & Year 2: the combined mean for the 'Module' means above.

The variables in Table 5.9 are the subject of analysis within this chapter. These variables are used in the investigations. The investigations which address the research questions posed in 1.3 fall into five main groups which are described below and also shown pictorially in Figure 5.6:

1. Analysis of cohorts (section 5.4): compares the means between cohorts for exam, coursework and a combined overall mean for year one and for year two – irrespective of A-level subjects.

2. Subject(s) analysis (section 5.5): compares the means between cohorts, first for Year 1 and, secondly, for Year 2 for a single A-level subject, e.g. Maths. In addition to this, the combined-cohort means are compared for those who have a particular A-level subject with those students who do not (section 5.5). Lastly, the 'combination of two common A-level subjects' means between cohorts are analysed for Year 1 Module (section 5.6).

3. Coursework categories (section 5.7): types of Computer Science coursework have been categorised (4.6.3). For each A-level subject, the

combined means of students with or without that subject are compared for each type/category of coursework.

4. Computing and ICT A-level syllabuses (section 5.8): results of the mapping between the syllabuses of these subjects and year one content in Computer Science were compared and revealed that any overlap occurs in only two modules, Computer Systems and in Programming & Data Structures. The relevant assignments in these modules were identified and, for each of these assignments, this investigation compares the means of the students who have the appropriate A-level with those who do not.

5. A-level Exam Boards (section 5.9): For students who have A-level Computing or ICT, the means achieved are compared, depending on which exam board they used, to determine if one exam board appears to prepare students better. The means for Year 1 modules are compared.

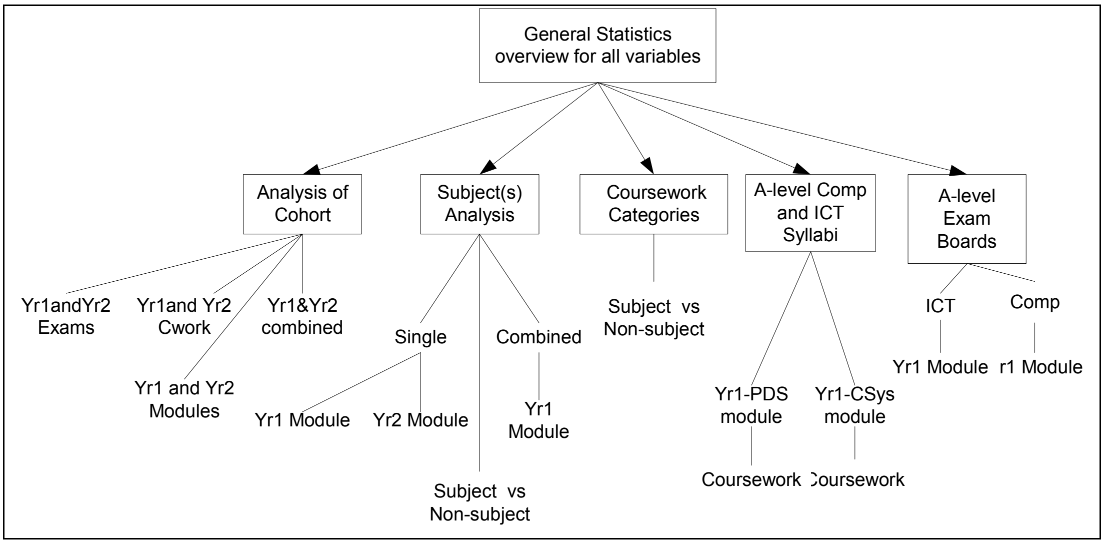

Figure 5.6: Overview of the investigation strategy

5.3. Interpretation of the results section

In the following section, each set of results is preceded by a table which gives an overview of:

- which investigation the test is related to
- the statistical 'procedural steps' which were required to generate the results
- the statistical test used.

The 'Investigation' column gives the investigation number. Further details of the variables that are being used and the rationale for the investigation are given in section 4.6. The 'Procedural Steps' column lists the statistical procedures and reporting mechanism which have been used to produce the results. Further details of these can be found in the Analysis Framework in section 4.7. The 'Statistical test' column provides the name of the actual statistical test(s) used. Further details of these tests can be found in section 4.5.

5.4. Analysis of cohorts

Investigation	Procedural Steps	Statistical test
1	1d, 2a, 5a, 5b, 6c, 6d.	Kruskal-Wallis and Mann Whitney

Table 5.10: Investigation 1 - Analysis of means by cohort

The Ranks table (Table 5.11) is provided to show the breakdown of mean ranks for each cohort. The Test Statistics table (Table 5.12) is produced by the Kruskal Wallis test.

Ranks

	University Entry Year	N	Mean Rank
Yr1 Modules	2004	46	77.58
	2005	34	55.21
	2006	42	48.99
	Total	122	
Yr2 Modules	2004	46	51.82
	2005	34	60.13
	2006	42	73.21
	Total	122	
Yr1 Exams	2004	46	78.61
	2005	34	52.01
	2006	42	50.44
	Total	122	
Yr2 Exams	2004	46	51.08
	2005	34	59.62
	2006	42	74.44
	Total	122	
Yr1 Coursework	2004	46	67.24
	2005	34	67.38
	2006	42	50.45
	Total	122	
Yr2 Coursework	2004	46	55.24
	2005	34	53.59
	2006	42	74.76
	Total	122	
Y1andY2	2004	46	62.66
	2005	34	56.81
	2006	42	64.02
	Total	122	

Table 5.11: Rank table: Analysis of means by cohort

Test Statistics[b,c]

			Yr1 Modules	Yr2 Modules	Yr1 Exams	Yr2 Exams	Yr1 Coursework	Yr2 Coursework	Y1andY2
Chi-Square			15.846	8.112	17.325	9.719	6.255	9.053	.862
df			2	2	2	2	2	2	2
Asymp. Sig.			.000	.017	.000	.008	.044	.011	.650
Monte Carlo Sig.	Sig.		.000[a]	.015[a]	.000[a]	.007[a]	.045[a]	.008[a]	.660[a]
	99% Confidence Interval	Lower Bound	.000	.012	.000	.005	.039	.006	.647
		Upper Bound	.001	.018	.000	.009	.050	.011	.672

a. Based on 10000 sampled tables with starting seed 2000000.

b. Kruskal Wallis Test

c. Grouping Variable: University Entry Year

Table 5.12: Results of Kruskal-Wallis test for Entry year

In Table 5.12, the significant values (Monte Carlo Sig) are $p < .05$ (except for Y1 andY2) which indicates there is a significant difference between cohorts. The lower and upper bound confidence interval for significance for each variable is also important, for example for Year 1 Coursework the actual 99% Confidence Interval Lower Bound value is .037 and the Upper Bound value is .048. The fact that these values do not cross the critical value .05 means that the significant effect is genuine.

Thus from Table 5.12 it can be noted that there is a significant effect between:

- Year 1 modules by the entry year (H (2) = 15.85, $p < .05$)
- Year 2 modules by the entry year (H (2) = 8.11, $p < .05$)
- Year 1 exams by the entry year (H (2) = 17.33, $p < .05$)
- Year 2 exams by the entry year (H (2) = 9.72, $p < .05$)
- Year 1 coursework by the entry year (H (2) = 6.3, $p < .05$)
- Year 2 coursework by the entry year (H (2) = 9.1, $p < .05$)

However there is no significant effect for:

- Year 1 and Year 2 overall by the entry year (H (2) =.862, $p >.05$)

The above results show there are that there are significant differences but not where the differences lie. By simply looking at the ranking table (Table 5.11), for example Year 1 Modules, it can be seen that there is a difference in ranking between 2004 and 2006. Visual representation of data can also be seen in the box-plots, e.g. Figure 5.7. However, whilst box-plots are visually useful, they can be considered subjective and, therefore, post hoc tests using Mann-Whitney is required. A Bonferroni correction has been applied (discussion on this can be found in 4.5.2.2) and so all effects are reported at a level of significance of .0167 (.05/3 tests = .0167) in the following Mann-Whitney tests.

Results of Mann-Whitney tests

For each variable shown in Table 5.9, Mann-Whitney tests between pairs of conditions are run. The conditions are the individual cohorts:

- Test 1: 2004 compared to 2005,
- Test 2: 2004 compared to 2006,
- Test 3: 2005 compared to 2006.

A Mann-Whitney test produces two tables and, therefore, for each variable, three sets of these tables are produced (one set for each test). For clarity, subsequent reporting of results will not provide the full set of tables but only those that produce an "interesting" outcome. However, for each of the following Mann-Whitney tests, results are preceded by a boxplot which provides an immediate visual representation of results.

Boxplots (sometimes knows as box-whisker diagrams) such as figure 5.7 and all subsequent boxplot diagrams, show the distribution of the overall year one modules marks over the three cohorts. The boxplots show the lowest score (the bottom horizontal line on each plot) and the highest (the top horizontal line on each plot). The distance between the lowest horizontal line and the lowest edge of the actual box is the range between which the lowest 25% of marks fall (the bottom quartile). The box itself represents the middle 50% of marks (the interquartile range). The line in the box represents the value of the median, that is the middle score. The distance between the top edge of the box and the top horizontal line shows the range that the top 25% of marks fall (the top quartile). Outlier data values are represented by circles.

5.4.1. Year 1 Overall Modules results

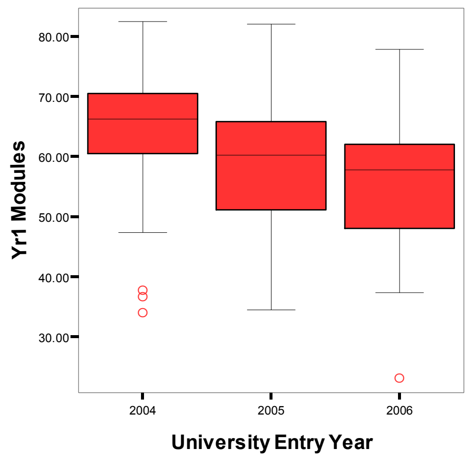

Figure 5.7: Year 1 overall module mean by cohort

Year 1 Module means (Figure 5.7) by entry year show a decline in performance between cohorts 2004 and 2006.

Table 5.13 is an example of one set of tables produced for Year 1 Modules showing the comparison between 2004 and 2005. Two other pairs of tables are also produced (not shown) which show the comparisons between 2004 and 2006, and 2005 and 2006.

Ranks

	University Entry Year	N	Mean Rank	Sum of Ranks
Yr1 Modules	2004	46	46.71	2148.50
	2005	34	32.10	1091.50
	Total	80		

Test Statistics[b]

	Yr1 Modules
Mann-Whitney U	496.500
Wilcoxon W	1091.500
Z	-2.779
Asymp. Sig. (2-tailed)	.005

b. Grouping Variable: University Entry Year

Table 5.13 : Tables produced from Mann-Whitney tests

The results of the three comparison tests clearly show a statistically significant difference between 2004 and 2005, and 2004 and 2006.

Results: Year 1 Module in:

- 2004 differ significantly from 2005: U=496.50, $p < .0167$, $r = -.31$ medium effect (see section 4.5.3 for discussion on effects)
- 2004 differ significantly from 2006: U=512.88, $p < .0167$, $r = -.40$ medium effect
- 2005 does not differ significantly from 2006: U=640.50, $p > .0167$, $r = -.09$ small effect

5.4.2. Year 1 Overall Exam results

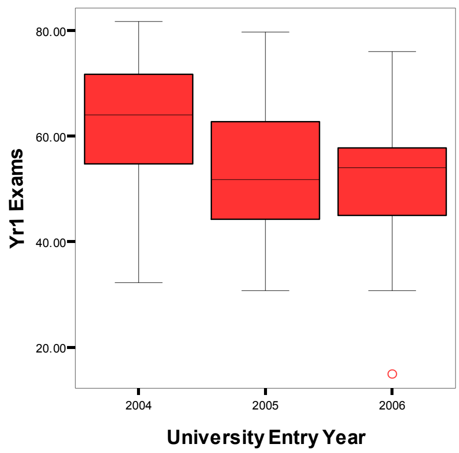

Figure 5.8: Year 1 overall exam mean by cohort

Year 1 exam means (Figure 5.8) by entry year show a highly significant difference between 2004 and 2005, and 2004 and 2006. There is no significant statistical difference between 2005 and 2006.

Results: Year 1 Exams in:

- 2004 differ significantly from 2005: U=453.0, $p < .0167$, $r = -.36$ (medium effect).
- 2004 differ significantly from 2006: U=508.0, $p < .0167$, $r = -.40$ (medium effect).
- 2005 does not differ significantly from 2006: U=707.50, $p > .0167$, $r = -.009$ (small effect).

5.4.3. Year 1 Coursework results

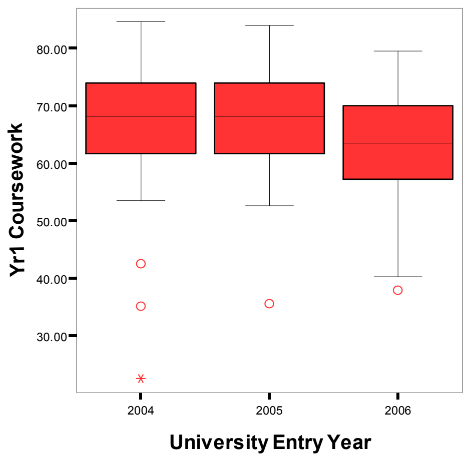

Figure 5.9: Year 1 Coursework mean by cohort

Year 1 coursework-means (Figure 5.9) by entry year show there is no significant statistical difference between cohorts. Thus -

Results: Year 1 Coursework in:

- 2004 does not differ significantly from 2005: U=778.0, $p > .0167$, $r = -.004$.
- 2004 does not differ significantly from 2006: U=698.0, $p > .0167$, $r = -.24$.

- 2005 does not differ significantly from 2006: U=518.0, $p > .0167$, $r = -.23$.

In each of these cases, the effect r is defined as small.

5.4.4. Year 2 Module results

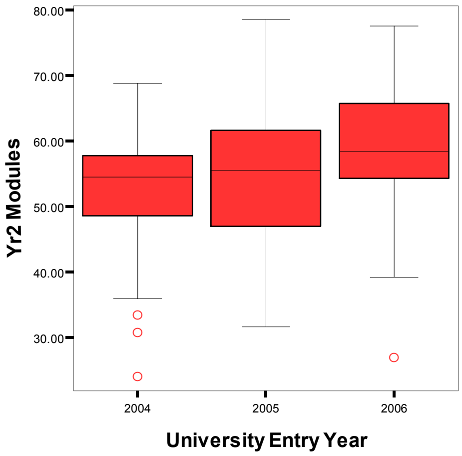

Figure 5.10: Year 2 Module mean by cohort

Year 2 module-means (Figure 5.10) by entry cohort visibly show an improvement in performance between cohorts 2004 and 2006. However, the only statistically significant difference is between cohorts 2004 and 2006. There is no significant difference between 2005 and 2006 or 2004 and 2005.

Results: Year 2 Module in:

- 2004 does not differ significantly from 2005: U=682.0, $p > .0167$, $r = -.10$.
- 2004 differs significantly from 2006: U=620.50, $p < .0167$, $r = -.31$.
- 2005 does not differ significantly from 2006: U=567.50, $p > .0167$, $r = -.18$.

In each of these cases, the effect r is defined as small.

5.4.5. Year 2 Exam results

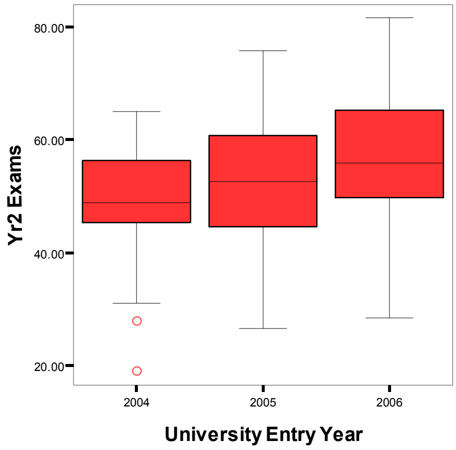

Figure 5.11 Year 2 Exam-mean by cohort

Year 2 exam-means (Figure 5.11) result in a highly significant difference between 2004 and 2006. There is no statistically significant difference in means between 2004 and 2005 or 2005 and 2006.

