Report on the Provision of Courses in Computer Science in Upper Second Level Education Internationally

Report on the Provision of Courses in Computer Science in Upper Second Level Education Internationally

Authored by *Neil Keane and Clare McInerney, the Irish Software Research Centre*

Expert Group: *Prof. Kevin Ryan, Prof. Tiziana Margaria, Prof. Rory O'Connor, Dr. Chris Exton, Dr. Oliver McGarr, Prof. Sibel Erduran and Mr. Ted Parslow*

Lero THE IRISH SOFTWARE RESEARCH CENTRE

NCCA An Chomhairle Náisiúnta Curaclaim agus Measúnachta National Council for Curriculum and Assessment

EPI·STEM

Table of Contents

Table of Figures

List of Tables

Executive Summary

This report details the findings of research conducted into the international provision of Computer Science courses at upper second level across a number of selected jurisdictions. The research was commissioned by the National Council for Curriculum and Assessment (NCCA), and conducted by a consortium of researchers from Lero (The Irish Software Research Centre), the National Centre for STEM at the School of Education in the University of Limerick and the Third Level Computing Forum.

The Computer Science curricula of the five selected jurisdictions (England, Scotland, New Zealand, Ontario and Israel) were analysed to answer research questions provided by the NCCA, resulting in the following key research findings:

- **Rationale & Motivations.** Adding a new subject to a school curriculum needs a robust rationale for its inclusion, as each subject competes for teaching resources and time in an often-crowded curriculum. Many important goals and objectives have been articulated internationally as motivations for the establishment of Computer Science on a curriculum. These include educational aspects such as Computational Thinking, problem solving, social and cultural objectives in relation to inclusion, equality and empowerment, and economic motivations such as innovation, employment and economic development.

- **Clarity of Terminology.** It is important to distinguish between Computer Science and Information Communications Technology, when establishing a Computer Science programme to purposely cultivate a better understanding about the nature of

Computer Science amongst the many stakeholders, students, parents, teachers, management etc.

- **Computer Science.** As each of the selected jurisdictions (re)established their Computer Science programmes, the theory and application of the fundamental principles and concepts of Computer Science was central to each curriculum containing substantive course content. This centrality of the theory of Computer Science is further evidenced in the renaming of the revised programmes from titles of an ICT designation to Computer Science/Computing Science.

- **Programming.** Programming is a central component of every Computer Science curriculum reviewed in this report, with course designers interweaving Computer Science and programming components to provide both conceptual and experiential learning outcomes. Programming learning outcomes involve problem solving and Computational Thinking rather than just coding as an activity.

- **Programming Languages and Tools** Across each jurisdiction, the choice of programming language for upper second level was decided locally. In principle, this allowed teachers to choose languages that fitted their own context. However, in practice the choice of languages was very often determined by the availability of supporting materials and course resources for a language, a teacher’s own knowledge and experience of a language, and/or the languages supported by assessment authorities.

- **Information Technology & Digital Literacy.** Each of these terms are prominent in the senior cycle curricula of the researched jurisdictions as implicit and global educational goals rather than as discrete subjects. Instruction in Information Technology and Digital Literacy are increasingly delivered from primary schooling onwards on a cross disciplinary basis, rather than viewed as the sole realm of computing subjects.

- **Participation.** Student participation rates in Computer Science in the jurisdictions that have recently revised their programmes (England, Scotland & New Zealand) are low, but have grown steadily in many instances since their introduction. As these jurisdictions are still in the process of transitioning from legacy ICT programmes and are still within the subject introductory phase, it is too early to draw any firm conclusions from the statistics.

- **Participation by Gender.** Participation by girls in upper second level Computer Science is very low across the reports jurisdictions (with the notable exception of Israel which has a long-term female participation rate of over 40%).This finding reflects the low levels of female participation in technology and computing related fields across industry, academia and society in general. Statistics from the jurisdictions indicate that when girls participate in Computer Science courses, they tend on average to achieve better grades than their male counterparts.

- **Pre-requisites.** Apart from Scotland, none of the report jurisdictions have strict requirements for pre-requisite studies prior to a student undertaking an upper second level Computer Science programme. In England, the examination boards make

recommendations to the extent of a student's prior studies, the remaining jurisdictions have no expectations of prior learning.

- **Assessment.** The assessment strategies detailed within our report jurisdictions combine the provision of traditional written examinations and course work such as project work that students complete independently during their school year. While the weighting assigned to each of these two components varied by jurisdiction, a key finding was the universal recognition of the importance of providing assessment that enabled each student to exhibit the practical application of skills learnt, alongside their understanding of the subject content.

- **Teacher Professional Development.** Each of the report jurisdictions recognised the central role of teachers in the adoption, implementation and sustainability of a Computer Science curriculum. As most of the jurisdictions had a large cohort of ICT teachers from legacy programmes an emphasis was on the professional development of these existing teachers in a manner that was cost and time effective. The resulting strategy of fostering communities of practice, integrated with multiple professional development approaches (such as training, mentoring, research, accreditation and peer to peer knowledge sharing) was found to be both a welcome and effective approach to these challenges.

The report concludes by drawing together its key findings to provide the following recommendations to assist the NCCA's work in developing advice for the introduction of a Computer Science course for Leaving Certificate:

- **Content.** An Irish curriculum should reflect the extensive and largely agreed body of content required to teach a Computer Science programme which is available to those charged with developing the new leaving certificate Computer Science subject.

- **Assessment.** An Irish curriculum should reflect the importance of providing assessment that enables each student to exhibit the practical application of skills learnt, alongside their understanding of the subject content.

- **Pedagogy.** An Irish curriculum should employ computer science pedagogical techniques and practices that harness new tools and technologies for teaching computer science and enable a seamless transition for learning between the school and home environments.

- **Supporting Student Learning.** An Irish curriculum should provide students with appropriate learning opportunities in Computer Science earlier in their schooling. The current pilot of the short course in coding in the junior cycle provides one such example of an earlier support.

- **Computational Thinking.** An Irish curriculum should consider the extent to which Computational Thinking and other aspects of Computer Science may influence and inform other complementary subjects. The new course in computational mathematics

at Leaving Certificate which focusses on modelling rather than application exemplifies this concept.

- **Subject Sustainability.** An Irish curriculum should acknowledge that participation rates in Computer Science programmes have been found to be initially low internationally. As other jurisdictions face the challenge of sustainability, an opportunity exists to learn what works well and not so well in early adopters.

- **Teacher Professional Development.** An Irish curriculum should ensure that teachers are supported in their professional development to acquire the subject, curricular and pedagogical knowledge and skills needed to promote student engagement and enjoyment of the subject, whilst maintaining its rigour. The more recent Computer Science programme implementations in England, Scotland and New Zealand offer many valuable lessons in the provision of pre-and in-service training.

The introduction of Computer Science into the upper second level in Ireland will meet a need that currently exists in the Irish education system. Particularly in the context of a national and international debate about reinforcing Computational Thinking and problem-solving skills through education. Computer Science students learn problem solving, Computational Thinking, innovation and creativity. From a societal and cultural standing, it helps students become active creators and producers rather than passive users of technology, while offering potential opportunities for enhanced inclusion and equity. Economically, students with an understanding of Computer Science are required across all industries to support innovation, development and employment.

While implementation challenges exist, Ireland is well positioned to learn from the international experience detailed in this report, avoiding the pitfalls while being able to benefit from valuable lessons learnt. The rapid growth of the importance of Computer Science educational programmes internationally, strongly suggests that now is the opportune time to introduce Computer Science into the Irish upper second level curriculum.

Introduction

This report details the findings of research conducted into the international provision of Computer Science courses at upper second level across a number of jurisdictions. The research was commissioned by the National Council for Curriculum and Assessment (NCCA), and conducted by a consortium of researchers from Lero (The Irish Software Research Centre), the National Centre for STEM at the School of Education in the University of Limerick and the Third Level Computing Forum.

The commissioning of this report reflects the recognised importance "*of supporting the development of new opportunities for learners to undertake in-depth study of ICT in the senior cycle*" detailed in the Department of Education and Skills Digital Strategy for Schools(2015 – 2020) (2015 ,P.23). This report continues an established research tradition of the NCCA into the provision of Computer Science courses in the Irish education system, which have been examined at various stages since computers first became available to schools (National Council for Curriculum and Assessment (NCCA), 2004a, 2004b, 2005; O'Doherty, Gleeson, Johnston, McGarr, & Moody, 2004).

This report details the curricula from each of the jurisdictions selected for analysis (Table 1), with each examined under four distinct themes: current challenges, course content and learning outcomes, nature of the subject, and the provision of teacher professional development. The report concludes by collating the important messages arising from the research findings and discusses the implications for the NCCA's work in developing advice for the introduction of a Computer Science course for the upper second level of Irish secondary schools.