Results: Year 2 Exam in:

- 2004 does not differ significantly from 2005: U=681.0, p > .0167, r = -.11 (small effect).
- 2004 differs significantly from 2006: U=587.50, p < .0167, r = -.34 (medium effect).
- 2005 do not differ significantly from 2006: U=549.0, p > .0167, r = -.20 (small effect).

5.4.6. Year 2 Coursework results

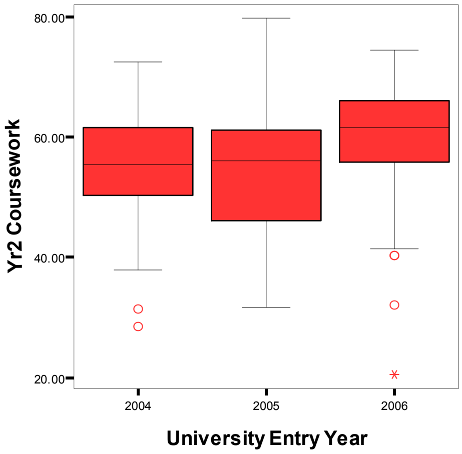

Figure 5.12: Year 2 coursework-mean by cohort

Year 2 coursework means by cohort in Figure 5.12 shows there is a statistically significant difference between 2004 and 2006 cohorts. Thus -

Results: Year 2 Coursework in:

- 2004 does not differ significantly from 2005: U=740.0, $p > .0167$, $r = -.05$ (small effect).
- 2004 differs significantly from 2006: U=636.0, $p < .0167$, $r = -.30$ (medium effect).
- 2005 does not differ significantly from 2006: U=487.0, $p > .0167$, $r = -.27$ although this is marginal ($p = 0.018$).

5.4.7. Year 1 and Year 2 Combined

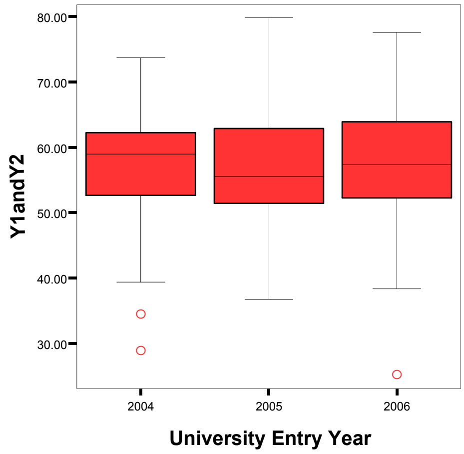

Figure 5.13: Year 1 and Year 2 combined-means

Figure 5.13 provides an overall picture of student means at the end of the two years of study for each of the three cohorts. A number of outliers are identified for cohorts 2004 and 2006 (shown by a circle in Figure 5.13). These outliers have been described in 4.4.1.

For 2005, the distribution is skewed, as the top quartile scores are spread out over a wider range than the bottom quartile. The median is higher for 2004 than for the following two cohorts. 2005 and 2006 have a similar median with the inter-quartile range (essentially the height of the box – spread of the middle 50% of the data) being wider for 2006 than 2005. There is no statistically significant difference between any pair of cohorts.

Results: Overall for Year 1 and Year 2 combined:

- 2004 does not differ significantly from 2005: U=703.50, $p > .0167$, $r = -.14$.
- 2004 does not differ significantly from 2006: $U = 941.00$, $p > .0167$, $r = -.08$.
- 2005 does not differ significantly from 2006: $U = 633.0$, $p > .0167$, $r = -.06$.

In each of these cases, the effect r is defined as small.

5.4.8. Summary of analyses of cohorts

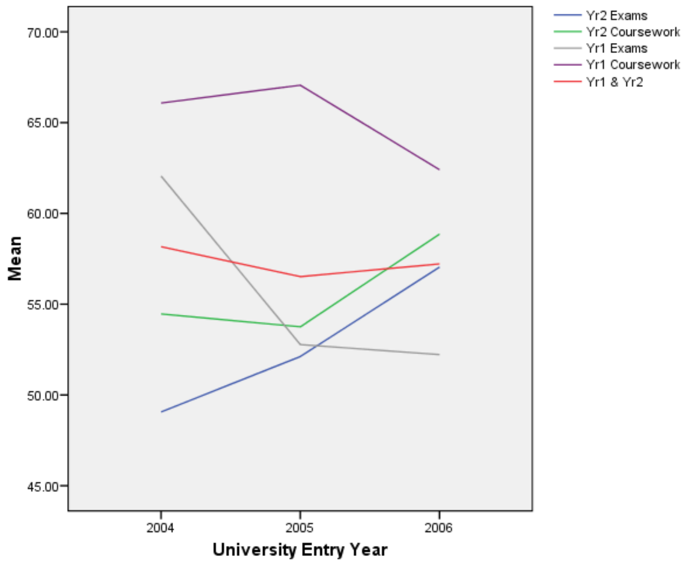

Figure 5.14: Line graph - summary of analysis of cohorts

Figure 5.14 presents an overview by cohort of the means for exams and coursework (values are shown rather than rankings). It also shows the combined Year 1 and Year 2 mean which is the combination of the two years of study.

Results of the means comparison for Year 1 and Year 2 show no significant difference between cohorts. 2004 does, however, have a smaller third quartile range (as shown in Figure 5.13) than subsequent years.

Further breakdown of the data reveals that Year 1 Exam performance declined from 2004 onwards. However, for Year 2 Exams, the opposite effect is shown in that there is an improvement in exam performance from 2004 onwards, with a significant difference in means between cohort 2004 and cohort 2006.

For Year 1 Modules (exam and coursework), a decline in performance is observed after the 2004 cohort. Year 2 Modules, however, show the opposite effect in that the overall performance of Year 2 Modules improves from 2004 onwards, with a significant difference being reported between 2004 and 2006. The Year 1 coursework and Year 2 coursework show there is no significant difference between the cohorts.

The Year 1 Exam-means get progressively worse from 2004, which results in a drop in Year 1 Module-means for subsequent cohorts. As Year 1 Coursework remains stable it is the exam performance which has caused the difference between cohorts.

5.5. Subject analysis

This section presents each A-level subject identified in Table 5.6 and provides the test results for each subject. Test results presented for each A-level subject are:

- Single subject: for each A-level subject, cohort statistics for Year 1 Module and Year 2 Module are given to identify if there are any significant differences between cohort-means.
- Combination of two subjects: this looks at determining if having a specific **combination** of subjects, irrespective of cohort, results in any significant differences.
- Subject v non-subject: test results for students taking or not taking each A-level subject are provided based on combined-cohort data.

For each A-level subject a descriptive table, for example Table 5.15, has been included to provide immediate general descriptive statistics about the population such as the Mean and Std Deviation. As described in the previous section (section 5.3) each A-level subject has associated investigation and procedural information included. Further details of these can be found in section 4.5, 4.6 and 4.7.

A full set of the tables which are generated by the statistical procedures have been included for the first of the A-level subjects under investigation (Maths) in the following section. However for clarity, tables have been excluded for subsequent subjects where the results are not significant.

5.5.1. Single Subject analysis

5.5.1.1. Mathematics

The following tables are comparisons between cohorts where students have Maths A-level.

5.5.1.1.1. Yr 1 Modules (Maths)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 6a	One-way ANOVA

Table 5.14: Investigation 2 – Maths -Year 1 Module

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
2004	29	67.4598	6.51763	1.21029	53.67	77.67
2005	14	62.1667	10.15668	2.71449	41.33	75.67
2006	25	55.9200	11.41769	2.28354	23.00	77.67
Total	68	62.1275	10.54282	1.27850	23.00	77.67

Table 5.15: Descriptive statistics for Year 1 cohorts – Maths A-level

Levene's statistic in Table 5.16 show $p > .05$, a non-significant statistic, indicating that the variances within each cohort are statistically the same.

Test of Homogeneity of Variances

Yr1 Modules

Levene Statistic	df1	df2	Sig.
1.772	2	65	.178

Table 5.16: Maths - Year 1 Modules Homogeneity of Variance results table

A significant value $p < .05$ (Table 5.17) indicates there is a difference between the means of the groups. Post Hoc tests (Table 5.18) provide a set of comparisons for each pair of years such as year 2004 compared to 2005 and then 2004 compared to

2006 etc. These comparisons reveal that the significant difference found is between cohort 2004 and cohort 2006 (indicated by an asterisk). Students in 2006 who had Maths A-level had a statistically significant lower mean (M = 55.92) for Year 1 Modules than those students with Maths in 2004 (M = 67.46).

ANOVA

Yr1 Modules

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1787.908	2	893.954	10.268	.000
Within Groups	5659.210	65	87.065		
Total	7447.118	67			

Table 5.17: Maths - Year 1 Modules ANOVA results table

Multiple Comparisons

Dependent Variable: Yr1 Modules
Scheffe

(I) University Entry Year	(J) University Entry Year	Mean Difference (I-J)	Std. Error	Sig.
2004	2005	5.29310	3.03663	.227
	2006	11.53977*	2.54653	.000
2005	2004	-5.29310	3.03663	.227
	2006	6.24667	3.11472	.142
2006	2004	-11.53977*	2.54653	.000
	2005	-6.24667	3.11472	.142

*. The mean difference is significant at the .05 level.

Table 5.18: Maths – Year 1 Modules Post Hoc test results

Result:

There is a significant difference in mean for having Maths A-level on Year 1 Modules between cohort 2004 and cohort 2006: $F(2,65) = 10.3$, $p < .05$, $r = .49$, which represents a medium to large effect size. The mean for 2004 was significantly higher than for subsequent years.

5.5.1.1.2. Year 2 Modules (Maths)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 6a	One-way ANOVA

Table 5.19: Investigation 2 – Maths - Year 2 Module

Result:

There was no significant difference in means for having math A-level on Year 2 Module between cohorts. $F(2,65) = 1.077$, $p > .05$, $r = .18$ (small effect)

5.5.1.2. Maths v Non-Maths – combined-cohorts

"Does having Maths A-level result in an improved performance in year one and year two compared with not having that A-level?"

5.5.1.2.1. Year 1 Modules (Maths)

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.20: Investigation 3 - Maths v Non-Maths (combined cohort) – Year 1 Module

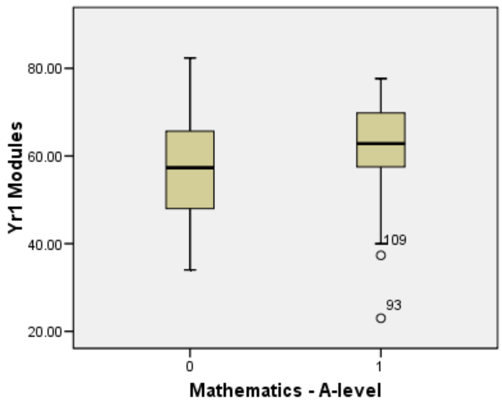

Figure 5.15: Non-Maths and Maths Yr 1 Modules – combined-cohorts

Key: 0 = Does not have subject, 1 = has subject

The unusually low value (outlier) in this data set (Figure 5.15) is clearly shown. Figure 5.15 shows skewness for Maths (1) (i.e. those students who have Maths A-level) as the mean is shifted away from the centre and the position of the box is skewed negatively indicating a bunching of scores to the higher end. This bunching of scores can also be seen in Figure 5.16 for Maths (1) whereas Maths (0) (i.e. those students who do not have Maths A-level) has a wider spread of means shown by a lower kurtosis (lower curve and with longer and more evenly distributed tails).

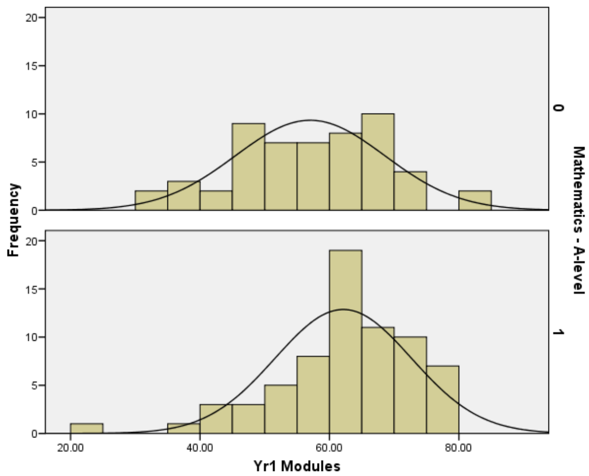

Figure 5.16: Histogram Non-Maths and Maths and Yr 1 Modules - all cohorts

The test results tables produced for the Mann-Whitney test are shown below (Table 5.21) for Year 1 Modules combined-cohorts.

Ranks

	Mathematics - A-level	N	Mean Rank	Sum of Ranks
Yr1 Modules	0	54	51.91	2803.00
	1	68	69.12	4700.00
	Total	122		

Test Statistics[a]

	Yr1 Modules
Mann-Whitney U	1318.000
Wilcoxon W	2803.000
Z	-2.670
Asymp. Sig. (2-tailed)	.008

a. Grouping Variable: Mathematics - A-level

Table 5.21: Non-Maths and Maths – Mann-Whitney results

Result:

There is a significant difference in Year 1 Module means for students with Maths compared with those students without Maths. The mean was higher for those with Maths: $U = 1318$, $p < .05$, $r = .24$

5.5.1.2.2. Year 2 Modules (Maths)

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 3b, 4b, 4bii	Independent t-test

Table 5.22: Investigation 2 - Maths v Non-Maths (combined cohorts) – Year 1 Module

Result:

There is no significant difference in means for students with Maths to those students without Maths: $t(120) = -1.568$, $p > .05$, $r = 0.14$
Maths mean = 56.20 and Non-Maths mean= 53.23.

5.5.1.3. Computing

The following tables are comparisons between cohorts where students have Computing A-level.

5.5.1.3.1. Yr 1 Modules (Computing)

Investigation	Procedural Steps	Statistical test
2	1d, 2a, 5a, 5b, 6c, 6d	Kruskal-Wallis, Mann-Whitney

Table 5.23: Investigation 2 - Computing- Year 1 Module

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
2004	21	63.0952	11.29363	2.46447	34.00	74.67
2005	19	59.6140	10.18245	2.33601	45.33	82.00
2006	22	59.7273	6.18809	1.31931	48.00	70.00
Total	62	60.8333	9.39475	1.19313	34.00	82.00

Table 5.24: Descriptive statistics for Year 1 cohorts – Comp A-level

Result:

There was no significant difference in means between cohorts for students with Computing A-level for Year 1 Modules: ($H(2) = 4.701$, $p > .05$)

Result of the Mann-Whitney Post hoc tests:

There is no significant difference between:

2004 and 2005: U=140, p >.0167, r = -.35 medium effect size
2004 and 2006: U=145, p > .0167, r = -.31 medium effect size
2005 and 2006: U=194, p > .0167, r = -.06 small effect size

5.5.1.3.2. Year 2 Modules (Computing)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 6a	One-way ANOVA

Table 5.25: Investigation 2 – Computing - Year 2 Modules

Result:

There is a significant difference in mean between cohorts for students with Computing A-level for Year 2 Modules: $F(2,59) = 7.171$, $p < 0.05$, $r = 0.44$.

Post hoc tests reveal this effect is between cohort 2004 and cohort 2006. Cohort 2006 has the highest mean of all cohorts.

5.5.1.4. Computing v Non-Computing (combined-cohorts)

"Does having Computing A-level result in an improved performance in year one and year two compared with not having that A-level?"

5.5.1.4.1. Year 1 Modules

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.26: Investigation 3 - Comp v Non-Comp (combine cohort) – Year 1 Module

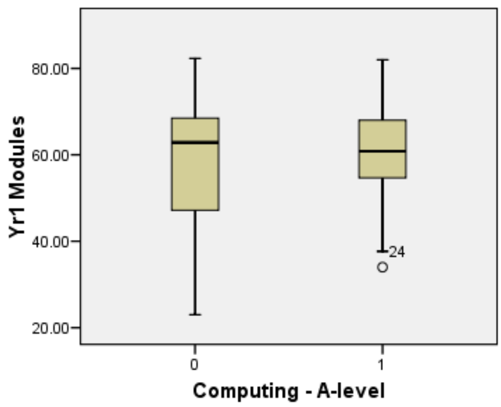

Figure 5.17: Non-Comp and Comp Year 1 Modules - combined-cohorts

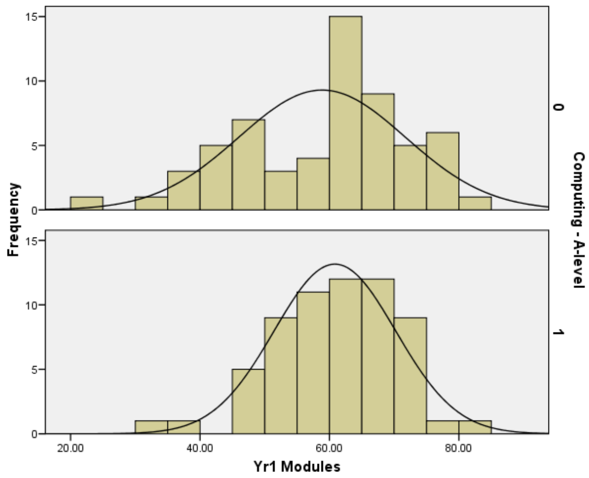

Figure 5.18: Histogram Non-Comp and Comp Yr 1 Modules – combined cohorts

Figure 5.17 and 5.18 clearly show skew towards the upper end of the score for non-Computing and with a much wider spread of scores than for Computing. Figure 5.18 shows a bi-modal distribution for non-Computing, whereas Computing shows a reasonably well distributed set of scores.

Result:

> There is no significant difference in Year 1 Module means for students with Computing compared with those students without Computing: $U = 1778.50$, $p > .025$, $r = -.03$

5.5.1.4.2. Year 2 Modules

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.27: Investigation 3 - Comp v Non-Comp (combined cohorts) – Year 2 Module

Result:

> There is no significant difference in Year 2 Module means for students with Computing compared with those students without Computing: $U = 1647.50$, $p > .025$, $r = -.10$

5.5.1.5. ICT

The following tables are comparisons between cohorts where students have ICT A-level.

5.5.1.5.1. Yr 1 Modules (ICT)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b,4a, 4aii, 6a	One-way ANOVA

Table 5.28: Investigation 2 - ICT- Year 1 Module

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
2004	13	63.8974	12.61054	3.49753	36.67	82.33
2005	9	55.9259	15.83226	5.27742	34.33	82.00
2006	10	56.2333	10.45159	3.30508	41.67	70.33
Total	32	59.2604	13.15900	2.32621	34.33	82.33

Table 5.29: Descriptive statistics for Year 1 cohorts – ICT A-level

Result:

There was no significant difference in means between cohorts for students with ICT A-level for Year 1 Modules: $F(2,29) = 1.395$, $p > .05$, $r = .30$

5.5.1.5.2. Year 2 Modules (ICT)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 6a	One-way ANOVA

Table 5.30: Investigation 2 - ICT- Year 2 Module

Result:

There was no significant difference in mean between cohorts for students with ICT A-level for Year 2 modules: $F(2,29) = 0.779$, $p > 0.05$. $r = .23$. Cohort 2006 has the highest mean of all cohorts.

5.5.1.6. ICT v Non-ICT (combined-cohorts)

"Does having ICT A-level result in an improved performance in year one and year two compared with not having that A-level?"

5.5.1.6.1. Year 1 Modules

Investigation	Statistical test	Procedural Steps
3	Mann-Whitney	1c, 1d, 2a, 5b, 6d

Table 5.31: Investigation 1 - ICT v Non-ICT (combine cohort) – Year 1 Module

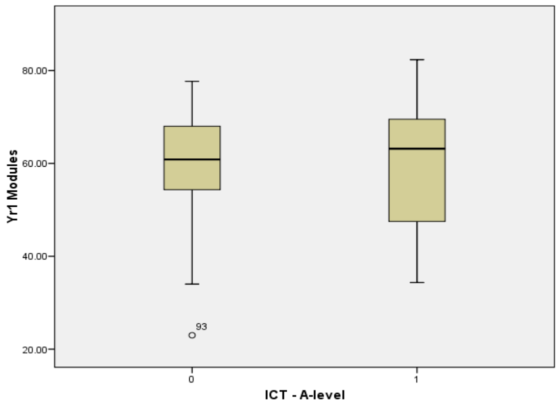

Figure 5.19: Non-ICT and ICT Year 1 Modules – combined-cohorts

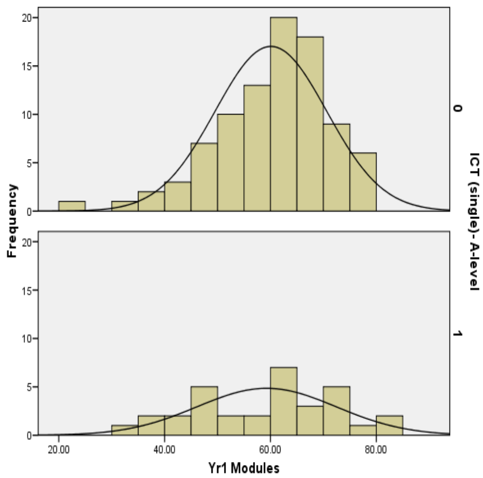

Figure 5.20: Histogram Non-ICT and ICT Yr 1 Modules - combined-cohorts

Figures 5.19 and 5.20 both show a positive skew for non-ICT students compared to ICT students

Result:

There is no significant difference in Year 1 Module means for students with ICT to those students without ICT: $U = 1418.50$, $p > .025$, $r = .01$

5.5.1.6.2. Year 2 Modules

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.32: Investigation 3 - ICT v Non-ICT (combine cohort) – Year 2 Module

Result:

There is no significant difference in Year 2 Module means for students with ICT to those students without ICT: $U = 1257.50$, $p > .025$, $r = .10$

5.5.1.7. Physics

The following tables are comparisons between cohorts where students have Physics A-level.

5.5.1.7.1. Yr 1 Modules (Physics)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 4aiii, 6a	One-way ANOVA

Table 5.33: Investigation 2 - Physics- Year 1 Module

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
2004	20	66.0500	10.21080	2.28321	37.67	82.33
2005	14	64.0714	10.09034	2.69676	45.33	82.00
2006	25	56.6800	12.35152	2.47030	23.00	77.67
Total	59	61.6102	11.78302	1.53402	23.00	82.33

Table 5.34: Descriptive statistics for Year 1 cohorts – Physics A-level

Result:

There was a significant difference in means for students with Physics A-level between cohorts: $F(2,56) = 4.368$, $p < 0.05$, $r = .37$.
Post hoc tests reveal this difference is between cohort 2004 and cohort 2006. Cohort 2004 has the highest mean of all cohorts.

5.5.1.7.2. Year 2 Modules (Physics)

Investigation	Procedural Steps	Statistical test
2	1d, 3, 3b, 4a, 4aii, 6a	One-way ANOVA

Table 5.35: Investigation 2 - Physics- Year 2 Module

Result:

There was no significant difference in means for students with Physics A-level: $F(2,56) = 0.883$, $p > 0.05$, $r = .17$. Cohort 2006 has the highest mean of all cohorts.

5.5.1.8. Physics v Non-Physics (combined-cohorts)

"Does having Physics A-level result in an improved performance in year one and year two compared with not having that A-level?"

5.5.1.8.1. Y1 Modules

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.36: Investigation 3 - Physics v Non-Physics (combined cohorts) – Yr 1 Module

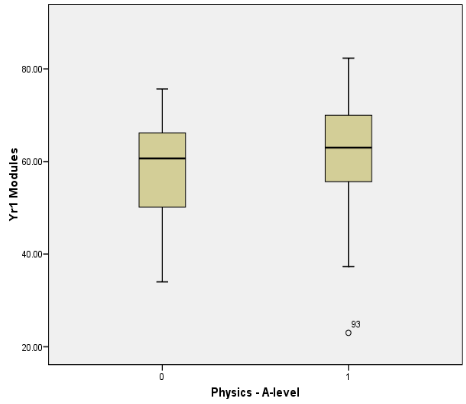

Figure 5.21: Non-Physics and Physics Year 1 Modules - all cohorts

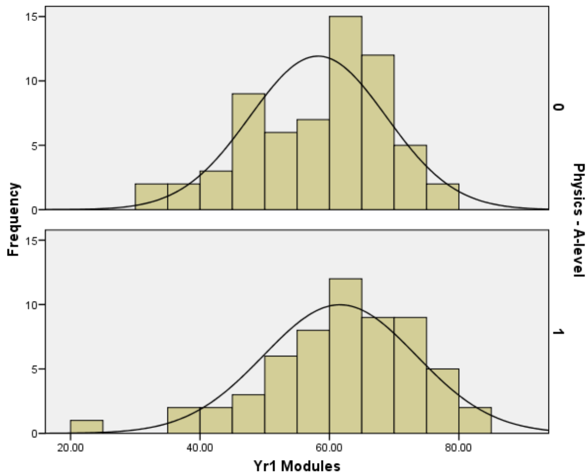

Figure 5.22 : Histogram Non-Physics and Physics Yr 1 Modules - all cohorts

Figure 5.21 and 5.22 show a more even distribution for Physics students in comparison to non-Physics where there is a bi-modal distribution.

Result:

There is no significant difference in Year 1 Module means for students with Physics compared with those students without Physics: $U = 1509$, $p > .025$, $r = .16$

5.5.1.8.2. Year 2 Modules

Investigation	Procedural Steps	Statistical test
3	1c, 1d, 3, 3a, 4b, 6b	Independent t-test (transformed data)

Table 5.37: Investigation 3 - Physics v Non-Physics (combine cohort) –Year 2 Module

Result:

There is no significant difference in Year 2 Module means for students with Physics to those students without Physics: $t(120) = -.888$, $p > .05$, $r = 0.08$

5.5.2. Summary for Single-Subject Analysis

Data analysis has been performed on means for Year 1 and Year 2 study based on a number of single A-level subjects. The analysis has taken two forms, firstly for each A-level subject identified in Table 5.6, a comparison has been made between cohorts to determine if there are any statistical differences. The second part of the subject analysis has taken the combined-cohort means and, for each subject, determined whether the presence or absence of a particular A-level has had an impact on the module means.

Single-subject analysis revealed that, between cohorts, there is no significant statistical difference in means, except for Year 1 Module Maths and Year 1 Module Physics, and, in each case, this was between the cohorts 2004 and 2006. For Year 1 Module Maths, the means are higher for 2004 than for subsequent years and, whilst this pattern of increase is the same for non-Maths cohorts, there is no statistical difference in the means for non-Maths cohorts. For Physics, the same decrease in means is shown from 2004 onwards and this pattern is replicated with significant differences for non-Physics cohorts. For Year 2 Module, there is no statistically significant difference between means, except for Computing, which resulted in a significant difference between cohorts 2004 and 2006. However, in this instance, the mean for 2006 was higher than 2004. There were no statistical differences for the Year 2 Module non-Computing cohorts.

For combined-cohorts, the comparisons, between those students who have particular subjects to those who haven't, revealed that the only significant differences were for the A-level Maths and Physics combination. For Year 1 Module, it was shown that the overall mean for this combination is significantly higher (and especially in 2004) than for Non-Maths and Physics students. For Year 2 Module, there is no difference between having this combination or not.

5.6. Combinations of A-level Subjects

The previous section looked for patterns of performance where students have a particular A-level subject, e.g. Maths. It could, however, be asked whether a particular combination of subjects complement each other and, if so, what these subjects are. The A-level subjects under investigation are those which are most common and statistically comparable, but this means that many other potentially complementary subjects are excluded from the analysis. However, based on the most common combinations of entry A-level subjects, a further set of investigations provide results for these subject combinations. These subjects have been identified in Table 5.7. Only those subject combinations that have been studied by 25 or more students are reported. Results for the pairing of these subjects do not take into account the third subject studied. The A-level combinations investigated are:

- Maths and Physics (43 students)
- Maths and Computing (29 students)
- Computing and Physics (25 students)

General descriptive statistics are provided for each subject combination in year one, e.g. Table 5.39. These statistics include individual cohort-means for each subject combination and for comparison, those students who have neither of the subjects. Combined 'Total' means for the combination and for the non-combination are provided. The following results are for each of the combination subjects.

5.6.1. Maths and Physics

Investigation	Procedural Steps	Statistical test
4	1d, 3, 3b, 4a, 4ai, 4b, 6a	One-way ANOVA

Table 5.38: Investigation 4 -Subject Combination - Maths and Physics

Descriptives

		N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
Yr1MathPhys	2004	15	67.5556	7.45533	1.92496	53.67	77.67
	2005	9	65.5926	8.06360	2.68787	51.00	75.67
	2006	19	54.3333	12.63300	2.89821	23.00	77.67
	Total	43	61.3023	11.79786	1.79916	23.00	77.67
Yr1NMathNPhys	2004	12	57.8333	13.51281	3.90081	34.00	74.67
	2005	15	53.6444	10.18516	2.62980	34.33	67.67
	2006	11	52.8182	7.45261	2.24705	41.67	64.00
	Total	38	54.7281	10.64126	1.72624	34.00	74.67

Table 5.39: Descriptive statistics for Math-Physics A-level combinations

Result:

There is a significant difference in means between cohorts for students with Maths and Physics A-levels for Year 1 Module ($F(2,40) = 8.032$, $p < .05$).

Further analysis reveals that these differences are between 2004 ($M = 67.55$, $SE = 1.9$) and 2006 ($M = 54.3$. SE 2.9) and they are significantly different: $t(32) = 3.6$, $p < .025$ $r = .6$, which is a large effect. Between 2005 ($M = 65.6$, $SE = 2.69$) and 2006, there is also a significant difference: $t(26) =2.44$, $p >$ .025, $r = .43$, which is a medium effect.

5.6.2. Maths and Computing

Investigation	Procedural Steps	Statistical test
4	1d, 3, 3a, 2a, 5a, 6c	Kruskal-Wallis

Table 5.40: Investigation 4 - Subject Combination - Maths and Computing

Descriptives

		N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
Yr1MathComp	2004	12	67.1111	5.42969	1.56742	57.33	74.67
	2005	6	61.7778	11.71261	4.78165	47.33	75.67
	2006	11	60.4242	4.59974	1.38687	54.00	69.33
	Total	29	63.4713	7.31953	1.35920	47.33	75.67
Yr1NMathNComp	2004	8	60.2500	13.74224	4.85862	36.67	82.33
	2005	7	49.9048	12.48067	4.71725	34.33	65.67
	2006	6	52.7222	11.54973	4.71516	41.67	70.33
	Total	21	54.6508	12.95054	2.82604	34.33	82.33

Table 5.41: Descriptive statistics for Math-Computing A-level combinations

Result:

There was no significant difference in means between cohorts for students with Maths and Computing A-level for Year 1 Modules ($H(2) = 5.850$, $p > .05$).

5.6.3. Computing and Physics

Investigation	Procedural Steps	Statistical test
4	1d, 2a, 5a, 6c	Kruskal-Wallis

Table 5. 42: Investigation 4 - Subject Combination - Computing and Physics

Descriptives

		N	Mean	Std. Deviation	Std. Error	Minimum	Maximum
Yr1CompPhys	2004	6	64.7222	13.45679	5.49371	37.67	74.33
	2005	7	63.8571	13.78789	5.21133	45.33	82.00
	2006	12	60.8889	6.74849	1.94812	50.33	70.00
	Total	25	62.6400	10.44869	2.08974	37.67	82.00
Yr1NCompNPhys	2004	11	63.6667	11.71703	3.53282	36.67	75.67
	2005	8	49.8750	13.19564	4.66536	34.33	69.67
	2006	7	51.9048	8.47374	3.20277	41.67	64.00
	Total	26	56.2564	12.78273	2.50690	34.33	75.67

Table 5.43: Descriptive statistics for Computing Physics A-level combinations

Result:

There was no significant difference in means between cohorts for students with Computing and Physics A-level for Year 1 Modules: ($H(2) = 1.758$, $p > .05$).