The jurisdictions selected for inclusion in the research all have active Computer Science programmes in the upper second level of their post-primary schools, some of which are long established with others beginning as recently as 2014. The analysis conducted for the report endeavours to highlight the many successes and challenges experienced in meeting their programme aims and objectives.

Implemented	Jurisdiction
2014	***England***
2012	***Scotland***
2012	***New Zealand***
2009	***Ontario***
1998	***Israel***

Table 1: Jurisdictions Selected for Research Report

The research group recognise the limitations of this desktop research. Difficulties encountered include:

- making general comparisons across jurisdictions due to differing naming conventions and levels of abstraction used in each.
- the evolving and emergent nature of the programmes implemented since 2010.

- the restricted access to detailed resources in some jurisdictions.
- and a reliance on secondary data and case studies interpreted by third parties.

Despite these limitations, the research group remain satisfied that its analysis of the prescribed curriculum, academic papers and digital media, provides a balanced and comprehensive account of the current provision of Computer Science programmes internationally.

Section One

Challenges in the Field

What’s in a name

It is vital when establishing a Computer Science (CS) programme in schools to purposely cultivate a better understanding about the nature of CS amongst the many stakeholders, students, parents, teachers, management etc. Success in other jurisdictions has only followed after the distinction between CS (involving programming and rigorous analytical thinking, with a consistent set of core principles) and Information Communications Technology (typically involving the use of computer software) has been made clear (Brown et al., 2013). The reasons why this clarity is vital are threefold:

- **Confusion.** The ‘use’ of technology (computing as a tool), cross-discipline e-learning, and instruction on Digital Literacy are not the same as teaching CS as a core academic discipline (Wilson et al., 2010; Jones, S., 2010). An example of this within the Irish context, is the successful ECDL courses which is frequently undertaken in transition year and often referred to as ‘ICT’ or even computer science.

- **Illusion.** The interchangeable use of the terms Computer Science, Information Technology, ICT, and Computing can create the illusion that CS is already being taught in a school system. A suitable analogy is that CS is no more about the use of computers than astronomy is about the use of telescopes, biology about the use of microscopes or chemistry is about the use of beakers and test tubes (Fellows & Parberry, 1993).

- **Lack of Identity.** CS as a relatively young discipline has suffered from loss of identity owing to various incomplete or inadequate perceptions of its nature (Denning, 2009).

> Some of the sampled jurisdictions that revised their curricula to shift from ICT to a more rigorous CS approach (Hubwieser, Giannakos, et al., 2015), have also revised the subject name to better communicate the nature of the subject (i.e. the English curriculum no longer uses its previous term ICT in any revised documentation, instead using only CS).

The outcome of this confusion, illusion and lack of identity has caused a number of difficulties including but not limited to:

- A misperception among increasingly technological savvy students that CS is boring, involving demotivating and routine ICT activities (Mitchell, 2004).
- A lost opportunity for students to experience the rich discipline of CS, with knock-on effects on future educational and career choices and on gender equality.
- A lack of identify and recognition leads to CS teachers being undervalued, making the task of attracting new teachers into CS more difficult. This can lead to CS teachers being under qualified and a continuation of the cycle.

As a response to these problems and considering the importance of clearly articulating the nature of the subject, Simon Furber in his seminal report to the Royal Academy suggested that the term ICT be reviewed and disaggregated into a framework of Computer Science, Information Technology and Digital Literacy. Each component is acknowledged as important in a modern education. The purpose of disaggregating them is simply to make visible the discipline of CS which provides enduring principles and methods. In this report, for conceptual clarity Furber’s definitions of these terms are utilised throughout.

Glossary of Terms

- **Computer Science** *"referring to the scientific discipline of Computer Science, covering principles such as algorithms, data structures, programming, systems architecture, design, problem solving etc"(Furber, 2012, P.17).*

- **Information Technology** *"should be understood to mean the assembly, deployment, and configuration of digital systems to meet user needs for particular purposes"(Furber, 2012, P.17).*

- **Digital Literacy** *"should be understood to mean the basic skill or ability to use a computer confidently, safely and effectively, including: the ability to use office software such as word processors, email and presentation software, the ability to create and edit images, audio and video, and the ability to use a web browser and internet search engines."(Furber, 2012, P.17).*

Goals & Objectives of Computer Science Programmes Internationally

The introduction of a new subject onto a school curriculum is not without its challenges, as each subject competes for teaching resources and time in an often-crowded curriculum. Each new subject needs a robust rationale for its admission, one that will resonate strongly with a range of stakeholders (Furber, 2012).

The (re)establishment of Computer Science (CS) on the curricula of the jurisdictions sampled, reflected a universal recognition of the burgeoning importance of information and digital technologies in modern society. This importance is reflected in the many rationale and motivations articulated for the introduction of CS programmes in the curricula and in the related documentation of our selected jurisdictions. These motivations are categorised under three headings: educational, social and cultural, and economic.

Educational Motivations

- **Problem Solving:** Per Gal-Ezer and Stephenson, "*Problem solving lies at the very heart of CS. Learning and doing computer science requires students to state problems clearly and unambiguously, write an algorithmic solution for the problem that takes into account all boundary conditions (robustness), determine that the algorithm produces the right answer (correctness), and test that the solution is efficient (complexity considerations). At each step in this process, students are learning core skills*"(2014 ,P.2).

- **Computational Thinking:** (Wing, 2006) is a transferable skill that comprises of a set of problem-solving techniques (abstraction, modelling, algorithmic thinking etc.) and a set of intellectual practices (dealing with complexity, persistence, tolerance for ambiguity etc.), that have value across disciplines and relevance in a modern education (Brown et al., 2013). "*Science is a way of thinking much more than it is a body of knowledge*" (Sagan, 1997).

- **STEM:** "*Computing is a fascinating blend of engineering, mathematics, science, and technology….In short, computing is the quintessential STEM subject*" (Jones, 2010 ,P.5). CS makes an intellectual and educational contribution as a scientific discipline emphasizing creativity, constructiveness and precision.

Social & Cultural Motivations

- **Inclusion:** "*Computing is embedded in the world around us and plays an important part in many aspects of our lives at home, work and at leisure*"(Scottish Qualifications Authority, 2016a ,P5). As the use of computing user interfaces become ubiquitous across society, it is of critical importance that individuals do not become dis-enfranchised through an inability to use computers (i.e. E-voting, Government E-services, E-billing, Online Banking etc.) and instead possess the important digital competencies to fully participate as citizens.

  As a young and emergent profession, Computer Science offers individuals career opportunities that are unencumbered from their previous economic, educational, and social & cultural circumstances.

- **Equality**: The statistics for female participation in technical careers are stark across Europe with just 6 to 7% of these careers being filled by women (Accenture, 2014). In contrast, females made up 38% of the ICT workforce in Israel in 2013 (Israel Central Bureau of Statistics, 2013) . The Israeli high school CS programme, considered by many to be the most progressive in the world, has had an average female participation rate of over 40% between 1995 – 2011 (Armoni & Gal-Ezer, 2014).

  The Israeli experience supports the view that "*if learners are never introduced to Computer Science as a disciplinary area and to the knowledge base and approaches that computing academics and professionals use, then they are unlikely to be able to determine whether this is for them, and this is widely regarded as a key factor in the lack of gender diversity in the computing industry*" (Webb et al., 2016 ,P13)

- **Empowerment:** Having a diverse range of active creators and producers rather than just passive consumers of technology is of considerable value to both individuals and the society they live in. The "*Cultural rationale rests on enabling people to be drivers of cultural change rather than having change imposed by technological developments.*"(Webb et al., 2016 ,P.2).

- **Reflecting Society's Interests.** Across each of the jurisdictions researched, grassroots movements in the form of coding clubs and events have become commonplace. These clubs and events reflect the natural appetite and interest of children, their parents, and of the many volunteers from both education and industry who run them. 'CoderDojo' started in Cork in 2011, is now a worldwide movement with more than 1,100 verified Dojos in 63 countries (CoderDojo, 2016).

Economic Motivations

- **Economic Development**. CS is a cross-disciplinary field which increasingly facilitates new breakthroughs in arts, commerce, engineering, medicine, and the natural sciences. Enabling students to take CS in upper second level has been shown to lead to more students taking the discipline in higher education (Armoni & Gal-Ezer, 2014; Tim Bell, Newton, Duncan, & Jarman, 2014). This flow-on effect helps to develop and sustain a nation's economic competitive advantage in a world driven by technology and will also ensure that many graduates of other disciplines will have the knowledge to use computing in their own fields.

- **Innovation and Entrepreneurship.** At the heart of the CS discipline are problem-solving, intellectual and creativity skills such as abstraction, algorithmic thinking, and dealing with complexity, tolerance for ambiguity, persistence and design. These transferable skills all help support innovation and entrepreneurship in the economy.