5.6.4. Summary of Combinations of A-level subjects

For students who had Maths and Computing or Physics and Computing combinations, there was no statistical difference in means between cohorts. However, for those who have Maths and Physics, there was a significant difference in mean between cohorts for Year 1 Module. The 2004 cohort had a considerably higher mean than subsequent cohorts and, in particular, 2006. Cohort 2005 was significantly different to 2006. Cohort 2006 had the highest number of students with Math-Physics combination but the lowest mean for Year 1 Module.

Comparing the subject combination with non-subject combination it can be seen that other than for 2004, the non-subject combinations generally have the lower means. All three subject combinations show a higher mean for the 2004 cohort which is followed by a decline in means. For the non-subject combination this pattern is replicated but only for the 2004 cohort.

5.7. Coursework category analysis (combined-cohorts)

All summatively assessed coursework undertaken by students in Year 1 and Year 2 has been categorised into ‘types’ of work such as ‘theoretical’, ‘programming’ etc. and students are expected to complete all of them. This categorisation has been discussed in 4.6.3. The following results presented are based on A-level subjects – Maths, Computing, ICT and Physics.

For each A-level subject, a pairing is used, for example Maths v Non-Maths, and for each category of coursework, the means for the pairing are compared to see if, for any category of coursework, there is a significant difference. Examples of this would be Maths v non-Maths students for the ‘theoretical’ category; or the ‘theoretical and reporting’ category etc.

Results are based on one A-level subject, e.g. Maths, at a time and do not take into account any second or third A-level subjects.

5.7.1. Maths compared to Non-Maths

Investigation	Procedural Steps	Statistical test
5	4b, 4bii, 6b	Independent *t*-test

Table 5.44: Investigation 5 - Maths v Non-Maths for categories of coursework

Table 5.45 and Table 5.46 provide an example of the result tables produced from an Independent *t*-test. These tables show the coursework categories and the associated means for Maths and Non-Maths. The Sig(2-tailed) column in Table 5.46 shows any significant differences in means, $p < .05$. To read Table 5.46 information is provided in 4.7 Step 4bi

Tables will not be shown for subsequent A-level subjects in this section, because of their large size.

Group Statistics

	Mathematics - A-level	N	Mean	Std. Deviation	Std. Error Mean
Theory	1	68	67.5368	10.42560	1.26429
	0	54	60.7716	14.20228	1.93269
TheoryRep	1	68	51.5956	16.81467	2.03908
	0	54	47.5000	17.01415	2.31533
Programming	1	68	59.7169	15.11047	1.83241
	0	54	52.3426	17.17620	2.33739
ProgRep	1	68	61.6699	10.91471	1.32360
	0	54	61.6029	10.53002	1.43295
RepEssay	1	68	64.0417	10.19809	1.23670
	0	54	63.9969	10.23239	1.39245

Table 5.45: Maths v Non-Maths – Group Statistics for *t*-tests: combined-cohorts

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means				
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
Theory	Equal variances assumed	1.563	.214	3.033	120	.003	6.76516	2.23072
	Equal variances not assumed			2.929	94.392	.004	6.76516	2.30948
TheoryRep	Equal variances assumed	.036	.850	1.329	120	.186	4.09559	3.08102
	Equal variances not assumed			1.327	113.2	.187	4.09559	3.08522
Programming	Equal variances assumed	2.574	.111	2.520	120	.013	7.37432	2.92655
	Equal variances not assumed			2.483	106.4	.015	7.37432	2.97004
ProgRep	Equal variances assumed	.334	.565	.034	120	.973	.06705	1.95883
	Equal variances not assumed			.034	115.5	.973	.06705	1.95071
RepEssay	Equal variances assumed	.001	.976	.024	120	.981	.04475	1.86163
	Equal variances not assumed			.024	113.7	.981	.04475	1.86235

Table 5.46: Maths v Non-Maths A-level – t-test results for categories: combined-cohorts

Results:

- Student means are higher overall in Theory coursework for students with Maths A-level (M = 67.54, SE = 1.26) compared with those without (M = 60.77, SE = 1.93). The difference is significant: $t(120) = 3.033$, $p < 0.05$, $r = 0.27$ which has a medium effect (near the threshold of 0.3 for medium effect).

- In Programming coursework, student means were also higher overall for those with Maths A-level (M = 59.72, SE =1.83) compared with those without (M = 52.34, SE = 2.34) $p < 0.05$, $r = 0.22$. However, the effect is small.

Further investigation found that having a Maths A-level has a positive effect on the mean for the Theory coursework in both Year 1 and Year 2. However, for the

Programming coursework, the small positive effect is shown only in Year 2 for the Maths students, not in Year 1.

There were no significant differences shown in the other categories of coursework.

5.7.2. Computing compared to Non-Computing

Investigation	Procedural Steps	Statistical test
5	4b, 4bii, 6b	Independent t-test

Table 5.47: Investigation 5 - Comp v Non-Comp for categories of coursework

Results:

- Student means are higher in Programming coursework when having Computing A-level ($M = 59.47$, $SE = 1.77$) than for students without Computing ($M = 53.33$, $SE = 2.35$), $p < 0.05$, $r = 1.9$. The difference appears in Year 1 but the effect is small. There was no significant difference between means for Programming coursework in Year 2.

- In Programming/Reporting coursework, student means are higher for students having Computing A-level ($M = 63.63$, $SE = 1.11$) than for students without Computing ($M = 5\ 9.59$, $SE = 1.56$), $p < 0.05$, $r = 0.19$. The difference appears in Year 1 but the effect is small. There was no significant difference with Programming/Reporting coursework in Year 2.

Having Computing A-level has a positive but small effect on Programming and Programming/Reporting (ProgRep) coursework.

There were no significant differences shown in the other categories of coursework.

5.7.3. ICT compared to Non-ICT

Investigation	Procedural Steps	Statistical test
5	4b, 4bii, 6b	Independent t-test

Table 5.48: Investigation 5 - ICT v Non-ICT for categories of coursework

Result:

There are no significant statistical differences in mean for any category of coursework for ICT. However, for those students with ICT, the overall Programming mean was lower ($M = 52.54$, $SE = 3.06$) than for those who didn't have ICT ($M = 57.84$, $SE = 1.68$), $p > 0.05$, $r = 0.14$. For Year 1 and Year 2 Programming there is no significant difference.

5.7.4. Physics compared to Non-Physics

Investigation	Procedural Steps	Statistical test
5	4b, 4bii, 6b	Independent t-test

Table 5.49: Investigation 5 - Physics v Non-Physics for categories of coursework

Result:

Having Physics A-level has a small positive effect overall for Programming coursework ($M = 59.72$, $SE = 2.17$) as opposed to without Physics ($M = 53.39$, $SE = 1.97$), $p < 0.05$, $r = 1.9$. The difference appears in Year 1 but, again, the effect is small $r = 0.20$

There were no significant differences shown in the other categories of coursework.

5.7.5. Summary of Coursework categories

Each of the A-level subjects has been investigated to determine if there are any differences in means for any category of coursework for students with and without that particular subject. There are a number of significant differences to report. Maths students perform better in theoretical coursework in both Year 1 and Year 2 than non-Maths students do and Maths students also do better in Year 2 programming. Computing students performed better in programming and programming/reporting in Year 1 than non-Computing students. There was a small positive difference for Physics students in Yr 1 programming. There were no statistical differences in any coursework means for ICT compared to non-ICT students.

5.8. Computing and ICT A-level syllabus comparison to Computer Science Module Syllabus

The Computer Science year one modules, Programming and Data Structures (PDS) and Computer Systems (CSys) have been identified through the mapping of the ICT and Computing syllabuses to the Computer Science year one syllabus (described in section 4.2.1). Each of these modules are further sub-divided to determine if having a particular A-level, e.g. Computing, improved the coursework marks for particular topic areas within these modules, when compared with the marks for those not having that A-level.

Investigation	Procedural Steps	Statistical test
7	1a, 1d, 2a, 5a, 6c	Kruskal-Wallis

Table 5.50: PDS and CSys coursework comparisons for Computing and ICT

Results: Computer Systems

- There is a significant difference for the Computer Systems Machine Architecture assignment between Computing students and non-Computing students, with Computing students performing better: $H(1) = 5.80, p <.05$. Otherwise, the assignments for Operating System, Databases and Networks, show there is no significant difference in having Computing A-level.

- There is no significant different for the Computer Systems assignments between ICT and non-ICT students.

Results: Programming and Data Structures

- There is a significant difference for one of the Programming and Data Structures bench tests (timed practical exam) assignment between Computing students and non-Computing students: $H(1) = 7.65, p < .05$.

- There is no significant different for the Programming and Data Structures assignments between ICT and non-ICT students.

5.8.1. Summary of Comparison between Computing and ICT syllabuses and the Computer Science syllabus

Year one assignments had been identified as relating to different parts of the overlap between the A-level Computing and ICT syllabuses and the year one modules. For none of these assignments, was there a significant difference between the means for the ICT and the non-ICT students. However, for A-level Computing students v non-Computing students, there is a positive significant difference for the A-level Computing students in the Computer Systems Machine Architecture assignment. There is also a positive difference for A-level Computing students for a Programming and Data Structures assignment on data structures. A point to note is that this Programming and Data Structures bench test was one of four timed bench tests, two of which were Java programming exercises and the other two were on data structures.

5.9. Computing and ICT A-level by exam board (combined-cohorts)

This section provides the results for comparisons of means for both A-level Computing and ICT students by the exam boards they studied. Table 5.51 and Table 5.52 provide the number of students for each of these subjects and the associated examination board used. The mean and standard deviation are also provided.

Table 5.51 shows that only one student took A-level Computing with Edexcel, therefore making any statistical analysis unfeasible and, so, further analysis for A-level Computing has been for AQA and OCR exam boards only.

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation
AQA	18	59.3333	13.48298
Edexcel	9	54.7407	13.31330
OCR	6	68.3889	9.08825
Total	33	59.7273	13.22651

Table 5.51: ICT Exam board descriptives

Descriptives

Yr1 Modules

	N	Mean	Std. Deviation
AQA	48	60.5347	9.25314
Edexcel	1	53.6667	.
OCR	12	61.4722	10.03575
Total	61	60.6066	9.30004

Table 5.52: Comp Exam board descriptives

This following investigation will show if there is any difference in mean for Year 1 Modules for A-level Computing and ICT students depending on which exam board is used.

5.9.1. ICT exam boards (three exam boards)

Investigation	Procedural Steps	Statistical test
6	1b, 1c, 1d, 3b, 4a, 4aii, 6a	One-Way ANOVA

Table 5.53: Investigation 6 – ICT A-level Exam Board comparison

Result:

For Year 1 Module there was no significant difference between cohort-means for those students who studied A-level ICT with AQA, Edexcel or OCR: $F(2,30) = 2.063$, $p > .05$, $r = .17$

5.9.2. Computing exam boards (2 exam boards)

Investigation	Procedural Steps	Statistical test
6	1a, 1c, 1d, 2a, 5b, 6d	Mann-Whitney

Table 5.54 Investigation 6 –Computing A-level Exam Board comparison

Result:

For Year 1 Module, there was no significant difference between cohort-means for those students who studied A-level Computing with AQA or OCR: $U = 279$, $p > .05$, $r = -.02$.

5.9.3. Summary of Exam Boards

The means for Year 1 Module were calculated for each group of students who had taken A-level ICT with a particular exam board and no significant differences were found between them. The results were similarly non-significant for A-level Computing students.

This result shows that the choice of exam board for A-level Computing or ICT does not seem to make any difference to performance in year one. A description of the assessment methods of each exam board and the syllabuses revealed very little difference in the delivery of these subjects and the results presented here support this. Discussion of the exam boards is provided in section 3.5.1.

5.10. Questionnaire: Student perception of Computing and ICT A-levels

The results presented so far in this chapter have focused solely on the statistical analysis of archived student assessment marks. The purpose of the questionnaire, which was directed only at students who had A-level ICT or Computing, was to solicit the perception from each student as to where they were able to apply this prior knowledge to their first year Computer Science degree programme. It is expected that the student perceptions of their having certain prior knowledge had helped them in their first year is reflected in the statistical results presented above (section 5.8) for Computing or ICT.

Further information on the questionnaire can be found in 4.9.

5.10.1. Results of the questionnaire

In total, there were 57 respondents from a possible 89 (64% response rate):

- 6 respondents had both ICT and Computing
- 21 respondents for ICT
- 30 respondents for Computing

The responses from each of these groups of students have been analysed with the focus being on whether they perceived that A-level ICT or Computing was good preparation for year one core modules.

Figure 5.23 shows the combined responses for both A-level Computing and ICT students. It can be clearly seen that both subjects provide very little preparation for Formal Aspects (maths module). Similarly approximately 38% of responses state that they are not well prepared for Programming and Data Structures.

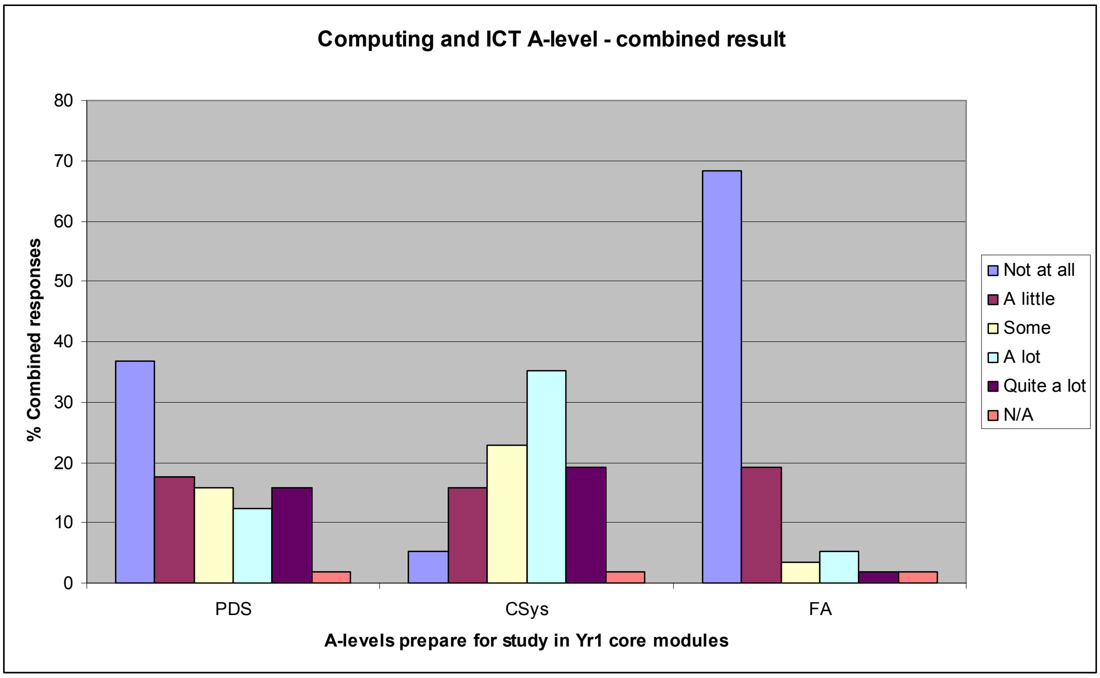

Figure 5.23: Combined responses for Computing, ICT and Computing/ICT

Key: Programming and Data Structures (PDS); Computer Systems (CSys), Formal Aspects of Computer Science (FA)

Figure 5.24 shows the responses by each A-level group of students where, for example Comp-PDS is the Computing A-level student responses for the PDS module, etc.