- **Employment**. Ireland is emerging as a global technology hub, with 105,000 people directly employed in both indigenous and multinational technology firms. Many of these roles are well paid and require high level ICT skills (ICT Ireland, 2016) . The potential employment prospects in the sector are substantial. The "ICT Skills Action Plan" published by the Department for Education and Skills, and the Department for Jobs, Enterprise and Innovation, indicates a continuing strong demand for high-level ICT skills with 44,500 job openings forecast to arise over the period to 2018 from both expansion and replacement demand (2014) .

Discussions & Debates

Are there any current discussions or debates about the future of the subject?

Introduction. As computers and their use reaches near ubiquity across modern societies, the importance of the provision of instruction in 'computing' is accepted as an essential component of a contemporary education. The debates that follow this acknowledgement are numerous, varied and complex, with answers most often dependent on the educational, social, cultural and economic context of the jurisdiction in question.

This report touches briefly on three important and interwoven debates that have emerged from the experience of our jurisdictions in implementing a Computer Science(CS) curriculum in their upper second level, namely:

- At what age is it best to introduce Computing into schools?
- Is Computer Science for everyone?
- How can teachers be best supported to meet the challenge of curriculum change?

When to introduce Computing subjects onto the curriculum?

Introducing school children as young as six to aspects of CS and computational thinking is regarded as beneficial to their longer term learning in the subject(Duncan, Bell, & Tanimoto, 2014). Many facets of CS require gradual acquisition and development, with evidence mounting that "*exposing students to programming and computer science concepts before the*

age of 12 seems likely to be more effective in building their confidence and reducing the gender gap in the area"(Tim Bell et al., 2014, P.5).

The emergence worldwide of 'coding clubs' such as Apps for Good[1], Code Club[2] ,CodeClub Aotearoa [3], and CoderDojo[4], points to the appetite of younger children and their parents for learning in this area. Where there are concerns on the teaching of young children 'skills for employment', these are countered by a focus on problem solving through Computational Thinking often without computers, and in the cases computers are used to 'program', cut down systems that match the cognitive level of the students- such as "ScratchJnr", "Beebots", and "Scratch".

Several of our jurisdictions have made recent changes (or have proposed changes) to their curricula to introduce younger students to Computing and Computational Thinking. In 2012, Israel introduced a new six-year program covering grades seven through twelve introducing a new CS curriculum for middle schools (ages 12 – 15) (Zur Bargury, 2012; Zur Bargury et al., 2012).

The New Zealand Education Ministry announced in July 2016, that it was "*Formally integrating digital technology into the curriculum is intended to support young people to develop skills, confidence and interest in digital technologies and lead them to opportunities across the IT sector*"(Ministry of Education, New Zealand, 2016a). This announcement of formal integration, reflected the fact that the NZ curriculum already allowed for Computing subjects in its lower school levels, but it had been left to " *the enthusiasm and awareness of a particular teacher or their school leadership to make that happen*" (Tim Bell, 2014, P.29). These changes to be implemented across the whole learner pathway by 2018, is the

[1] https://www.appsforgood.org/
[2] https://www.codeclub.org.uk/
[3] http://www.codeclub.nz/
[4] https://coderdojo.com/

continuation of a 2014 initiative called “A nation of curious minds” which along with other aspects was charged with “*considering which level topics around computational thinking should be introduced, including the possibility of introduction in the first year of compulsory schooling”*(Webb et al., 2016, P.8).

In England, changes made by the Department for Education in 2014, introduced a national curriculum for Computing that makes teaching of the subject compulsory from the age of 5 to 16 years (Brown, Sentance, Crick, & Humphreys, 2014). In Scotland, the introduction of the 2011 Curriculum for Excellence, placed Computing firmly in the curriculum, with the requirement that all students must at the very least be exposed by the age of 14 to a level of Computing to meet set learning outcomes and experiences(Jones, 2011).

Is Computer Science for everyone?

This is a multi-faceted question for which there is no singular answer. The experience in each of the research jurisdiction reflects the nature of their society, culture, economy, and existing and historic school experiences. The debate is further complicated by how the various parties may define CS and related subjects, very often reflecting different levels of granularity.

Using Furber’s disaggregation of the term ICT into Digital Literacy, Information Technology and Computer Science (2012) (Figure 1), the following can be advanced from the analysis of our jurisdictions. Digital Literacy is advocated as an essential skill in a modern education and as an entitlement for every student in their years of compulsory schooling. Information Technology being increasingly delivered on a cross disciplinary basis from primary school onwards. CS is considered as a discrete scientific subject leading to academic qualifications in middle and upper second level schooling.

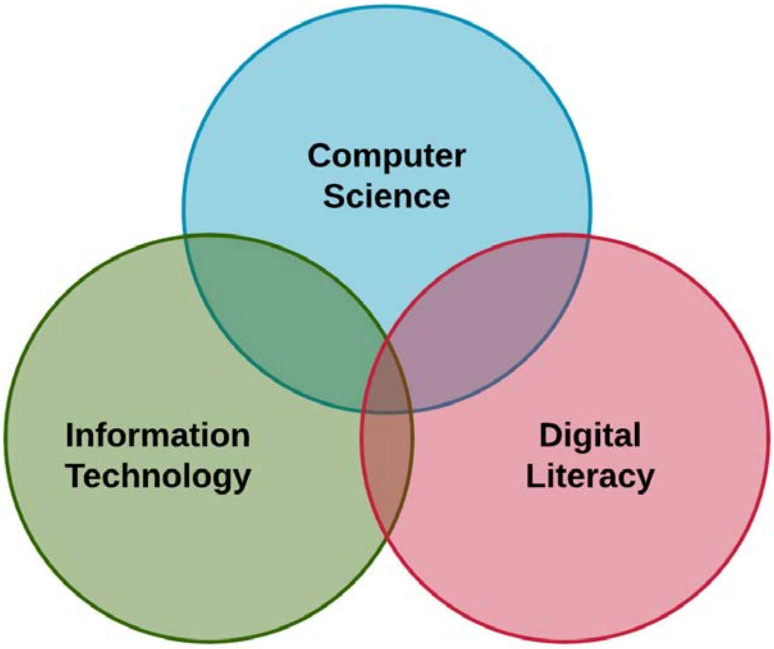

Figure 1: Furber's Disaggregation of ICT

The exposure of students to each aspect is considered important within our jurisdictions, with each striving to offer its students curriculum that provide learnings across all three appropriate to their age and cognitive abilities.

How can teachers be best supported to meet the challenge of curriculum change?

How best to support teachers in meeting the 'grand' challenge of curriculum change is the subject of extensive debate across all our jurisdictions. This challenge is being addressed in England, Scotland and New Zealand as the CS curricula has changed, where each is facing the additional complication of needing to support the professional development of large numbers of existing teachers of related studies (ICT) in making the change over.

This challenge is regarded as the greatest risk and constraint to the success of a new CS curriculum. How each jurisdiction has and is addressing these challenges is discussed in section four of this report.

Section Two

Course Content & Learning Outcomes

Introduction

This section of the report provides details on course content and learning outcomes from the various computer science programmes offered across our sampled jurisdictions. It focusses on the prescribed curriculum as detailed by each authorised body, noting that the actual experience of curricula may often differ across classes, schools and school districts within a jurisdiction, and across the jurisdictions themselves. The authors recognise the importance of observing a curriculum in its structural and sociocultural context, however detailed consideration of these aspects lies beyond the scope of this report. Table 2, provides descriptive information on the upper second level programmes within each jurisdiction.

Last Major Revision	Jurisdiction	Program Name in Upper Second Level	Age: Upper Second Level	Terminology	Qualifications 3rd Level Entry
2014	**England**	Computer Science	15 - 17	Year 11-13	A Level's
2012	**Scotland**	Computing Science	15 - 17	S5-6	Highers & Advanced Highers
2012	**New Zealand**	Digital Technologies	15 - 17	Grade 10-12	NCEA
2009	**Ontario**	Computer Studies	15 - 17	Grade 10-12	Provincial Report Card
1998	**Israel**	Computer Science	16 - 18	Grade 10-12	Bagrut

Table 2: Structural Detail of Upper Second Level CS Courses by Jurisdiction

The published curricula provided a detailed and thorough research resource (detailed in the bibliography). To categorise the specified course content and learning outcomes the guiding research questions were used as sorting criteria, in turn relating these to Furber's framework of Computer Science, Information Technology and Digital Literacy.

While conducting this research, it emerged that a large proportion of the course content and outcomes specified in the curricula straddled all five research questions provided, making attempts at parsing and direct comparison a difficult and sometimes inexact endeavour.

Computer Science

To what extent do the courses include:
Understanding and applying the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

Introduction. In answering this research question, it was very notable that Computer Science (CS) and Programming content were most often interwoven into course designs throughout our jurisdictions' curricula, providing courses that offered both conceptual and experimental learning outcomes. This made it difficult to parse content between CS and Programming, leading to duplication and overlapping classifications on occasion.