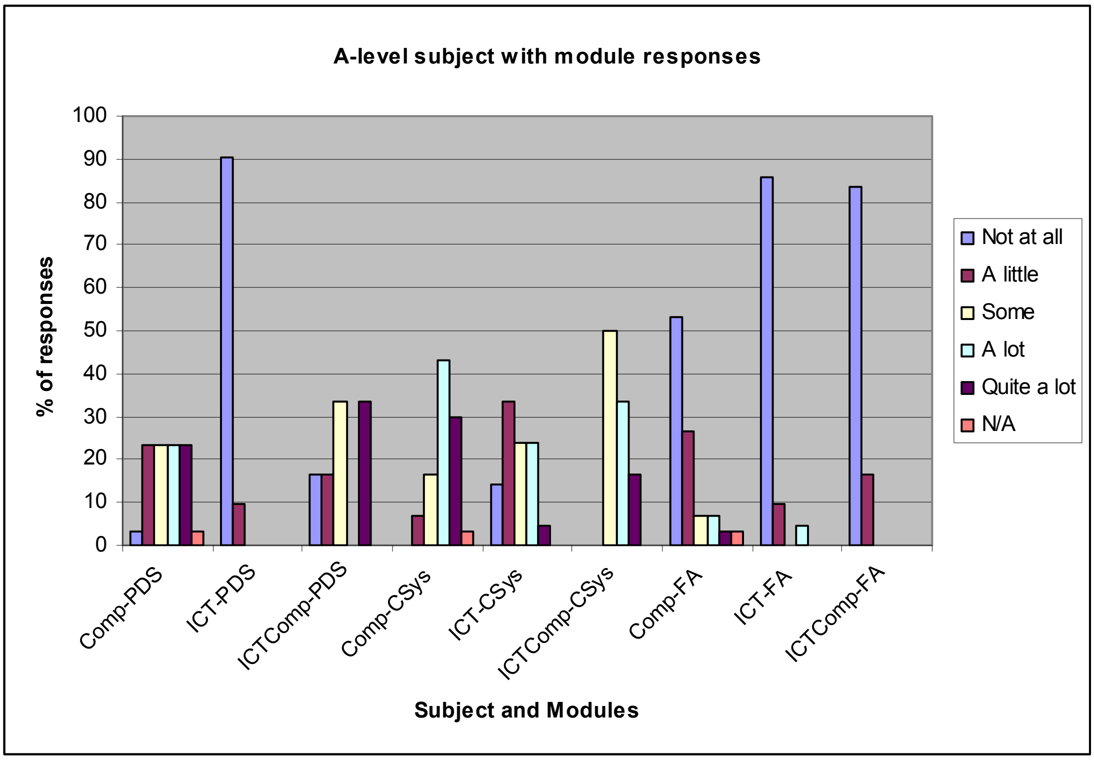

Figure 5.24: Responses for A-level subject with Year 1 Module

Figure 5.24 shows a breakdown for the specific A-level subjects and the associated student responses for the Year 1 core modules.

For those students who had A-level ICT, 90% felt that this subject had not helped them with PDS. For FA it was very similar, 85.7%. However, nearly 50% stated that for CSys, ICT has helped "some".

Students who had A-level Computing felt that they where reasonably prepared ("quite a lot", "a lot") for PDS (46. 6%) and CSys ("quite a lot" and "a lot")(53.4%) but 53.3% did not feel that Computing had helped at all for FA.

Two thirds of students who had taken both Computing and ICT felt that these had helped "some" or "quite a lot" with PDS but a significant majority, 83.3% felt that they had helped "not at all" for FA.

The overall results reveal a lack of preparation for FA and, in particular, from those students who had A-level ICT. However, approximately 50% of all respondents felt that ICT or Computing has helped for CSys preparation.

Identifying what was the prior knowledge gained was provided through the free-text comments:

- PDS: for Computing students, approaches to programming in general were known with various levels of experience in using programming languages such as VB or Pascal identified. Data structures such as stacks, binary trees and search algorithms had been covered at A-level. For ICT students, they perceived that they had done virtually no programming and it was, therefore generally felt that ICT was no preparation for PDS.

- CSys: Computing and ICT syllabuses both covered databases and basic networking to some degree, with students commenting on their familiarity with topics, especially in databases. Computing students felt more familiar with machine architecture, including the fetch-exe cycle, simple theory, binary numbers and operating system concepts.

- FA: The Computing and ICT syllabuses did not cover any material presented in FA. However, one student did comment that he had used some graph theory in his Computing coursework project at school. Some students did comment that they had also done A-level Maths and so were familiar with FA topics through prior knowledge gained in Maths studies but not in Computing.

An important question on the questionnaire was "Overall do you feel that your prior studies in ICT or Computing at school have helped in your studies here in the Department?" (Q7). Table 5.55 presents the overall results for this question where it can be seen that Computing, either as a single subject or combined with ICT, would appear to be perceived as better preparation for Year 1 studies than ICT.

A-level	Helped YES %	Helped NO %
Computing	86.7	13.3
ICT	38.1	61.9
Computing and ICT	100	0

Table 5.55: Overall results for prior studies helping in further study

Free text comments associated with question (Q7) revealed that ICT students felt that ICT was aimed at giving an overview of the field of computing in the context of business and the application of IT. Other than some basic database work and a familiarity with some terminology, these students felt they were not prepared and this was especially so for those who lacked any programming experience. In contrast, the Computing students felt that Computing gave them a good grounding in basic computing concepts, key terms and 'number theory' which was a solid foundation to build on. Specifics they learnt which helped with further studies included topics such as linked lists, basic logic, databases and networks. Having had some programming experience, irrespective of programming language, was seen as an advantage even though the course quickly became more difficult with the introduction of Object Oriented Programming which reduced "their edge".

5.10.2. Summary of the Questionnaire

Neither Computing nor ICT as individual subjects provide any preparation for FA. However, some students did comment that they were also taking A-level Maths which negated this.

The Computer Systems module syllabus has overlaps with both A-level Computing and, to a lesser degree, ICT. This overlap was in databases, networks, operating systems and machine architecture. However, for Computing students, the only topic in this module that showed a statistical advantage was for Machine Architecture (section 5.7). Both Computing and ICT students said they had experience of databases. However, this was not reflected in any results presented in earlier sections of the analysis (section 5.7).

Computing students had prior knowledge of a number of topics covered in PDS such as an introduction to programming, linked lists etc., which was not apparent in ICT. Results section 5.7.2 showed that programming coursework performance was better for those having Computing compared than those who didn't.

5.11. Chapter summary

This chapter has undertaken a number of investigations which have determined initially if there are any noteworthy differences between the cohorts in this study. What is interesting is the sizeable drop in year one exam performance and, in particular, from 2004 to 2005 (Figure 5.14). The A-level grade point average for 2004 is lower than 2005 but, in 2004, there was a greater percentage of students who had Maths and/or Physics compared to 2005. This, in itself, could explain the differences in these means. However, 2006 had a similar student A-level profile to that of 2004 and yet the exam means for 2006 were lower again (but not significantly) than 2005. This finding would imply that other factors are involved, rather than just the students' A-level profile. It had been noted and discussed in the department in 2005 that coursework was being marked high, mainly by relatively new international staff. It is, therefore, possible that, when marking exam scripts, staff erred on the side of caution and marked 'harder' in 2005 or simply had too high an expectation of the students in the exams.

There was a significant positive difference in mean for students with Maths and Physics compared to those who do not have this combination. This difference is once again related to Year 1 performance for the 2004 cohort.

Secondly, through a series of tests on combined-cohort data it has been determined that Maths students perform better in theoretical coursework in both Year 1 and Year

2, and programming in Year 2 compared to those who don't have Maths. Both Computing and Physics students do better in programming than non-Computing or non-Physics students. However, as the percentage of students having combined Maths and Physics is higher than any other combination of A-levels and especially for 2004 and 2006, the findings above are not surprising.

The 2005 cohort student A-level profile is quite different from that of 2004 and 2006 in that it has the lowest percentage of students having Maths and/or Physics. This has resulted in the lowest (although this is marginal) performance overall at the end of the two years of study Figure 5.14.

Finally, analysis from the Computing and ICT student questionnaire has shown that Computing or ICT does not prepare students for the Maths content of the Computer Science programme. However, there is overlap in some topics such as databases, networks etc. which students perceive as affording them a slight advantage in year one.

Chapter 6. Discussion

This chapter will compare three different studies that relate to Computer Science students and their previous studies. This thesis is compared against the other two studies to draw common conclusions and highlight the contribution which this thesis has made to the research area. Suggestions for the A-level Computing curriculum are then made. The chapter concludes with a discussion, based on the findings in this research, of the implications both for Learning and Teaching and for Admissions policies.

6.1. Comparison of the three studies

There have been a number of studies published by researchers which take different approaches in trying to determine if pre-entry qualifications are an indicator of success at university. For Computer Science in particular, the two approaches which are included for discussion and comparison within this chapter are, first, an ‘international’ (Alexander, Martyn et al. 2003) study and, secondly, a ‘national’ study (Boyle, Carter et al. 2002). This thesis is a ‘local’ study and relates to students from a single UK University and, thereby, it complements and adds a different dimension to the conclusions drawn from the previous two. The remainder of this section will describe and discuss both the ‘international’ and ‘national’ studies and will include an overview of their conclusions. A comparison will be made between their results and the results from this thesis.

6.1.1. International study

The study by Alexander (Alexander, Martyn et al. 2003) involved seven countries worldwide and focused on qualifications on entry, subjects studied in the early university curriculum and the subsequent grades achieved.

The approach taken in this study required the gathering and comparison of data for each ‘case study’ (country). This included admission requirements from recruitment information, the structure of degree programmes as well as actual pre-university qualification grades and marks from the first year of study at university. As there was a diverse range of entry qualifications, the survey derived numeric scores to represent these qualifications. Interpretive and quantitative approaches were used to analyse the data.

The ‘international’ study did not consider what subjects, e.g. Maths or Computing, had been taken by the students but did identify common areas such as programming and Discrete Maths in the first year study for each country. Results highlighted the fact that school achievements, that is subjects and associated grades, did not indicate which students would be successful in the studying of programming. However, good entry grades in Maths, and specifically ‘traditional’ Maths compared with Discrete Maths or Logic (which are often introduced at university level), may indicate that they are likely to do well in the Mathematical part of the university curriculum.

Tentative conclusions drawn from this study found that, in general, overall entry scores, from whichever country, were poor predictors of how students would perform in Computer Science. This study also concluded that, if students had prior programming experience gained from “outside of school”, they would have an advantage over those students who did not have this experience. This additional experience is likely to be the result of the student’s motivation and general interest in the subject. The student may not even have studied programming in school. It may even be seen as more of a ‘hobby’ than a school subject.

Alexander et al. recognised the problems associated with such international variability, for example the rules for progression from year one to year two, and the number of years of study required within each degree programme. In the UK, a degree programme usually lasts three years whilst in other countries this can be five years or more. When comparing year one performances, the difficulty arises in that the UK year one is a larger proportion of the overall study period (i.e. one third) and, as such, has to cover a large amount of core disciplinary material. In other countries, this material may not appear until year two or year three and, therefore, factors such as this need to be taken into consideration when interpreting the results for year one performance across universities.

6.1.2. National study

On a national level, a comparison has been made between two UK universities (Boyle, Carter et al. 2002). One of the focuses of this study was on entry subjects and grades as indicators of year one performance and final success (graduation), with emphasis on A-level Computing and Maths. The study involved an arbitrarily chosen cohort of single honour students from each of the universities. The universities are of different types and size but do have comparable programmes of study and a similar student profile, e.g. percentage of non-traditional students.

Conclusions drawn from this study are that there is no correlation between performance, specifically the degree classification, and any of the measures such as entry subjects or grades. Similarly, performance in year one was not a predictor of the final year outcome. Discipline-specific subjects, such as Computing, were found to be “irrelevant” and there was no distinction in performance in year one between having and not having Maths.

Interestingly in this ‘national’ study it was found that non-traditional entry students, which made up approximately 30% of the cohorts, performed equally well as the traditional A-level students. Non-traditional backgrounds are classed as students who are mature or who have BTEC, Access or Foundation Programme qualifications. If these students do equally well as traditional A-level entry students then "the insistence of certain subjects and grades for entry to Computer Science is called into question". (Boyle, Carter et al. 2002)

The data used in this study does contain anomalies, such as 'sandwich' students who return to study after a year out. With their new knowledge and experiences, compared to non-sandwich students, the study does recognise that these students can do disproportionately well.

6.1.3. Local study

While the work in this thesis is not too dissimilar to the aims and approaches of the studies previously described, in order to differentiate from these other studies this work will be referred to as the 'local' study.

Where the 'local' study approach is different from the others is in the unique data set and the student cohorts which were available for analysis. Three consecutive cohorts of single honour students from the same university (Durham), restricted to only those with A-level entry qualifications, have been used. These students all undertook essentially the same Computer Science programme, with a fixed core syllabus in year one and year two. They were all exposed to the same progression rules and were all taught essentially the same topics by the same lecturers. This uniformity removes quite a lot of the variability in the data compared to the 'international' and 'national' studies. For example, in the 'local' study, non-traditional students are too few in number for statistical analysis and, therefore, have been removed from the local data set altogether.

The discussion of results from statistical and qualitative analysis has been provided in more detail in the previous chapter. It is, however, useful at this point to summarise some of the more general ‘local’ findings so as to compare them with conclusions drawn in the ‘international’ and ‘national’ studies.

Where the main similarities in results appear is between the ‘local’ and ‘international’ studies. This in itself is quite interesting as these two studies are perhaps the most disparate in the approach used. The ‘local’ study found a statistically significant difference in the combined-cohort mean at the end of year one. Students who had A-level Maths achieved a higher mean overall (when compared with non-Maths students). This difference is however statistically small and there were no statistical differences by the end of year two.

At a lower level of granularity, the ‘local’ study also found that, perhaps not surprisingly, the Maths students do better in the more theoretical coursework in both year one and year two. This is similar to the conclusion drawn from the ‘international’ study. Conversely, the ‘national’ study found that having Maths A-level does not influence performance in year one. This ‘national’ finding is, however, discussed further in Boyle (Boyle and Clark 2002) where further work, using one cohort of students from one institution, found that, for the obvious areas such as algorithms and maths, there was a marked advantage in having Maths A-level.

In all three studies, a general consensus is that Maths A-level is certainly advantageous to students in first year, but only in certain areas of the syllabus. Results from the ‘local’ study found that, by the time the student cohorts reached the end of year two, there was no correlation between mean performance and the A-level subjects (or at least those for which sufficient data was available) that the students arrived with. It can, therefore, be concluded reasonably safely that A-level subjects do not indicate how a student will perform overall in their studies.

Contributory factors for success in Computer Science include the approaches to learning a student brings with them and their prior subject-specific knowledge. In the ‘international’, ‘national’ and ‘local’ approaches, all authors have acknowledged the lack of measurement with regard to student motivation, prior educational background, self-organisation and other personality traits which can contribute hugely to the success or not of a student’s performance at university.

6.2. A-level Computing and ICT

A-level Computing, in the ‘local’ and ‘national’ studies, and ICT, in the ‘local’ study, do not impact on performance in year one or year two and, whilst they do not improve marks, these subjects are not detrimental.

The ‘local’ analysis of the content of A-level Computing and ICT subjects by exam board showed that there are no discernible differences between what is being offered by one subject exam board, or the nature of its assessment, and that of another. Students’ performance is not, therefore, affected by which exam board they studied under.

Boyle (Boyle, Carter et al. 2002) states there is a scepticism in universities about the value of the Computing A-level because of the variability of facilities and teaching offered by each school and that it is, perhaps, an ‘easier’ subject as “students score very highly in this subject on the basis of project work”. Table 3.3 in this thesis has found that, certainly for the Computing A-level, the exam/coursework ratio is 60/40. This is, however, very similar to the ratio used for exam and coursework assessment in Computer Science at Durham and so from this perspective, the A-level work is providing some preparation in terms of assessment balance.

6.2.1. Student perceptions and programming experience

The perception of the ‘local’ students who have A-level ICT and more specifically Computing, is that this has initially helped in some of their year one studies, primarily in the Computer Systems module. It could be expected, therefore, that these students would perform better in certain areas where they have subject-specific knowledge. However, the statistical analysis has shown that, even where there is reasonable overlap of topics between these subjects and the Computer Science syllabus, there was no significant difference in year one mean between those who had these A-level subjects and those who didn't. The analysis found no evidence that this knowledge had given them an overall advantage. The 'local' result did, however, show that those students with Computing A-level did better in programming coursework than those without. There is, however, limited provision for gaining programming experience within the A-level Computing curriculum and even less in ICT. Programming experience within A-level Computing is usually restricted to one programming language and this is often a language such as Visual Basic or PHP (Drummond and Jamieson 2005). The variability of languages taught across schools is considerable but most students seem to opt to use Visual Basic or Microsoft Access for their project work.