England. The English Department of Education stipulate that the AS and A-Level courses in CS "*must encourage students to develop: an understanding of, and the ability to apply, the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation*"(2014, P.1). These aims and objectives are interwoven into the new subject curriculum specifications provided by the three academic qualification providers[5] for the English CS AS and A-Levels. These specifications which replaced previous AS and A-Levels in Computing and ICT, were released for teaching in September 2015, and examined in June 2016 for AS and June 2017 for A-Levels for the first time.

[5] Assessment and Qualifications Alliance (AQA), Oxford Cambridge and RSA (OCR) & Eduqas

Scotland: Among the stated aims of the Scottish curriculum for Computing Science, launched in 2012, is to enable learners to "*extend and apply knowledge and understanding of advanced concepts and processes in computing science*" and "*communicate advanced computing concepts clearly and concisely, using appropriate terminology*"(Scottish Qualifications Authority, 2016a ,P.5).

These aims are manifest in course specifications for the Highers and Advanced Highers qualifications in year's 12 & 13. The curriculum consists of two courses, "Software Design and Development" and "Information System Design and Development", the focus of which is on the practical implementation of computing skills ('Curriculum for Excellence' ethos) underpinned by the theoretical and conceptual foundations of the CS discipline.

The "Software Design and Development" course offers specific modules in CS topics such as 'data types and structures', 'algorithm specification' and 'programming constructs' (Scottish Qualifications Authority, 2016a).

New Zealand. In 2011 the NZ Ministry for Education introduced a new curriculum for Digital Technology which introduced CS as a mainstream subject for upper second level students. Prior to this introduction, the NZ curriculum focused on teaching students to 'use' computers rather than focus on CS as a discipline in its own right (T. Bell, Andreae, & Robins, 2014).

The new Digital Technologies curriculum over the three senior high school years consists of thirty-five achievement standards (modules) across five specialist areas: Programming and Computer Science, Digital Information, Digital Media, Digital Infrastructure and Electronics. Each standard is worth between three and six credits (1 credit = ten hours), with schools combining standards of between eighteen and twenty-four credits to make a course each year.

As three of the thirty-five standards focus on CS and five on Programming over the three senior years, teachers combine these modules with those from the other digital technologies areas (such as Information Systems, Web Design, Operating Systems, Embedded Systems), to develop courses that suit the local context and the needs of their students (T. Bell et al., 2014).

Year	Reference	Standard	Credits
10	1.44	Demonstrate understanding of basic concepts from computer science	3
11	2.44	Demonstrate understanding of advanced concepts from computer science	4
12	3.44	Demonstrate understanding of areas of computer science	4

Table 3: NZ Computer Science Standards (Modules)

The three CS standards (modules) focus on the understanding and application of core concepts such as algorithms, data representation and programming constructs (New Zealand Qualifications Authority, 2011, 2014a, 2015a) (Table 3)

Ontario. In Ontario "*the term computer studies refers to the study of computer science, meaning computer and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society*"(Ministry for Education, 2008). Algorithms and data structures are denoted as one of four critical areas of learning which are taught throughout the revised curriculum. The three other critical learning areas are

software development, program correctness and efficiency, and professional and ethical responsibility.

The Ontario Computer Studies curriculum provides students with an opportunity for greater specialisation in their second year of upper second level (grade 11), providing separate tracks in CS and Computer Programming(CP) respectively. The CS track is designed as the university preparation course, with students getting to explore a broad spectrum of CS concepts and applications in the 'Introduction to Computer Science' course in Grade 11, and 'Computer Science' in grade 12. The CP track covers many of the topics in the CS track, but with a greater focus on practical programming skills leading to opportunities in further education in community college programmes and industry careers.

Israel. The Israeli CS curriculum, in situ since 1998, is considered by some to be the "*the most rigorous Computer Science high school program in the world*" (Jones, 2011 , p.6) . The curriculum has undergone a number of changes in emphasis in course delivery over time to reflect technological and pedagogical advances, however the underlying principles remain unchanged (Hazzan, Gal-Ezer, & Blum, 2008). Three of these principles relate directly to the research question:

- *Computer Science is a full-fledged scientific subject.*
- *The program should concentrate on the key concepts and foundations of the field.*
- *Conceptual and experimental issues should be interwoven throughout the program.*

These principles are embodied in each of the CS courses offered in Israeli high schools. The "zipper principle" is utilised, where conceptual classroom based learning is intermixed with experimental computer lab based learning. In this manner the lasting theory and concepts of the CS discipline underpin the practical side where the implementation technologies are

prone to change (Gal-Ezer & Harel, 1999). Programming is viewed as a method of teaching key concepts and foundations of the field, and not purely as an end in itself(Gal-Ezer & Stephenson, 2014).

Summary of findings. Substantive course content that comprises of the understanding and application of the fundamental principles and concepts of CS are very evident in the curricula of each of the reviewed jurisdictions.

From the review of the documentation it is evident that as each jurisdiction's curriculum has undergone a substantial revision, a strengthened emphasis on the fundamental principles and concepts of CS has been adopted. This bolstering of the conceptual elements of CS in the revised curricula, is further evidenced in the renaming of the programmes from titles of an ICT designation to Computer Science/Computing Science.

These changes can be broadly interpreted as resulting from stakeholder consensus in each of the jurisdictions on the need to refocus on CS as a discipline and away from computing 'use as a tool' (Carrell, Gough-Jones, & Fahy, 2008; Department of Education, 2014; Furber, 2012; Gal-Ezer et al., 1995; Ministry of Education, Ontario, 2008; Scottish Qualifications Authority, 2016a).

Table 4 displays exemplar courses from each of the jurisdictions researched, which have a focus on understanding and applying the fundamental principles and concepts of CS, including abstraction, logic, algorithms and data representation.

	Exemplars of Related Course Provision Across Jurisdictions
Israel	• Fundamentals of CS 1 • Fundamentals of CS 2 • Theory of CS
Ontario	• Introduction to CS • Programming Concepts & Skills • Topics in CS
New Zealand	• Basic concepts from CS • Advanced concepts from CS • Demonstrate understanding of areas of computer science
Scotland	• Standard algorithms • Data representation • Algorithm design and implementation
England	• Theory of computation • Fundamentals of algorithms • Data types, data structures and algorithms

Table 4: Exemplars of Related Course Provision Across Jurisdictions

Programming

To what extent do the courses include:

Analysing problems in computational terms, with repeated practical experience of writing computer programs to solve such problems. Programming is defined in the broader sense to include the design of the algorithms underlying the programmes and, to some extent, considerations of correctness and efficiency.

Introduction. Programming enables students to learn experientially about the underlying principles of algorithms, data structures, and Computational Thinking; bringing the fundamentals of 'computing' from the conceptual to a tangible, visible and palpable existence. In modern Computer Science(CS) curricula, computer programming is offered as a problem-solving process rather than just a coding activity.

Central to writing computer programs to solve a problem are the underlying Computational Thinking skills such as abstraction, evaluation and decomposition (Wing, 2006). And whereas in principal writing a computer program is but one approach to solving such problems, in practice "representing a solution to a problem as a program provides a perfect way to evaluate the solution, as the computer will execute the instructions to the letter, forcing the student to refine their solution so that it is very precise."(Webb et al., 2016 , P.5). In this manner students acquire Computational Thinking skills by learning to design and implement software programs.

England. The revised English National Curriculum specifies subject content for CS with aims and objectives, reflecting key aspects of programming, that all examination board's specifications must be built upon. These specifications must encourage students to develop (Department of Education, 2014 ,P.1):

- *an understanding of, and the ability to apply, the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation*
- *the ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so*
- *the capacity for thinking creatively, innovatively, analytically, logically and critically*

With specifications requiring students to develop knowledge, skills and understanding of the fundamentals of CS and Programming to include (2014 ,P.2):

- *fundamentals of programming*
- *the importance of the efficiency of an algorithm; that this can be measured in terms of execution time and space requirements, and that the efficiency of algorithms that perform the same task can be compared*
- *design, write and test programs to either a specification or to solve a problem*
- *to appropriately structure programs into modular parts with clear, well-documented interfaces*

Table 6 provides an illustration course content covered by the English examination boards for A-Levels. These specifications from the 'Assessment and Qualifications Alliance'(AQA) feature programming both in discrete courses and/or as central in other course's content *(2016)*.

AQA Course	Course Content
Fundamentals of programming	**Programming:** Data Types, Programming concepts, Arithmetic operations, Relational operations, Boolean operations, Constants and variables, Random number generation, String-handling operations, Exception handling, Subroutines, Parameters of subroutines, Returning a value/values, Local variables in subroutines, Global variables, Role of stack frames, Recursive techniques. **Programming Paradigms:** Procedural-orientated programming, Object-orientated programming.
Fundamentals of Data Structures	**Data Structures and Abstract Types:** Data structures, Single- and multi-dimensional arrays, Fields, records and files, Abstract data types/data structures. **Queues, Stacks, Graphs, Trees, Hash tables, Dictionaries, Vectors**
Fundamentals of algorithms	**Graph-traversal, Tree-traversal, Reverse Polish, Searching algorithms:** Linear search, Binary search, Binary tree search **Sorting algorithms:** Bubble sort, Merge sort. **Optimisation algorithms**
Theory of computation	**Abstraction and automation:** Problem-solving, Following and writing algorithms, Abstraction, Procedural abstraction, Functional abstraction, Data abstraction, Problem abstraction/reduction, Information hiding, Decomposition, Composition, Automation. **Regular languages:** Finite state machines (FSMs) with and without output, Maths for regular expressions, Regular expressions, Regular language. **Context-free languages:** Backus-Naur Form (BNF)/syntax diagrams **Classification of algorithms:** Comparing algorithms, Maths for understanding Big-0 notation, Order of complexity, Limits of computation, Classification of algorithmic problems, Computable and non-computable problems, Halting problem. **A model of computation:** Turing machine
Fundamentals of functional programming	**Functional programming paradigm:** Function type, First-class object, Function application, Partial function application, Composition of functions. **Writing functional programs, Lists in functional programming.**
Systematic approach to problem solving	**Aspects of software development:** Analysis, Design, Implementation, Testing, Evaluation
Computing Practical Project	**Purpose of the project:** The project allows students to develop their practical skills in the context of solving a realistic problem…during which they can extend their programming skills and deepen their understanding of computer science. The most important skill that should be assessed through the project is a student's ability to create a programmed solution to a problem or investigation.

Table 5: AQA ‘A-Level’ CS Specifications for Programming

Scotland. The purpose of the Computing Science (CS) curriculum “is to develop learners’ knowledge of the technological world and to develop their skills in developing computer-based solutions to problems”(Scottish Qualifications Authority, 2012 ,P.1).

Students over the course of their upper senior CS studies are expected to develop skills, knowledge and understanding of: (Scottish Qualifications Authority, 2015d ,P.7; 2016a ,P.8)

- *developing skills in computer programming and the ability to communicate how a program works by being able to read and interpret code*
- *analysing, designing, developing, implementing, testing and evaluating digital solutions (including computer programs) to complex problems across a range of contexts*
- *applying computational thinking to solve complex computing problems*
- *analysing complex problems within computing science across a range of contemporary contexts*

The Scottish CS curriculum consists of two programmes in both Highers and Advanced Highers, Software Design and Development, and Information System Design and Development. These programmes for both qualifications, specified by the Scottish Qualification Authority, have substantive programming learning outcomes as detailed in Table 7 (2015e, 2015f, 2015g, 2015h).

Qualification	Unit	Outcomes (*The learner will be able to)*
Highers	Software Design and Development	• Explain how programs work, drawing on an understanding of advanced concepts in software development and computer architecture. • Develop modular programs using one or more software development environments.
Highers	Information System Design and Development	• Develop information systems using appropriate development tools. • Consider the factors involved in the design and implementation of an information system.
Advanced Highers	Software Design and Development	• Explain how well-structured, complex modular programs work, drawing on understanding of programming constructs, algorithms and data integration. • Develop well-structured, complex modular programs. • Investigate and report on some contemporary programming paradigms.
Advanced Highers	Information System Design and Development	• Develop complex information systems using appropriate development tools. • Explain how information systems are developed and managed. • Investigate the implications of a contemporary development.

Table 6: SQA Computing Science Unit Programming Outcomes

New Zealand. The New Zealand Digital Technologies curriculum contains eight primary components, two of which are specific to our research question: (Ministry of Education, New Zealand, 2012)

- **Design a Software Program Structure**: *focuses on designing the structure of software programs. Initially students learn to specify variables and their data types, construct flexible and robust plans, and determine structures that combine well-chosen actions, conditions and control structures that provide well-structured logical solutions to tasks.*
- **Construct a Software Program**: *focuses on constructing a computer program for a specified task including testing and debugging the program to ensure the program works correctly.*

Five standards (modules) through years 11-13 provide the content to deliver the two components: (New Zealand Qualifications Authority, 2014b, 2015b, 2015c, 2015d, 2015e)

Year	Reference	Standard	Credits
10	**1.45**	Construct a plan for a basic computer program for a specified task. (Design)	3
10	**1.46**	Construct a basic computer program for a specified task. (Implement)	3
11	**2.45**	Construct a plan for an advanced computer program for a specified task. (Design)	3
11	**2.46**	Construct an advanced computer program for a specified task. (Implement)	3
12	**3.46**	Develop a complex computer program for a specified task. (Design & Implement)	6

Table 7: NZ Programming Standards (modules)

Ontario. The Ontario Ministry of Education curriculum states that the term Computer Studies "*is not about learning how to use the computer, and it is much more than computer programming…. It involves defining problems; analysing problems; designing solutions; and developing, testing, and maintaining programs*"(2008 , P.3). The program being *"relevant for all students because it incorporates a broad range of transferable problem-solving skills and techniques, including logical thinking, creative design, synthesis, and evaluation"*(2008 , P.3).

The importance and prevalence of programming in the Ontario upper second level Computer Studies is highlighted by the fact that three out of the four critical learning areas articulated in the curriculum are pertinent to it, namely : (Ministry of Education, Ontario, 2008)

- Software development
- Algorithms and data structures
- Program correctness and efficiency

These critical learning areas are comprehensively addressed in the Computing Studies courses specifications for Grades 10 to 12 (upper second level), detailed in table 9.

Grade	Course	Component
10	**Introduction to Computer Studies**	• Introduction to Programming Concepts, Writing & Code Maintenance
11	**Introduction to Computer Science**	• Programming Concepts and Skills Data types, Control Structures & Simple Algorithms, Subprograms & Maintenance • Software Development Problem Solving Strategies, Designing Software Solutions, Designing Algorithms, SDLC • Computer Environments and Systems
11	**Introduction to Computer Programming**	• Programming Concepts and Skills • Software Development • Computer Environments and Systems structure as above with differences in emphasis & constructs used
12	**Computer Science**	• Programming Concepts and Skills Data types, Modular Programming, Designing Algorithms • Software Development Project Management, Software Project • Designing Modular Programs Modular Design, Algorithm Analysis
12	**Computer Programming**	• Programming Concepts and Skills Data Structures, Using Standard Algorithms, Object-orientated Programming, Code Maintenance • Software Development Designing Standard Algorithms, Object-oriented Software solutions, GUI's, Project • Programming Environment Project Management Tools, Software Development Tools

Table 8: Programming in the Ontario Curriculum

Israel. Programming is an essential element in the Israeli upper second level CS curriculum, as it enables the application of the conceptual based class room learning in a practical manner. The ‘zipper’ principle is a term used to describe this interweaving of conceptual and experimental topics. Each module in the curriculum uses this approach, with it being most visible in the ‘Fundamentals of Computer Science 1 & 2’ and ‘Data Structures modules’.

A further principle of the Israeli programme is the inclusion of a module on other programming paradigms. This serves both to highlight alternative ways of algorithmic thinking in addressing a problem, and to reinforce a key learning principle that the underlying concepts of CS and Computational Thinking are more important than any specific implementing language. The “*emphasis is on the basics of algorithmics”* (Gal-Ezer & Harel, 1999 , P.1), providing students with “*insight, knowledge, and skills independent of specific computers and programming languages*”(Gal-Ezer et al, 1995, P.1). The second paradigm module includes several alternatives such as functional, logic and system-level programming (Gal-Ezer et al.,1995; Gal-Ezer & Harel,1999).

Summary of Findings. Programming is a fundamental component in the curricula of every jurisdiction detailed in this report, as it underpins the development of the knowledge and skills by students across all aspects of their CS courses.

CS and Programming content is interwoven in course designs throughout our selected jurisdictions’ curricula, providing both conceptual and experiential learning outcomes.

The concept of Computational Thinking is implicit in an underlying and prevailing manner throughout the CS curriculums of each jurisdiction, where it is most often used to express educational goals, aims or a pedagogical approach. The underlying skills it denotes are

universally present in programming courses, where Computational Thinking skills are acquired by students learning to design and implement software programs.

Programming Languages and Tools

If **Programming** is taught, what programming languages and tools are used and what, if any, is the first programming language and why is it been chosen.

Introduction. Programming is taught across all jurisdictions featured in this report.

In this dynamic field, there are a vast range of programming languages and tools available. These choices can be categorised as follows: (Hubwieser, Giannakos, et al., 2015 ,P.75)

- *Hardware based systems*, e.g. LEGO Mindstorms or Raspberry Pi,
- *Educational programming environments with own language* (Alice / Scratch)
- *Educational programming environments based on other languages* (BlueJ. Java's Cool)
- *Professional Used Programming languages* **(**C#, Java etc.)

In addressing this research question, each jurisdiction was analysed to identify (a) if the tools and teaching methods are prescribed centrally or decided locally, (b) what is the first programming language students are taught and why was it chosen. And (c) what programming language(s) is currently utilised in the schools for upper second level.