In the 'local' study, the student questionnaire revealed that prior experience gained solely through A-level Computing, without any extra-curricular computing, afforded a short-lived advantage to these students in comparison to complete novice programmers. However, this 'programming' advantage, when further investigated, revealed that A-level Computing students (as opposed to non-Computing students) performed better in only one of four assignments in the programming module (PDS). Perhaps surprisingly, this was not the Java programming component but work on data structures. The first Java assessed coursework test is normally held half-way through the first term. This would suggest that any advantage A-level Computing students may have had, had already vanished.

A Computing A-level student's programming ability and ultimate success within that element of year one would seem to be dependent on prior programming experience gained other than from school. This supports the findings of the 'international' study. Work by Hagan (Hagan and Markham 2000) goes further in that the results showed that, for students with programming experience starting their Computer Science degrees, there was a significant advantage if a student had studied or used a number of programming languages.

The question of how much prior programming experience students had before they came to university was restricted in the 'local' study to those students who had A-level Computing or ICT. Question 6 in the questionnaire asked "Had you any experience in ICT or Computing (other than from school or college) before coming to University?" Answers which were applicable to prior programming experience (as opposed to any wider computing experience) included:

- "*I've been programming 5 years, 2 of which in a paid capacity….."*
- "*Programming since I was 13 …"*
- *"Yes – Programming in Visual Basic to a high(?) level"*.

When asked if prior experience had helped in their studies in Computer Science (question 6a), student responses included:

- *"Learning to program ahead of time helps a lot"*
- *"Before arriving at university, I was fluent in 2 programming languages, was knowledgeable about object oriented design (including Design Patterns)…"*
- *"I taught myself to program in C# in Year 10 so by the time I came here I was able to focus on the mathematical elements of the course straight away without worrying if I could program"*.

From these student comments, it can be assumed that these students have been motivated to learn to program pre-university and value the understanding and skill that this has provided. Learning to program requires an understanding which is not simply the memorising of factual knowledge; an approach which can be successful with some A-level material. Student motivation is crucial and it has been identified as a ‘threshold concept’ in learning to program (Drummond and Jamieson 2005). Ausubel (Ausubel, Novak et al. 1968) states that “Motivation although not indispensable for limited and short-term learning is absolutely necessary for … mastery of a given subject”. Learning to program is, for many students, not a skill that can be learned in the short-term or ‘surface’ learned. It is a skill where this ‘mastery’ or, at the very least, competence, is necessary for progression in a Computer Science programme.

6.2.2. Implications for A-level Computing

Currently, A-level Computing is not held in high esteem by academics in Computer Science. However, one opportunity to change this perception is for the A-level Computing curriculum to incorporate and place more emphasis on the teaching of programming. Having mathematical ability has been shown to be advantageous to students in year one. Therefore, if A-level Computing provided students with a more rigorous programming experience within the syllabus, it could only serve to prepare them more adequately for what is expected in year one. A down-side of any increase in programming could be that schools may move away, because of a lack of expertise in their teachers to teach programming, from Computing and towards ICT.

ICT A-level is held in perhaps even lower esteem, partly because of an almost total lack of programming and its focus on business and the use of applications. If Computing rises in esteem, it is likely that ICT will fall further in esteem. What is almost impossible to gauge is how far any ‘improved’ Computing A-level might attract more students to want to study Computer Science at university.

6.3. Implications for Learning and Teaching

It would not be an unreasonable assumption that students who come with subject-specific, prior knowledge should have an advantage in certain aspects of their studies over those students who do not. In Computer Science at Durham, there is a range of student abilities as no one specific subject was required for entry. Therefore, to address this disparity, an aim of the first year of study is to ensure that all students are at, or above, a level that allows them to proceed to year two.

In year one, it has been assumed that students have no specific knowledge of computing and the teaching is, therefore, done accordingly. It can be difficult to keep students who have Computing and ICT A-level engaged in topics, for example databases, they have already covered at school. In contrast to this, there are students who have none of this foundational knowledge. It is necessary to challenge and motivate both of these groups of students. The course material and its presentation can be tailored to address the disparity of knowledge. This should also involve the learners in accepting responsibility for their own learning and achievement.

6.3.1. Personalised learning

Whilst the overlap has been identified between the Computing A-level and the Computer Science syllabus, it does not guarantee the students' engagement with, or knowledge of, particular topics taught at school.

The results of the 'local' study revealed that there are no statistically significant differences in mean between A-level Computing and non-Computing students, in particular, for the coursework associated with the Computer Systems module. Informal discussions with students have led to insights that many students may be bored with course content because they have covered some of the material at school irrespective of whether they had gained an understanding or not . It can be very difficult to motivate a student to 're-do' material – irrespective of how successful they were first time round.

Diagnostic testing at the beginning of a module in year one could be undertaken to determine their understanding of the subject-specific knowledge. This testing would provide students with the opportunity to show competency in a particular topic, e.g. networks, and, once their level of competency has been established, would allow the students to be divided into those who may be required to cover this material again and those who would be exempt from re-studying the work and would be provided with new challenges in the form of advanced material.

With the support of, e.g. a virtual learning environment, adaptive releases of course material can be provided for each student. For example, if a formative test resulted in the student's achieving 60% or above, they would then be presented with more challenging material. For those scoring less, they would be expected to work through extra support material provided on-line or given reference to certain texts, before retaking the test.

A strength of virtual learning environments is that, if a student fails to grasp a concept, which is shown by failure in particular questions within a test, descriptions or definitions can be presented in an alternative way. By introducing this scaffolding (section 2.5) students can concentrate on completing those parts of the exercise that are within their range of competence before moving on.

This adaptive release can be used for both groups of students described earlier. For students with no prior knowledge of a topic, core concepts or techniques can be presented in a variety of forms using different media such as podcasts, audio etc. The students would be expected to work through this information in their own time. For example in Databases, a technique commonly used is Normalisation. This is a step-by-step technique and students would be expected to familiarise themselves with this ahead of the lectures on the topic. The lecture and associated practicals could then be used to reinforce the use of such techniques based on the fact that 'all' students share the same basic understanding of Normalisation. For those students who have prior knowledge, they can forgo, if they wish, pre-study of this material. For both groups, the use of formative testing allows the student to progress, with material being released as appropriate, and allows the teacher to determine if concepts are being understood. This is a process of the personalisation of timing as well as content.

Learning to program is a fundamental skill that Computer Science students must acquire and is one of the core building blocks for progression in the programme. Personalising the learning environment for students is to a large extent already in place in Durham in that students work through blocks of exercises at their own pace. This allows the novice programmers to build their confidence without holding back the more experienced programmers who can be challenged with a variety of more difficult exercises. Providing these learning blocks or ‘learning objects’ which are small, self-contained chunks of work allows the student to make progress in stages suitable for them (within limits imposed by assessment requirements and deadlines) with the intension that this builds their academic self-confidence (one of the five approaches defined by Tait and Entwistle in section 2.3.4) and motivation to learn.

6.4. Implications for Admissions

In the research described in the ‘local’ and ‘national’ sections, a common conclusion reached, supported by statistical analysis, is that there is no correlation between A-level subjects and overall first year performance for Computer Science students. A-level subjects therefore, are not good indicators of student potential. On the other hand, gaining three A-levels to a satisfactory level can indicate a student’s commitment and motivation to study. Despite this, Computer Science departments often mandate Maths as a pre-requisite qualification as it is seen as a necessary foundational subject for Computer Science.

Whilst there are aspects of Computer Science which require mathematical competence, the discipline is more than just Maths and this has been reflected in the results of the ‘local’ study which reveal that students who are mathematically qualified do have an advantage over their counter-parts in the more theoretical components of the degree but not in the overall achievement at the end of year two.

Computer Science departments are, therefore, potentially turning away good students who do not have Maths A-level. Admissions tutors need to be aware of the outcome of the research done at 'local', at 'national' and at 'international' level in which there is general agreement that pre-university subjects alone are not good indicators of a student's potential.

6.5. Summary

Comparisons have been made within this thesis between the results in the 'international' and 'national' studies and the 'local' results to help determine if pre-university qualifications are an indicator of how a student will perform at university. Each of these three studies places a slightly different emphasis on what is being measured. However, the commonality between them has led to some similarity of findings. The 'local' study has used a controlled data set and, as such, has been able to replicate investigations over three cohorts of students, therefore, making the statistical results more compelling. The 'local' results have confirmed the corresponding findings from the 'international' and 'national' studies, thereby adding to the research area.

Implications for university admission, learning and teaching, and the A-level curriculum have been discussed. Further discussion of these implications is in the following Conclusions chapter (Chapter 7).

Chapter 7. Conclusions

This thesis has demonstrated the potential impact on the way Computer Science is taught in the early stages of a degree programme. Recommendations include the implementation of strategies, based on student study background, which lead to more-personalised learning-approaches being adopted. This recommendation can be considered through Learning and Teaching committees.

As A-levels will remain the primary recruiting qualification in the UK for the foreseeable future, this research has the opportunity not only to influence Admissions Tutors into broadening the range of A-level subjects considered for admission but also to raise awareness among academic staff of the syllabuses in subjects such as Computing and Maths. This awareness is necessary for academic staff to be able to have a proper understanding of the subject-specific knowledge and skills which students 'should' have acquired and to be able to tailor their teaching to these.

The results presented are primarily from each cohort's quantitative performance over their first two years of study. A number of investigations have been run to determine if, and where, any significant differences lie, firstly, between individual cohorts and, secondly, within the combined-cohorts with regard to specific A-level subjects.

All evidence suggests that the A-level subjects investigated in this thesis are valid entry subjects and none of these subjects is to the detriment of the Durham student performance. It has become clear from many of the results that some of the differences revealed serve to support and enhance the findings of similar studies which have adopted different approaches and to some extent, different measurements. The contribution of this thesis is to add further to the body of knowledge surrounding A-level subjects as predictors of success at university.

Statistical analysis in this and other studies has shown that having specific subjects such as A-level Computing or A-level Maths makes no significant difference to the final outcome – whether that be at the end of year one or year two or even the final degree classification. Aside from the statistics, it would be difficult to argue that having subject-specific knowledge, such as that acquired in A-level Maths or Computing is irrelevant. Having this prior knowledge affords a student some advantage, however small.

Maths and programming, which between them make up three quarters of Durham's core modules in year one, are important for progression in a Computer Science programme. There is no doubt that some aspects of Computer Science require mathematical competence, which is invariably provided through A-level Maths. Results presented in this thesis confirm that these students achieve a higher year one mean than their counterparts (sections 5.5.1.2.1. and 5.7.1). Prior programming experience has also been shown to provide students with some advantage but this advantage would appear to be not only from school experience but also from personal interest and having the intrinsic motivation to want to be able to program well.

It is recognised within this thesis that other factors can contribute to a student's level of success and many of these are subjective and, therefore, difficult to measure. Motivation is one of these factors and is crucial for a student to reach their academic potential. Often this motivation comes as a natural characteristic of the student but also by engagement with the subject and this can often be driven by the approach to learning which a student adopts and the learning environment provided for the student.

Many students can be 'coached' by their school teachers to get the grades they need for university admission; they may have been taught to be surface learners (section 2.3.4) and often in their experience it works. In contrast to this instructivist or behavioural approach (section 2.2.1), learning in Computer Science takes, in many instances, a constructivist approach in that problem-based learning (section 2.4.1) and peer learning, e.g. group projects and pair programming, feature highly in some of the Durham curriculum. Increasing student motivation and their performance requires them to be able to integrate and organise new knowledge with prior knowledge, thereby taking more responsibility for their learning and being less dependent on the teacher.

The data sample used in this thesis is representative of Durham Computer Science students and this has lent itself, in the main, to quantitative data analysis. The data has provided a reasonably wide and inclusive coverage of students so that results are likely to be representative of a wider section of the A-level student population and, from this, more generalised conclusions can be drawn. For each of the Durham cohorts investigated, a similar student profile has been shown, with the majority of students having at least one of the A-level subjects: Computing, Maths, Physics or ICT, with a good percentage of students having at least two of these.

Student perception, and in particular the perception of those who studied A-level Computing, was that it provided reasonable preparation for the year one Computer Systems and the year one Programming and Data Structures modules. For some students, having confidence in believing that they have a good grasp of subject-specific knowledge, whether this is justified or not, can in itself be a motivating factor

Having A-level ICT has resulted in no significant impact on attainment other than ICT students having a lower (but not statistically significant) mean for programming, compared with those that have not. This result may well be anomalous as there are no obvious justifications for this. For instance, there is virtually no programming within A-level ICT and so these students are at no particular disadvantage compared to other novice programmers.

An important outcome of this research identified that students should not be discouraged from studying A-level Computing and should not then be discouraged from doing Computer Science at university simply because they do not have A-level Maths (although departmental policy sometimes does not allow the latter). Recommendations made in section 6.3.1 highlight how subject-specific prior knowledge should be welcomed but mechanisms put in place, first, to determine a student's level of ability and, secondly, to keep students engaged in the learning process and especially in those areas where there is an overlap between this prior knowledge and the year one syllabus. Providing a range of learning experiences for students, those with as well as those without prior learning in a subject, can be done by supporting or 'scaffolding' (section 2.5) with e-learning systems which allow for the personalisation of learning, with students engaging in controlling their own learning.

7.1. Criteria for success

A number of research questions were posed in Chapter 1 (section 1.3); this research can be judged in terms of success in providing answers. The answer to each of these research questions is summarised below.

Research questions:

- **Analysis of cohort:** *Does student performance in exams and coursework and their end-of-year performance differ between cohorts?*

There are differences detected between the cohorts but these differences are the consequence of the year one exam mean which shows a significant decline from a particularly high mean in 2004 (section 5.4 Table 5.11). Possible reasons for this have been discussed in section 5.11. In contrast, for year two exams the opposite effect is shown with an improvement of means from 2004 onwards (section 5.4 Table 5.11). The result of this is that, overall, the means are balanced out and so, combining the two years, all cohort module means are not statistically different (section 5.4.7).

- **Subject Analysis:** *Does having a particular A-level (e.g. Computing) result in an improved performance in year one and year two compared with not having that A-level?*

Having Maths A-level resulted in a better performance at the end of year one for those students, compared with non-Maths students (section 5.5.1.2). However, this difference is statistically small and there were no statistical differences between these two groups at the end of year two. There were no statistically significant differences for other subjects.

a. **Subject Analysis (subject-combinations):** *Does having a combination of A-level subjects, e.g. Maths and Computing, result in an improved performance compared with having neither subject?*

Maths and Physics is the most common combination of A-level subjects which students arrive with (section 5.1 Table 5.7). There is a significant difference in means between cohorts for those students with this combination of subjects (section 5.6.1). Cohort 2004 has the highest mean for year one and 2006 the lowest mean. Cohort 2006 had the largest number of students with this combination, compared with the other cohorts, and yet had the lowest mean at the end of year one.

For combined-cohort means, this particular subject combination results in higher year one means compared with the means for those students who have neither subject (section 5.6.1 Table 5.39).

- **Coursework Categories:** *Does having a particular A-level subject result in a better performance for certain types of coursework compared with not having that A-level?*

Specific A-level subjects did result in differences in student performance for coursework

i Having Maths A-level resulted in a medium positive effect in theoretical coursework for year one and year two and, similarly, a small positive effect for year two programming, compared with not having Maths (section 5.7.1).

ii Having Computing A-level resulted in a small positive effect in year one for programming and also for programming and reporting coursework, compared with not having Computing (section 5.7.2)

iii Having Physics A-level resulted in a small positive effect in programming, compared with not having Physics (section 5.7.4).