England. Only the required modality of a programming language to be used with the upper second level English Computer Science(CS) curriculum is specified, with an exact choice not mandated.

In Key Stage 3 (age 11 – 14), the national curriculum specifies the "*use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions*"(Department of Education, 2013). Given the popularity of the Scratch programming language, these department guidelines are "*broadly interpretable as 'Scratch plus a text-based programming language'; while the requirements before age 11 can be fully satisfied in Scratch, the 11–14 curriculum deliberately ensures that pupils move to full-text programming*"(Brown et al., 2014 ,P.9).

The English examination boards that provide assessment for the CS A-Level Qualification support a variety of general purpose programming languages, with Java (Greenfoot/BlueJ educational programming environment), Python and Visual Basic prominent across all three boards. As the new CS programme was rolled out in September 2015, it is too early to tell which languages will prove to be the most popular amongst the teachers delivering the content and the students being taught.

Scotland. In the upper second level of Scottish secondary schools**,** "*pupils learn to program in a "real" language of the teacher's choice*"(Computing At Schools Scotland, 2014 ,P.5). The Scottish Qualifications Authority (SQA) simply specifies that an 'appropriate' programming language be used in completing it assessments (Scottish Qualifications Authority, 2015g, 2015h).

The SQA does however specify a formal reference language called 'Haggis'(CAS Scotland, 2014) which is used in its assessment papers to provide examination questions for students independent of formally defined languages (e.g. programming languages). Students can answer these questions asked using the Haggis reference language in the language they are most familiar and comfortable with.

Students in early secondary in general use the block based drag and drop educational programming environments which allow for the development of rich multimedia applications ('Scratch') and mobile apps ('App Inventor').In lower second level, students use more conventional text based languages such as 'LiveCode' which allows cross platform development and deployment of rich applications (Scott & Bundy, 2015).

Anecdotally[6] from reviewing the forums and content areas of the Computing at School Scotland website[7], it can be gathered that Python, PHP and Java (using Greenfoot educational programming environment) are all popular languages in the upper second level.

New Zealand. "*The choice of programming language has been left to teachers, and this has been useful since they can choose languages that will work in their context*"(T. Bell et al., 2014 ,P.14). Teachers can use any appropriate language they wish if it meets a rationale of providing sufficient constructs to cover the content in each level of achievement standard.

In the first year of its upper second level programme, "*the programming language could be graphical, drag-and-drop or text based. The language chosen must support the required data types and procedural structures, and good comment facilities*"(New Zealand Qualifications Authority, 2015c ,P.1). 'Scratch' is the most popular language at this level, with a survey

[6] Research ongoing

[7] http://www.cas.scot/

conducted among teachers implementing the new ‘Programming and Computer Science’ standard finding that at Level 1: 73% were using Scratch, 15% Visual Basic, 13% Alice and 6% were using Python.

In its second year, “*It is preferable for the programming language for this standard to be text-based. Any language chosen must support indexed data structures, modules with parameters, global and local scope of variables and good comment/document facilities*”(New Zealand Qualifications Authority, 2015e ,P.2).

And in the final third year, “*The programming language for this standard must be a text-based Object-oriented programming language that supports graphical user interfaces (GUIs) and event based programming” (New Zealand Qualifications Authority, 2014b ,P.2)*

The most popular programming language used by teachers for year two and above was Python with 78% of teachers reporting using it in the 2013 survey(Thompson & Bell), with 25% reporting using Java, 18% VB, 10% Alice, and 10% C#. (note multiple languages were used by many teachers).

Ontario. The Ontario Curriculum for Computer Studies specifies that “*The computer programming language used and the order in which concepts are taught is left to the teacher’s professional judgement. Some teachers may decide to use an “objects-first” approach, while others may prefer to use structural program techniques to teach the same concepts*”(2008 ,P.20). It specifies that ‘Industry Standard’ languages should be used, rather than individual programming languages.

The rationale that the language be ‘Industry Standard’ has led to several Integrated Development Environments(IDE) being chosen by different high schools, with a decision

made based on a teacher(s) preference and experience. Anecdotally[8], a review of on-line forums[9] and blogs[10] used by Ontario Computer Studies teachers suggests that Java is the most popular language, with Java IDE's Eclipse[11], DRJava[12], BlueJ[13], and Netbeans for BlueJ[14] prominent. C++, C# and Visual Basic are also used in more limited cases.

'Alice'[15] and particularly Scratch educational programming environments were prominent in their use for teaching students from late primary school onwards in the fundamentals of Computer Studies.

Israel. The CS curriculum in Israeli schools is designed to be largely independent of any specific programming language, reflecting a central axiom of the programme that it "*should provide insight, knowledge, and skills independent of specific computers and programming languages*" (Gal-Ezer et al., 1995 ,P.73). This agnostic approach to the choice/use of a programming language(s) leaves it open to individual schools to implement their CS programme in any language (as none is imposed)(Gal-Ezer & Harel, 1999). However in actuality schools generally use the same programming language due to the availability of language specific course material from the Ministry of Education.

The first programming language ("Mother Tongue") taught in Israeli High Schools has evolved since the introduction of the programme in the late nineties to reflect changes in technologies and underlying programming paradigms. The programme started with a "vanilla" procedural approach using 'Pascal' and later 'C' programming languages, before

[8] Research ongoing
[9] http://www.acse.net/software
[10] https://www.teachontario.ca/
[11] http://www.eclipse.org/
[12] http://drjava.org/
[13] http://www.bluej.org/
[14] https://edu.netbeans.org/bluej/
[15] http://www.alice.org/index.php

changing to a higher-level Object Orientated language ‘C++’ in the 2000s, with ‘C#’ and especially ‘Java’ being used currently. Two of the key contributing authors of the original Israeli CS curriculum, Judith Gal-Ezer & David Harel, describe the rationale behind the choices of language used in the programmes introduction, as “*to remain in the conservative mainstream*” (1999 ,P.4).

The Israeli high school CS programme also specifies as a principle that its students should ideally be introduced to a second programming paradigm of a radically different nature, albeit on a humbler scale. Appropriate languages are chosen to implement “programming paradigm such as Logic programming, or a more application-focused perspective such as computer graphics or management information systems” (Gal-Ezer & Stephenson, 2014 ,P.8).

In 2011, the Israeli Ministry of Education launched its STEP programme (Science and Technology Excellence Program) which introduced CS into its lower second level (ages 12 - 15). They considered the following principles when choosing a suitable programming environment that exposes students to the fundamentals of Computational Thinking and programming: (Zur Bargury et al., 2012 ,P.4)

- *Expose the students to algorithmic problems and their solutions and enhance algorithmic thinking.*
- *Enable students to implement various control structures.*
- *Make programming enjoyable, interactive, easy to use, and graphically appealing.*
- *Be translated to different natural languages.*
- *Be free of charge (if possible).*

Based on these criteria, ‘Scratch’[16] was chosen as the appropriate environment.

[16] https://scratch.mit.edu/

Summary of Findings. Across each of the jurisdictions, the choice of programming language at upper second level was decided locally. In principle, this allowed teachers to choose languages that fitted their own context, however in practice the choice of languages very often came down to the availability of supporting materials and course resources for a language, a teacher's own knowledge and experience of a language, and/or the languages supported by assessment authorities.

	Israel	Ontario	New Zealand	Scotland	England
Prescribed	Locally	Locally	Locally	Locally	Locally
Rationale for Choice(s)	"Conservative Mainstream"	Be of an Industry Standard	Enables meeting of Achievement standards	Must be a 'real' language	Meets modality of curriculum
Most popular introductory Language	Scratch	Scratch	Scratch	Scratch	Scratch
Most popular language in upper second level	Java	Java[17]	Python	Java, Python, PHP[18]	Java, Python, Visual Basic

Table 9: Programming Tools & Methods in Selected Jurisdictions

Block based drag and drop educational programming environments that use visual syntax free techniques predominate as the choice for an initial/early student programming experience.

[17] Anecdotal finding
[18] Anecdotal finding

These environments help remove constraints on a student's initial learning by making programming easier to learn, with evidence suggesting that students that use these simpler languages are more successful in progressing to sophisticated languages than students who had not (Hagan & Markham, 2000; Mannila, Peltomäki, & Salakoski, 2006)

Information Technology

To what extent do the courses include:

Evaluating and applying information technology, including new or unfamiliar technologies, to solve problems analytically.

Introduction. There are many labels used within our sampled jurisdictions (and globally) that could be deemed to encapsulate this research question. Terminology such as Information Technology (IT), Information Communication Technology (ICT) and Computing are used interchangeably across jurisdictions and even sometimes within the same jurisdiction. This phenomenon coupled with the variability of the content prescribed and indeed taught, makes it difficult to make relevant and viable comparisons across curricula (Furber, 2012; Hubwieser, Giannakos, et al., 2015; Jones, 2011; Webb et al., 2016). The observations within this section of the report are limited to where a jurisdictions' curricula explicitly refer to the "use" of computers to solve problems analytically as the primary learning objective.