- **Computing and ICT A-level syllabuses:** *Having identified an overlap of topics between the year one Computer Science syllabus and A-level Computing and ICT, do those students with these subjects have an advantage over non-Computing and non-ICT students in specific module assignments because of specific prior knowledge?*

Syllabus overlap was identified in two Computer Science modules, year one Programming and Data Structures and year one Computer Systems. The Programming and Data Structures (double module worth 40 credits) module structure is two thirds programming and one third data structures. A-level Computing students performed better, compared with non-Computing students, in only one of four assignments in this module (section 5.8). However, this was not in the Java programming component but an assignment on data structures.

The year one Computer Systems module (single module worth 20 credits) comprises equally weighted sub-modules: Operating Systems, Networks, Machine Architecture and Databases, each of which is present in the A-level Computing syllabus and, therefore, relevant to this module. It was, however, in only one of four assignments (Machine Architecture) where Computing students performed better than non-Computing students (section 5.8).

a. **Computing and ICT A-level syllabuses:** *Does it make any difference for students which ICT or Computing A-level exam board they used? Does one particular exam board better prepare these students than another exam board?*

From the results presented it would appear that the choice of A-level exam board for Computing and ICT does not have any impact on performance in year one (section 5.9.1 and 5.9.2). No one particular exam board provided students with better preparation for year one than another.

- **Students' perception of Computing and ICT A-level:** *Research question 4 identifies if there are significant differences in coursework assignments between students who do and do not have Computing/ICT A-level. Do these findings bear any relation to the ICT and Computing students' perception about their year one studies in respect of what these A-level subjects provided them with?*

Analysis of the student questionnaire revealed that ICT and Computing A-level students felt that they were already familiar with many of the topics and terminology presented to them in the Computer Systems module. Similarly Computing students also perceived that they had an advantage, albeit small, in programming. However, analysis shows that, other than for results reported in research question 4 and 4a, the advantage the Computing and ICT students perceived themselves to have, did not materialise in higher means for specific areas compared to those students who did not study these A-levels.

7.2. Limitations of study

In a study which involves individual students, there is a multitude of factors which can affect their learning. Therefore, it can be expected that any study will have limitations. Factors associated with each student can include the learning styles which students have adopted, A-level subjects and grades, type of school attended, facilities provided and teacher experience in subjects being taught, student motivation, gender, ethnicity and many others. Many of these factors are outside the scope of this thesis but are candidates for further work.

The statistical analysis in this thesis has not looked for correlation between the A-level subjects investigated and final degree classification. Measuring student performance in year three is problematic, because of the variability in student choice of modules which contrasts with the fixed nature of the first two years. In addition to this, students entering year three of study have matured as people and often their attitudes and priorities have changed. These factors, coupled with recognition of the need for a 'final push' towards the end of the degree programme, can result in a change in their approach and commitment to work.

Four A-level subjects have been investigated in this thesis. These subjects are the most common to Durham Computer Science students and the only ones that were sufficient in number for statistical analysis. This thesis has concluded that A-level entry subjects do not impact significantly on overall student performance. It is, however, recognised that it is possible that there could be other subjects, or combinations of subject, which could contribute to academic success in Computer Science.

The syllabus comparisons undertaken have been between the Computing and ICT A-levels and the Durham Computer Science degree programme. Similarly, only comparisons between the Computing and the ICT exam boards have been made. There is, however, a variety of Maths A-levels offered to students by the exam boards (section 4.3.1.a) and no distinction has been made between these Maths syllabuses in this thesis.
In fully determining a student's academic performance, it would also be necessary to take into account factors such as those described earlier, including learning styles and the quality of the teaching experienced in school. This is particularly important in view of the change in learning methods where, for many students, the subject content and method of presentation at university will differ considerably from their A-level experience. There are difficulties in determining and measuring these factors and they are, therefore, outside the scope of this thesis.

7.3. Future work

Section 7.2 highlighted that there is considerable scope for further work in this research area. For instance, of particular relevance to the outcomes described in this thesis are factors that include a student's motivation and their school experience, in terms of the prior knowledge they bring with them to university, and their engagement with the curriculum (Reason, Terenzini et al. 2006). Many of these factors can be affected by other inter-related factors. These factors include:

- the method(s) of delivery of teaching,
- student's teacher's experience (for instance, relatively few school teachers are specifically qualified to teach in the Computing discipline),
- whether students were educated in a state (either selective or comprehensive) or independent school.

All of these factors contribute to student achievement at A-level and are opportunities for further research to identify the impact they may have on a student's ability to study in higher education are discussed in the following sections.

7.3.1. Student motivation

Motivation is an important characteristic of a student as it is related to their level to engagement in the learning process. Motivation is a factor that has been mentioned many times within this thesis and numerous research projects identify how its importance is paramount (Reeve 2004). Some research has already been carried out in this area although this has mainly been in the context of problem-based learning (Feassler, Hinterberger et al. 2006; Sungur and Tekkaya 2006). Measuring a student's motivation alongside the measures used in this thesis, could lead to increased understanding of the impact of prior qualification compared to an individual's motivation in a specific discipline such as Computing.

7.3.2. Programming and the artists

This research has focused on science-based entry qualifications, however some Admission Tutors believe that art-based subjects could also teach students critical skills useful within the Computing discipline. Further analysis using other A-level subjects such as Art or Music could therefore reveal interesting results. For example, programming is thought to be a creative activity and so it would be interesting to determine if there is some connection between the more artistic students and their performance in programming. This analysis would, however, require a larger data set to make results viable and therefore is likely to need a National study.

7.3.3. A-level Maths syllabus

Whilst this research and other studies have shown that A-level Maths provides only a small advantage for student in year one, Maths is seen as a defining characteristic of Computer Science education. It would, therefore, be of interest to look closely at the syllabuses for each of the Maths A-levels offered to students by the exam boards (section 4.3.1.a). This would determine which, if any, of these 'flavours' of Maths is more appropriate for year one in a Computer Science degree. Such knowledge could allow the Computer Science community to offer a more specialised programme based on the Maths topics that were not within particular syllabi and would provide a solid foundation for year one of a Computer Science degree programme.

7.3.4. Teacher perception of Computing and ICT A-levels

The number of students taking A-level Maths is increasing in the UK but this and other 'traditional' subjects are still disproportionately an independent school domain. Independent schools traditionally enter higher numbers of students for subjects thought of as the most academic (Smith 2009). These schools accounted for just over 20% of the A-level Maths students in 2009, even though the sector only educates about 8% of the population (Shepherd 2009). However, the number of students wanting to take A-level Computing was down 7% in 2009 compared to 2008, with the curriculum being blamed for "putting students off" (Thomson 2009). Many of the issues surrounding the decline in interest would certainly relate to the regard in which Computing is held in school. Is it, for instance, a 'traditional' subject or just a 'nerdy hobby'?

It would therefore be very useful to study school teachers' perceptions of subjects like A-level ICT or Computing and also to investigate if certain beliefs are characteristic of certain types of schools. It would be interesting to investigate how teachers regard the ICT and Computing disciplines and to assess the impact of their perceptions on their students. The relative lack of 'specialist' teachers in Computing may well send out a clear message to students about the subject's importance.

This thesis has put into question the strategies adopted by many higher education Computing Admission Tutors. For instance, this work has shown that students who have not studied Maths at A-level perform, on average in year two, at an equivalent standard to peers who have studied A-level Maths. The sections above point to areas where much further work can, and should, be conducted to fully understand the impact of prior study on progression within the Computing discipline at degree level. However, this work already points to an important lesson. In a discipline that is struggling to get the students necessary to maintain the needs of its industry, policies such as those that are related to student admission, must be defined by research outcomes rather than by unjustified perception.

7.4. Contributions of this thesis

The contributions of this thesis to the wider community are that:

- It adds to the current body of knowledge surrounding A-levels as predictors of success at university level
- Durham data has provided the opportunity for a longitudinal study with three consecutive cohorts each exposed to a fixed core syllabus in year one and two. This has removed a considerable amount of variability of the data compared to other studies and therefore results are more compelling
- Investigations have taken the data outcomes down to the level of categories of coursework and specific assignments within these categories

- An analysis framework has been developed which can be used by other institutions and disciplines other than Computing. This framework has been developed using robust and well used statistical procedures.

APPENDIX A

Online Surveys
Develop, launch and analyse Web-based surveys

Main Survey Page

The purpose of this short survey is to determine what impact your previous studies at school/college in computing related subjects have helped (or not)in your studies here in the Department.

Your information

1. Please enter your name

2. Please enter your ITS user id

3. Please indicate which subject you studied at school/college at A2 or equivalent level (if your subject title is slightly different to the two below please tick the most appropriate).
(select all that apply)

ICT Computing

4. Please indicate your current year of study at University.

Level 1 Level 2 Level 3 Level 4

Relevance to Computer Science Degree

5. Did you find that the content of the subject (ICT or Computing) you studied at school/college useful for any of your modules? For modules you have not done please choose N/A.

	Select one answer for each module						**It would be extremely useful if you could provide examples for each module of how your prior knowledge gained at school/college, was useful (or not).**
	Not at all	**A little**	**Some**	**A lot**	**Quite a lot**	**N/A**	
a. Programming and data structure (PDS)							
b. Introduction to Programming (only for Nat Sci's) (IP)							

c. Foundations of Computer Science (FCS)							
d. Computer Systems (BCS)							
e. Formal Aspects of Computer Science (FA)							
f. Programming and Reasoning (PR)							
g. Software Engineering (SE)							
h. Software Applications (SA)							
i. Theory of Computing (TC)							
j. Computer Systems II (CSYSII)							
k. Other							

6. Had you any experience in ICT or Computing (other than from school or college) before coming to University?
If applicable please indicate what this was.

a. How well do you think this experience helped in your studies?

Not at all A little Some Quite a lot N/A

b. Please indicate why you chose the answer to (a) above. *(Optional)*

and Finally ....

7. Overall do you feel that your prior studies in ICT/Computing (or equivalent) at school have helped in your studies here in the Department?

Yes No

Please comment on why it has helped (or not). *(Optional)*

Continue > | Check Answers & Continue >

Top | Log out

Copyright | Contact Us

References

Adey, P., R. Fairbrother, D. Wiliam, B. Johnson and C. Jones (1999). Learning Styles & Strategies A Review of research, Kings College London School of Education.

Albanese, M. (2000). "Problem-based learning: why curricula are likely to show little effect on knowledge and clinical skills." Medical Education **34**(9): 729-738.

Alexander, S., C. Martyn, L. Ken, A. June, D. Mats, B. Roger, L. Cary and S.-K. Dermot (2003). "Case Studies in Admissions to and Early Performance in Computer Science Degrees." SIGCSE Bull. **35**(4): 137-147.

Allinson, C. and J. Hayes (1996). "The Cognitive Style Index: A Measure of Intuition-Analysis For Organizational Research." Journal of Management Studies **33**(1): 119-135.

Atherton, J. S. (2005). "Behaviourism." Learning and Teaching: Behaviourism, from http://www.learningandteaching.info/learning/behaviour.htm Accessed: 20 May 2008.

Ausubel, D., J. Novak and H. Hanesian (1968). Educational Psychology: a cognitive view. New York, Holt, Rinehard & Winston.

Barrett, T., I. M. Labhrainn and H. Fallon, Eds. (2005). Handbook of Enquiry and Problem-based Learning. Galway, AISHE and CETL, Nui Galway.

Barrows, H. S. and R. M. Tamblyn (1980). Problem based Learning, An Approach to Medical Education New York, Springer.

BCS. (2006). "IT Professionals in Education: increasing the supply." BCS Education and Training Forum, British Computer Society: www.bcs.org/server.php?show=ConWebDoc.3356 (accessed 10th Sept 2009).

Bekhradnia, B. and J. Thompson (2002). Who does best at University? Higher Education Funding Council, http://www.hefce.ac.uk/Learning/whodoes/ (accessed 12th July 2009)

Ben-Ari, M. (1998). Constructivism in Computer Science Education. Twenty-Ninth SIGCE Atlanta GA, USA, ACM.

Biggs, J. B. (1987). Learning Process Questionnaire Manual. Student Approaches to Learning and Studying, Australian Council for Educational Research, Hawthorn.

Bloomfield, S. L. (1967). NEA: Trojan Horse in American Education, The Paradigm Company.

Boustedt, J., A. Eckerdal, R. McCartney, J. E. Mostrom, M. Ratcliffe, K. Saunders and C. Zander (2007). Threshold Concepts in Computer Science: Do they exist and are they useful? SIGCE, Kentucky, USA, ACM.

Boyle, R., J. Carter and C. Martyn (2002). "What Makes them Succeed? Entry, progression and graduation in Computer Science." Journal of Further and Higher Education **26**(1): 3 - 18.

Boyle, R. and M. Clark (2002). A-Level Computing: its content and value. Research Report: 2005.15 School of Computing, University of Leeds, UK.

Brennan, J. F. (2003). The IBM Watson Laboratory at Columbia University A History. F. Cruz.

Bristol-Online-Surveys. (2008). "http://www.survey.bris.ac.uk/."

Brown, A. L. and J. C. Campione (1994). Guided Discovery in a community of learners. Classroom lessons: Integrating cognitive theory and classroom practice. K. McGilly. Cambridge, MA, MIT Press/Bradform Books.

Brush, T. A. and J. W. Saye (2002). "A Summary of Research Exploring Hard and Soft Scaffolding for Teachers and Students Using a Multimedia Supported Learning Environment." The Journal of Interactive Online Learning **1**(2): 1-12.

Burd, E. L., S. Drummond and B. M. Hodgson (2003). Using Peer and Self Assessment for Group Work. 4th Annual LTSN-ICS, Galway, Eire.

Butler, R., D. Inman and D. Lobb (2005). "Problem-based learning and the medical school: another case of the emperors' new clothes?" Adv Physiol Educ **29**: 194-196.

Campbell, P. F. and P. M. George (1984). "Predicting the success of freshmen in a computer science major." Commun. ACM **27**(11): 1108-1113.

Cassidy, S. (2004). "Learning Styles: An overview of theories, models and measures." Educational Psychology **24**(4): 419-444.

CEEBL. (2005). "Centre for Excellence in Enquiry-based Learning." Retrieved 27th Sept 08, from http://www.campus.manchester.ac.uk/ceebl/.

Clark, M. and R. Boyle (2006). Computer Science in English High Schools: We Lost the S, Now the C Is Going. Informatics Education – The Bridge between Using and Understanding Computers**:** 83-93.

Coffield, F., D. Moseley, E. Hall and K. Ecclestone (2004). Learning styles and pedagogy in post-16 learning - A systematic and critical review. L. S. R. Centre.

Coffield, F., D. Moseley, E. Hall and K. Ecclestone (2004a). Should we be using learning styles? What research has to say to practice., Learning and Skills Research Centre.

Collins, J. P., G. R. White and J. A. Kennedy (1995). "Entry to medical school: an audit of traditional selection requirements." Medical Education **29**(1): 22-28.

Davies, P. (2003). Threshold Concepts: how can we recognise them? European Association for Research on Learning and Instruction (EARLI), Padova, Italy.

Davies, P. and J. Mangan (2006). Embedding Threshold Concepts: from theory to pedagogial principles to learning activities. Threshold Concepts within the Disciplines Symposium, Glasgow.

DEMOS (2005). About learning: Report of the Learning Working Group.

Denning, P. J. (2005). "Is computer science science?" Commun. ACM **48**(4): 27-31.

Distlehorst, L. H. and R. S. Robbs (1998). "A comparison of problem-based learning and standard curriculum students: Three years of retrospective data." Teaching And Learning In Medicine **10**(3): 131-137
.

Downing, K., R. Ho, K. Shin, L. Vrijmoed and E. Wong (2007). "Metacognitive development and moving away." Educational Studies **33**(1): 1-13.

Drinan, J. (1997). The Limits of Problem-Based Learning. The Challenge of Problem-Based Learning. D. Boud and G. Feletti. London, Kogan Page: 333-338.

Drummond, S. and S. Jamieson (2005). The Threshold Concept: Helping Students Towards Mastery 6th Annual Conference of the Subject Centres for Information and Computer Sciences, York.

Drummond, S. A. and M. Devlin (2006). Software Engineering Students' Cross-site Collaboration: An Experience Report. Proceedings of The 7th Annual Conference of the ICS HE Academy Conference, Trinity College Dublin.

Dunn, R. (1984). "Learning Style: State of the Science." Theory into Practice **23**(1): 10-19.