England. Prior to the curriculum reforms in 2014 which introduced GCSE, AS and A-level qualification in Computer Science(CS), English secondary schools offered an Information and Communication Technology (ICT) and Computing curriculum in these grades.

The provision of ICT as a subject in the English curriculum was deemed in the Furber report to the Royal Society (2012) as "counter-productive", due to its unbalanced focus on the "use" of Information Technology at the expense of a deeper understanding of the underlying CS concepts. The reforms that followed led to the discontinuation of ICT and Computing as a

GCSE and A Level subject to be replaced by a CS curriculum that has an emphasis on greater technical and conceptual themes. Revealingly the term ‘ICT’ is completely absent from the specification documents of the English examination boards that offer CS A-levels. However, key aspects of the “old ICT curriculum” have not been dropped with the expectation that students will evaluate and apply information technology to solve problems analytically in the course of their CS studies (Webb et al., 2016).

Vocational qualifications are also available in the English education system that offer coursework-based certification in Computing and ICT, such as those offered by the Business and Technology Education Council (BTEC).

Scotland. The experiences and outcomes specified for CS contexts in the curriculum for excellence stipulates that a student can say: (Education Scotland, NA)

- *Through research, I can gain knowledge of computer systems or emerging technologies to understand their differing features and consider their suitability for the world of work.*
- *By learning the basic principles of a programming language or control technology, I can design a solution to a scenario, implement it and evaluate its success.*
- *Using appropriate software, I can work collaboratively to design an interesting and entertaining game which incorporates a form of control technology or interactive multimedia.*
- *I can create graphics and animations using appropriate software which utilise my skills and knowledge of the application.*
- *Having gained knowledge and understanding of the components of a computer, I can make an informed choice when deciding on the system required for a specific purpose.*

These guidelines are manifest in the stated aims of the CS Highers & Advanced Highers curriculum specifications for its ‘Software Design and Development’ and’ Information

System Design and Development' courses. Both courses are expected to enable learners to: (Scottish Qualifications Authority, 2016a)

- *apply skills and knowledge in analysis, design, development, implementation and evaluation to a range of digital solutions with increasingly complex aspects.*
- *apply creative problem-solving skills across a range of contexts.*

New Zealand. The technology curriculum in New Zealand (NZ) has a global educational rationale that students should learn to ***"****apply practical skills and discipline knowledge in an activity-based, project-driven environment in which they solve problems and create innovative solutions for real needs"*(Ministry of Education, New Zealand, 2016c).

Programming and CS make up 8 standards (modules)of the 35 in the Digital Technologies (DT) specialist strand. As a year's course in DT consists of an average of 4 to 5 standards, students taking Programming and CS standards also select from other DT components such as digital media, web programming, embedded systems, robotics and information management. In this manner students learn to use information technology to solve problems analytically, both as part of the Programming & CS standards and in a contextualised manner through the wider digital technologies field.

The NZ curriculum also contains unit standards (shorter duration), offered predominately by the polytechnic and private training sector, where students gain qualifications based on competency in learning practical skills to solve problems. These unit standards include ICT courses leading to National Certificates for progression into vocational and college pathways.

Ontario. A key goal of the Ontario computer studies program is to enable students to "*apply the knowledge, skills, and attitudes acquired through the study of computers to a variety of learning tasks and relate them to computer phenomena on the local, national, and global levels;* ", reflecting its fundamental purpose of enabling students to "*achieve success in secondary school, the workplace, postsecondary education or training, and daily life*" (Ministry of Education, Ontario, 2008 ,P.4)

To implement this central purpose, the ministry offers three programmes (Open, University preparation, and College / Vocational preparation) which differ in the level of applied and practical content provisioned in each. Each of the three streams contain Software Development (including project management and software engineering principles) as one of the four critical areas of learning.

- Open Stream: *'Introduction to Computer Studies' in Grade 10 – purpose is to broaden student's knowledge and skills in computing. Contains an 'Introduction to Programming' and 'Understanding Computers' component.*
- College Stream: *'Introduction to Computer Programming' in Grade 11 & 'Computer programming' in grade 12. Focus on practical and applied course elements with a focus on software development, and programming skills & environments.*
- University Stream: '*Introduction to Computer Science' in Grade 11 & Computer Science' in Grade 12. Focus on conceptual and underlying aspects of computing, applying these learning in part through its Software Development component*.

The Ontario curriculum also recognises the learning value that ICT provides both to teachers in their instructional strategies and to students who can use appropriate ICT tools "*to collect, organize, and sort the data they gather and to write, edit, and present reports on their findings*"(Ministry of Education, Ontario, 2008 ,P.20).

Israel. A principle of the Israeli CS curriculum is that "*Conceptual and Experimental issues should be interwoven throughout the program*" (Gal-Ezer et al, 1995, P.75) . This two-tracked approach called the "zipper principle" ensures that students "use" information technology throughout their CS studies to solve problems analytically in an applied manner. Given this applied nature of curriculum, it is interesting that an academic paper authored by two of the key architects of the Israeli CS curriculum (Judith Gal-Ezer and David Harel) entitled *"Curriculum and Course Syllabi for a High-School Program in Computer Science"* does not use the term "information technology" or "Information and Communications Technology (ICT)" anywhere in its 31 pages(1999).

Israeli high schools also offer a software engineering track which extends the CS programme to its upper second level students. Students that elect to take this track study an additional five units in the form of a large project, which are taken in conjunction with CS as well as a natural science subject. Currently this track offers seven alternative courses: graphic systems, information systems, expert systems, operating systems, web services, mobile programming, and cyber defence systems. In the completion of their large project students apply their learnt skills in a specific context to solve a defined problem in an analytical manner. This advanced track only attracts "a few hundred students each year."(Armoni, 2016; Haberman & Cohen, 2007)

Summary of Findings. The teaching of skills required for the effective "use" of computers to solve problems analytically is recognised across the sampled jurisdictions as important. It is most often captured in the curricula as an underlying and implicit global goal.

The "use" of computers is nearly ubiquitous as an applied learning mechanism employed in teaching computing subjects and indeed increasingly across all school disciplines. Where the

"use" of computers is explicitly detailed in the curriculum, it is most often in the provision of vocational type courses as direct career pathways.

Digital Literacy

To what extent do the courses include: Being responsible, competent, confident and creative users of information and communication technology.

Introduction. In addressing the research question, Digital Literacy(DL) as defined by Simon Furber in his seminal report to the Royal Academy is used:

> *"the basic skill or ability to use a computer confidently, safely and effectively, including: the ability to use office software such as word processors, email and presentation software, the ability to create and edit images, audio and video, and the ability to use a web browser and internet search engines. These are the skills that teachers of other subjects at secondary school should be able to assume that their pupils have, as an analogue of being able to read and write"* (2012 ,P.17).

DL is not offered in the reports jurisdictions as a separate specialist subject, but expected to be interwoven into all subjects on their curricula. DL is considered a major medium and methodology for student learning in every subject that they study (Cheema & Zhang, 2013; Webb et al., 2016), while also important as a complementary educational precursor to the separate but overlapping specialties of Information Technology(IT) and Computer Science(CS) (Furber, 2012; Snyder, 2012).

The joint informatics Europe & ACM Europe Working Group recommends that "*All students should benefit from education in digital literacy, starting from an early age and mastering*

the basic concepts by age 12. Digital literacy education should emphasize not only skills but also the principles and practices of using them effectively and ethically"(2013 ,P.3).

England. One of the central aims of the new 'National curriculum for computing programmes of study' introduced by the Department for Education, is to ensure that all its pupils from key stage 1 to 4 "*are responsible, competent, confident and creative users of information and communication technology*".

To reflect this aim in its upper second level, each student should be taught to "*understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns"* (Department of Education, 2013).

These aims are manifest in the A- level content specified by the various examination boards, both implicitly in the conduct of their various CS courses and explicitly through the provision of courses such as: 'Consequences of uses of computing' (AQA), and 'Legal, moral, ethical and cultural issues'(OCR).

The expectation is that each student will have attained a high level of DL by Key Stage 4 (age 14 – 16), through the provision of new courses of instruction (such as the GCSE in CS) and through a "position that digital literacy is to be embedded in every subject" (Hubwieser, Giannakos, et al., 2015 ,P.72).

Scotland. Education Scotland provide a set of guidelines on the expected experience and outcomes students should acquire over each key stage from its CS curriculum (Education Scotland, NA). Table 5 details the guidelines that relate to DL attainment for its key stage 4 / upper second level.