Dunn, R., K. Dunn and G. Price (1975). The Learning Style Inventory. PriceSystems. Lawrence, KS.

Eckerdal, A., R. McCartney, J. E. Mostrom, M. Ratcliffe, K. Sanders and C. Zander (2006). Putting Threshold Concepts into Context in Computer Science Education. ITiCSE, Bologna, Italy, ACM.

Entwistle, N. (2003). Concepts and Conceptual Frameworks Underpinning the ETL Project (Occasional Report 3). Enhancing Teaching-Learning Environments in Undergraduate Courses.

Feassler, L., H. Hinterberger, M. Dahinden and M. Wyss (2006). Evaluating Student Motivation in constructivistic, problem-based introductory computer science courses. World Conference on E-Learn, Honolulu.

Felder, R. M. and R. Brent (2005). "Understanding Student Differences." Journal of Engineering Education (January): 57-72.

Field, A. (2005). Discovering Statistics using SPSS. London, Sage.

Fosnot, C. T., Ed. (2005). Constructivism: theory, perspectives, and practice, Teachers College Press New York.

Friedman, P. and R. Alley (1984). "Learning/Teaching Styles: Applying the Principles." Theory into Practice **23**(1): 77-81.

Greening, T. (2000). Emerging Constructivist Forces in Computer Science Education: Shaping a New Future? Computer Science Education in the 21st Century. T. Greening. New York, Sprinter-Verlag**:** 47-81.

Greening, T., J. Kay, J. Kingston and K. Crawford (1997). Results of PBL Trial in First-Year Computer Science. ACSE, Melbourne, Australia, ACM.

Hagan, D. and S. Markham (2000). Does it help to have some programming experience before beginning a computing degree program? Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education. Helsinki, Finland, ACM.

Hannafin, M., S. Land and K. Oliver (1999). Open learning environments: Foundations, methods, and models. Instructional design theories and models Volume II. C. Reigeluth, Mahway, NJ: Erlbaum. **2:** 115-140.

Hmelo-Silver, C. E., R. G. Duncan and C. A. Chinn (2007). "Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006)." Educational Psychologist **42**(2): 99-107.

Honey, P. and A. Mumford (1992). The Manual of Learning Styles, Peter Honey.

Jarvela, S. (2006). Personalised Learning? New Insights into Fostering Learning Capacity. Personalised Education. E. R. Yahill, J. Cannon, D. Grandriieux and D. Instance, OECD Publishing**:** Chpt 2.

JISC. (2006). "e-Learning Capital Programme." Retrieved 4th August 2008, from http://www.jisc.ac.uk/whatwedo/programmes/programme_elearning_capital.aspx.

Kahn, P. and K. O'Rourke (2005). Understanding Enquiry-Based Learning. Handbook of Enquiry & Problem Based Learning. T. Barrett, I. Labhrainn and H. Fallons. Galway, Eire, Creative Commons.

Kenny, R. F. (2005). On-line Problem-based Learning: Panacea or Problematic? Distance Education Technology Symposium - DETS. Edmonton, Alberta.

Kirschner, P. A., J. Sweller and R. E. Clark (2006). "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching." Educational Psychologist **41**(2): 75-86.

Klahr, D. and M. Nigam (2004). "The Equivalence of Learning Paths in Early Science Instruction." Psychological Science **15**(10): 661-667.

Kolb, A. Y. and D. A. Kolb (2005). The Kolb Learning Style Inventory - Version 3.1 2005 Technical Specification. Experience Based Learning Systems, Hayes Group.

Kolb, D., R. Boyatziz and C. Mainemelis (2000). Experiential Learning Theory: Previous Research and New Directions. Perspectives on cognitive, learning, and thinking styles R. Sternberg and L. Zhang, Lawrence Erlbaum.

Kratzig, G. and K. Arbuthnott (2006). "Perceptual Learning Style and Leanring Proficiency: A Test of the Hypothesis." Journal of Educational Psychology **98**(1): 238-246.

Lajoie, S. P. (2005). "Extending the Scaffolding Metaphor." Instructional Science **33**(5-6): 541-557.

Land, R., G. Cousins and J. Meyer (2005). Threshold Concepts and Troublesome Knowledge (3): Implications for Course Design and Evaluation. Improving Student Learning 12 - Diversity and Inclusivity. C. Rust. Oxford, Oxford Brookes University**:** 53-64.

Lashley, C. and P. Barron (2006). "The learning style preferences of hospitality and tourism students: Observations from an international and cross-cultural study." International Journal of Hospitality Management **25**(4): 552-569.

Leadbetter, C. (2006). The Future of Public Services: Personalised Learning. Personalised Education. E. R. Yahill, J. Cannon, D. Grandriieux and D. Instance, OECD Publishing**:** Chpt 7.

Manouselis, N. and D. Sampson (2002). Dynamic Knowledge Route Selection for Personalised Learning Environments Using Multiple Criteria. Intelligence and Technology in Educational Applications Workshop (ITEA), Innsbruck, Austria.

Martin, S. (1999). "Behaviourism: the rise and fall of a discipline." American Psychological Association Monitor (online) **30**(11).

Marton, F. and R. Saljo (1976). "Outcome as a function of the learner's conception of the task." British Journal of Educational Psychology **46**: 115-127.

Mayer, R. E. (2004). "Should There Be a Three-Strike Rule Against Pure Discovery Learning." American Psychologist **59**(1): 14-19.

McManus, I. C., D. A. Powis, R. Wakeford, E. Ferguson, D. James and P. Richards (2005). "Intellectual aptitude tests and A levels for selecting UK school leaver entrants for medical school." BMJ **331**(7516): 555-559.

McNeill, K. L., D. J. Lizotte, J. Krajcik and R. W. Marx (2006). "Supporting Students' Construction of Scientific Explanations by Fading Scaffolds in Instructional Material." The Journal of the Learning Sciences **15**(2): 153-191.

Meyer, J. and M. Shanahan (2001). "A Triangulated Approach to the Modelling of Learning Outcomes in First Year Economics." Higher Education Research & Development **20**(2).

Meyer, J. H. F. and R. Land (2003). Threshold Concepts and Troublesome Knowledge (1): linkages to ways of thinking and practising within the disciplines. Improving Student Learning - Ten Years On. C. Rust. Oxford, OCSLD.

Meyer, J. H. F. and R. Land, Eds. (2006). Overcoming barriers to student understanding : threshold concepts and troublesome knowledge London ; New York :, Routledge.

Miliband, D. (2006). Choice and Voice in Personalised Learning. Personalised Education. E. R. Yahill, J. Cannon, D. Grandriieux and D. Instance, OECD Publishing**:** Chp 1.

Moström, J. E., J.Boustedt, A. Eckerdal, R. McCartney, K. Sanders, L. Thomas and C. Zander (2009). Computer Science Student Transformations: Changes and Causes. ITICSE 2009: 14th Annual Conference on Innovation and Technology in Computer Science Education Paris, http://db.grinnell.edu/sigcse/iticse2009/Program/viewAcceptedProposal.asp?sessionType=paper&sessionNumber=49 (assessed 10th Sept 1009).

Norman, G. R. and H. G. Schmidt (1991). "The psychological basis of problem based learning: A review of the evidence." Academic Medicine **67**: 557-565.

Norman, G. R. and H. G. Schmidt (2000). "Effectiveness of problem-based learning curricula: theory, practice and paper darts." Medical Education **34**: 721-728.

Ozga, J. and L. Sukhnandan (1998). "Undergraduate Non-Completion: Developing an Explanatory Model." Higher Education Quarterly **52**(3): 316-333.

Paludan, J. P. (2006). Personalised Learning 2025. Personalised Education. E. R. Yahill, J. Cannon, D. Grandriieux and D. Instance, OECD Publishing**:** 83-126.

Parnas, D. (1998). "Software Engineering programmes are not computer science programmes." Annals of Software Engineering **6**(1-4): 19-37.

Pavlov, J. P. (1941). Scientific Study of the So-Called Psychical Process in the Higher Animal. Lectures on Conditioned Reflexes - Twenty Five Years of Objective Study of the Higher Nervous Activity (Behaviour) of Animals W. H. Gantt. London, Lawrence and Whishart**:** 91-96.

Pea, R. D. (2004). "The Social and Technological Dimensions of Scaffolding and Related Theoretical Concepts for Learning, Education, and Human Activity." The Journal of the Learning Sciences **13**(3): 423-451.

Perkins, D. N. (1999). "The Many Faces of Constructivism." Educational Leadership **57**(3): 6-11.

Piaget, J. (2000). Piaget's Theory of Cognitive Development. Childhood Cognitive Development: The Essential Readings. K. Lee, Blackwell Publishing**:** 33-47.

Prince, K., H. van Mameren, N. Hylkema, J. Drukker, A. Scherpbier and C. v. d. Vleuten (2003). "Does problem-based learning lead to deficiencies in basic science knowledge? An empirical case on anatomy." Med Educ **37**(1): 15-21
.

Puntambeka, S. and R. Hubscher (2005). "Tools for Scaffolding Students in a Complex Learning Environment: What Have We Gained and What Have We Missed?" Educational Psychologist **40**(1): 1-12.

QAA. (2000). "Computing Subject Benchmarks."   Retrieved 15th January 2010, from http://www.qaa.ac.uk/academicinfrastructure/benchmark/honours/computing.pdf.

Quintana, C., B. J. Reiser, E. A. Davis, J. Krajcik, E. Fretz, R. G. Duncan, E. Kyza, D. Edelson and E. Soloway (2004). "A Scaffolding Design Framework for Software to Support Science Inquiry." The Journal of the Learning Sciences **13**(3): 337-386.

Rahm, E. and H. H. Do (2000). Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin. wwwiti.cs.uni-magdeburg.de
.

Rayner, S. (2007). "A teaching elixir, learning chimera or just fool's gold? Do learning styles matter?" Support for Learning **22**(1): 24-30.

Reason, R. D., P. T. Terenzini and R. J. Domingo (2006). "FIRST THINGS FIRST: Developing Academic Competence in the First Year of College." Research in Higher Education **47**(2): 149-175 (pp 156).

Reeve, J. (2004). Understanding Motivation and Emotion Wiley & Sons.

Reiser, B. J. (2004). "Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work." The Journal of the Learning Sciences **13**(3): 273-304.

Revell, P. (2005). Each to their own. Education Guardian.

Riding, R. and S. Rayner (1998). Cognitive Styles and Learning Strategies, Fulton.

Rosenshine, B. and C. Meister (1992). "The use of scaffolds for teaching higher-level cognitive strategies." Educational Leadership **49**(7): 26-33.

Rountree, N., T. Vilner, B. C. Wilson and R. Boyle (2004). "Predictors For success in studying CS." SIGCSE Bull. **36**(1): 145-146.

Ryan, G. (1997). Ensuring that students develop an adequate, and well-structured, knowledge base. The Challenge of Problem-Based Learning. D. Boud and G. Feletti. London, Kogan Page: 125-136.

Sadler-Smith, E. (1997). "Learning Style: frameworks and instruments." Educational Psychology **17**(1): 51-63.

Sampson, D., C. Karagiannidis and Kinshuk (2002). "Personalised Learning: Educational, Technological and Standardisation Perspective." Interactive Educational Multimedia **4**: 24-39.

Savery, J. R. (2006). "Overview of problem-based learning: Definitions and distinctions." Interdisciplinary Journal of Problem-based Learning **1**(1): 9-20.

Savery, J. R. and T. M. Duffy (1995). Problem Based Learning: An instructional model and its constructivist framework. Constructivist Learning Environments: Case Studies in Instructional Design. B. Wilson, Educational Technology.

Savin-Baden, M. (2000). Problem-based Leaning in Higher Education: Untold Stories, SRHE and Open University Press.

Schunk, D. H. (2008). Learning Theories An Educational Perspective, Pearson Education.

Sear, K. (1983). "The Correlation between A Level Grades and Degree Results in England and Wales." Higher Education **12**(5): 609-619.

Shapiro, A. M. (2004). "How including Prior Knowledge as a Subject Variable May Change Outcomes of Learning Research." American Educational Research Journal **41**(1): 159-189.

Shaw, M. (2000). Software engineering education: a roadmap. International Conference on Software Engineering, Limerick, Ireland, ACM Press.

Shepherd, J. (2009). A-level choices: Pupils pick subjects to impress. Guardian.co.uk, 20th August 2009. www.guardian.co.uk/education/2009/aug/20/a-level-students-pick-maths (accessed 19th Sept 2009).

Sheppard, D. (2007). "Threshold Concepts: Perspectives from Computer Science: Y and Recursion: A Threshold Concept in Computer Science." Retrieved 4th Sept 2008, from http://www.caret.cam.ac.uk/tel/outcomes.html.

Shneiderman, B. (1998). "Relate-Create-Donate: a teaching/learning philosophy for the cyber-generation." Computers & Education **31**(1): 25-39.

Simons, K. D. and J. D. Klein (2007). "The Impact of Scaffolding and Student Achievement Levels in a Problem-based Learning Environment " Instructional Science **35**: 41-72.

Skinner, B. F. (1953). Behaviourism. Science and Human Behaviour. New York, The Free Press.

Skinner, B. F. (1964). "New methods and new aims in teaching." New Scientist **122**(May).

Smith, A. (2009). Achievements mask school divide BBC News education: 21st August. http://news.bbc.co.uk/1/hi/education/8211955.stm (accessed 23rd Sept 2009).

StatSoft, I. (2007). Electronic Statistics Textbook, OK:StatSoft, http://www.statsoft.com/textbook/stathome.html.(accessed 2nd May 2009).

Strobel, J. and A. van-Barneveld (2008). "When is PBL More Effective? A Meta-synthesis of Meta-analyses Comparing PBL to Conventional Classrooms." Interdisciplinary Journal of Problem-based Learning **3**(1): Article 4: http://docs.lib.purdue.edu/ijplb/vol3/iss1/4 (accessed 17th Sept 2009).

Sungur, S. and C. Tekkaya (2006). "Effects of Problem-Based Learning and Traditional Instruction on Self-Regulated Learning." The Journal of Educational Research **99**(5): 307-317.

Tait, H., N. J. Entwistle and V. McCune (1998). ASSIST: a reconceptualisation of the Approaches to Studying Inventory. Improving Students as Learners. C. Rust, Oxford: The Oxford Brookes University, The Oxford Centre for Staff Development and Learning.

TeacherNet. (2009). "Types of Qualifications in England, Wales and Northern Ireland http://www.teachernet.gov.uk/educationoverview/uksystem/examinationsandqualifications/ (accessed 29th June 2009)."

The Joint Task Force (1991) "Computing Curricula 1991: A summary of the ACM/IEE-CS Joint Curriculum Task Force Report."

The Joint Task Force. (2001). "Computing Curricula 2001: Computer Science." from http://acm.org/education/curric_vols/cc2001.pdf.

The Joint Task Force. (2001). "Computing Curricula 2001: Computer Science pp11." from http://acm.org/education/curric_vols/cc2001.pdf.

The Joint Task Force. (2004). "Software Engineering 2004: Computing Curriculum Series." from http://sites.computer.org/ccse/SE2004Volume.pdf.

The Review Taskforce (2008) "Interim Review of the CC2001 Computing Curricula, Computer Science Volume." ACM/IEEE

Thomson, R. (2009). Technology A-level student numbers fall again. ComputerWeekly.com: 20th August 2009 http://www.computerweekly.com/Articles/2009/ (accessed 17th Sept 2009).

Tomayko, J. E. (1998). "Forging a discipline: An outline history of software engineering education." Annals of Software Engineering **6**: 3 - 18.

Vygotsky, L. S. (1978). Interaction between Learning and Development. Mind in Society The Development of Higher Psychological Processes

M. Cole, V. John-Steiner, S. Scribner and E. Souberman, Harvard University Press: pp86.

Williams, R. (2008). Statistics for Psychology, Durham University.

Wood, D., J. S. Bruner and G. Ross (1976). "The Role of Tutoring in Problem Solving." Journal of Child Psychol, Psychiat., **17**: 89-100.

Yin, R. K. (2003). Case Study Research: Design and Methods. London, Sage.