Key Stage	Guidelines for Skills & Knowledge
Stage 4 upper second level	• I can work with others to plan and use a learning group for sharing experiences, ideas and information within a secure online environment. • I can compare different forms of security software to gain knowledge and understanding of their functions in protecting contemporary technologies. • Through research, I can gain knowledge of computer systems or emerging technologies to understand their differing features and consider their suitability for the world of work. • I can create graphics and animations using appropriate software which utilise my skills and knowledge of the application • I can use features of software to create my own animation which can then be used to create an animated sequence.

Table 10: Scottish Curriculum Selected Experiences & Outcomes in Digital Literacy

The expectations detailed in Table 5 are integrated into course content, delivered implicitly cross-discipline with the use of ICT for learning, and explicitly in specific course aims. In Highers and Advanced Highers CS these include aims to enable learners to: (Scottish Qualifications Authority, 2016a)

- *develop an informed understanding of the role and impact of computing technologies in transforming and influencing our environment and society.*
- *develop awareness of current trends in computing technologies.*

New Zealand. The term Digital Literacy is not explicitly used within the learning objectives, indicators of progression, and teacher guidance in the NZ Digital Technologies curriculum. However, it is an underlying expectation that students would have a reasonable level of the core competency of DL prior to entering Year 11 in High School(Grimsey & Phillipps, 2008; Ministry of Education, New Zealand, 2009).The expectation for learners was that they would gain these skills through cross discipline use of ICT tools and applications (e-learning).

More recently, the Ministry of Education spurred on by its 2014 “Curious Minds” initiative, announced in July 2016 that it is to fully integrate Digital Technology into the NZ Curriculum and Te Marautanga o Aotearoa’ by 2018 (Ministry of Education, New Zealand, 2016a). The provision of DL and Computational Thinking throughout the early school curriculum (Year 1 – 10) is central to the consultation process for this initiative (Webb et al., 2016).

Ontario. “*The fundamental purpose of the computer studies program is to provide students with knowledge, skills, and attitudes that will enable them to achieve success in secondary school, the workplace, postsecondary education or training, and daily life*”, with a programme goal to enable students to “*develop lifelong learning habits that will help them adapt to computer advances in the changing workplace and world*” (Ministry of Education, Ontario, 2008 ,P.4).

This central statement of purpose is reinforced by the inclusion of ‘Professional and ethical responsibility‘ as one of the four critical areas for knowledge and skill development across the Ontario Computer Studies curriculum. In each of the Computer Studies strands (CS and Computer Programming) students take DL courses in:

- Grade 10: Introduction to Computer Studies: ‘Computers and Society’
- Grade 11 & 12: Computer Science stream includes topics such as environmental stewardship and sustainability.
- Grade 11& 12: Computer Programming stream includes topics on the role of computers in society, the ethical use of computers, and career exploration.

Israel. Every Israeli high school students is entitled to be *“instructed in digital and computer literacy as a major medium and methodology that contributes to their learning in all the subjects that they study*”(Webb et al., 2016 ,P.9). DL is taught within the Israeli curriculum across all subjects in the curriculum including CS, rather than as a discreet subject.

Summary of Findings. Digital Literacy is considered an important global educational goal/objective within each of the research jurisdictions, rather than as a discrete subject. Across the curricula, it is very often taught in conjunction with other subjects, a pedagogical approach that is becoming ubiquitous across Europe and further afield.

More recent curriculum revision in England (2014) and Scotland (2012) contain detailed goals and objectives for students in Digital Literacy, reflecting in part an increased awareness and emphasis in these core skills for all students.

Digital Literacies are knowledge and skills that are acquired over time, with many studies suggesting that children as young as five can learn and benefit from age appropriate teaching in this area (Tim Bell et al., 2014). Internationally several progressive educational systems have moved to introduce Digital Literacy and Computing into their primary and early secondary school curriculums on a cross disciplinary basis (i.e. Scotland & England) or in the process of doing so (i.e. New Zealand).

Section Three

Nature of the Subject

Participation

> Who takes the subject, is it offered as a choice or as part of a core subject. If it is an option what % of the students take it?

Introduction. Except for New Zealand where Computer Science(CS) is a part of the Digital Technologies programme, each of our jurisdictions offers CS as an optional and discrete subject to students as they enter their upper second level. As CS is an optional subject, analysing the level of student participation in each of our jurisdictions gives a valuable indication of the popularity of the subject.

Many of our research jurisdictions are too early in their transition of offering CS (i.e. New Zealand, Scotland & England) to draw firm conclusions from our statistical analysis. However, it is possible to observe early trends.

Israel as the longest established programme, offers a longer-term view on CS adoption. Ontario is also a long established programme but the absence of publicly available data relevant to the research question meant that no analysis of that jurisdiction was possible

During the analysis, participation in CS was compared with the natural sciences of Physics, Biology & Chemistry (as recorded data permitted). This enables the reader to make relative comparisons with similarly elective subjects that would be considered attractive to the same cohort of students.

England. The revision of the English curriculum to ‘disapply’ ICT and introduce the new CS curriculum has only taken place since 2014. Since then there has been a gradual transition by the teachers, schools and the examination boards, with the ICT qualifications no longer being offered to students from September 2016.

The need for revision was urgent, Brown et al. noted that “*Computer science in UK schools is a subject in decline: the ratio of Computing to Maths A-Level students (i.e. ages 16-18) has fallen from 1:2 in 2003 to 1:20 in 2011 and in 2012. In 2011 and again in 2012, the ratio for female students was 1:100, with less than 300 female students taking Computing A-Level in the whole of the UK each year*” (2013, P.1). It is hoped that as the new curriculum gains traction, coupled with the new GCSE qualification offered pre- upper second level, that this will in time lead to a vibrant CS discipline in English schools.

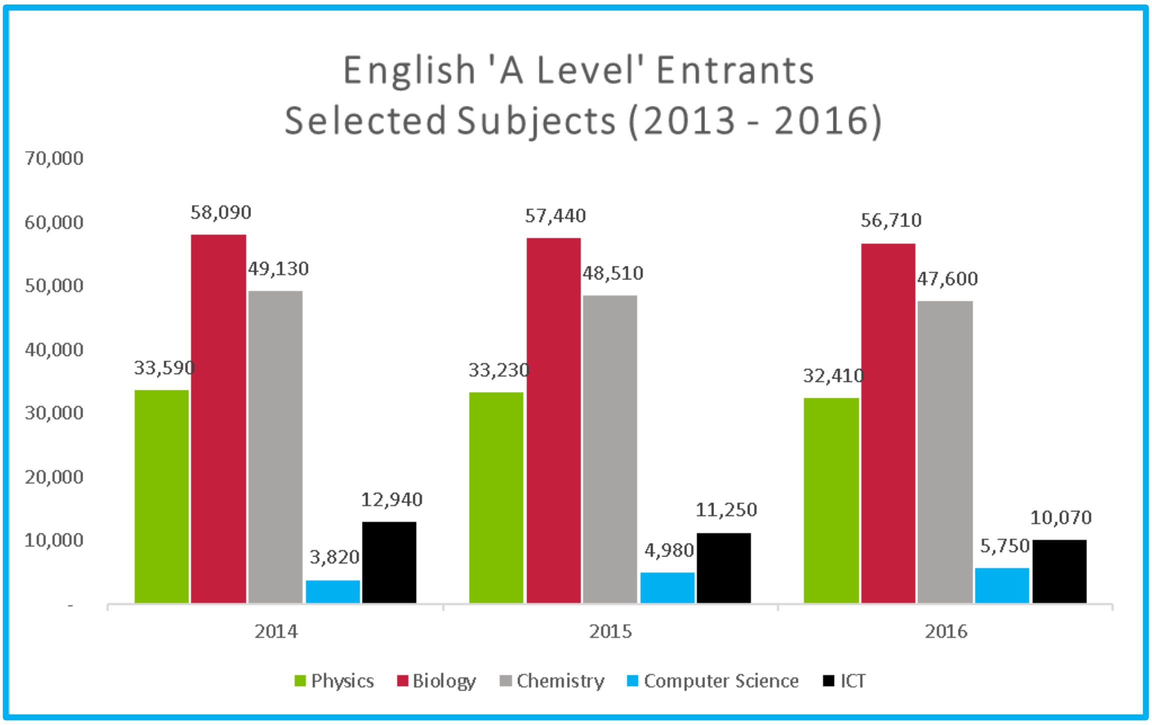

Figure 2: English ‘A Level’ Entrants Selected Subjects

Figure 2 shows the number of students taking CS for their A-levels between 2014 and 2016 in comparison to the natural sciences of Physics, Biology and Chemistry (Data from: The Office of Qualifications and Examinations Regulation, 2016a).

As statistics for total number of English students taking each of the national qualifications (A Level, GCSE etc.) each year are not publicly available, Figures 2 and 3 present the available data namely the number of entrants for each subject. A recent analysis from Roehampton University researchers established that in 2015 the figure of 4,980 CS entrants equalled 1.7% of A-Level candidates and 34,540 entrants to 5.5% of GSCE candidates (Kemp, Wong, & Berry, 2016).

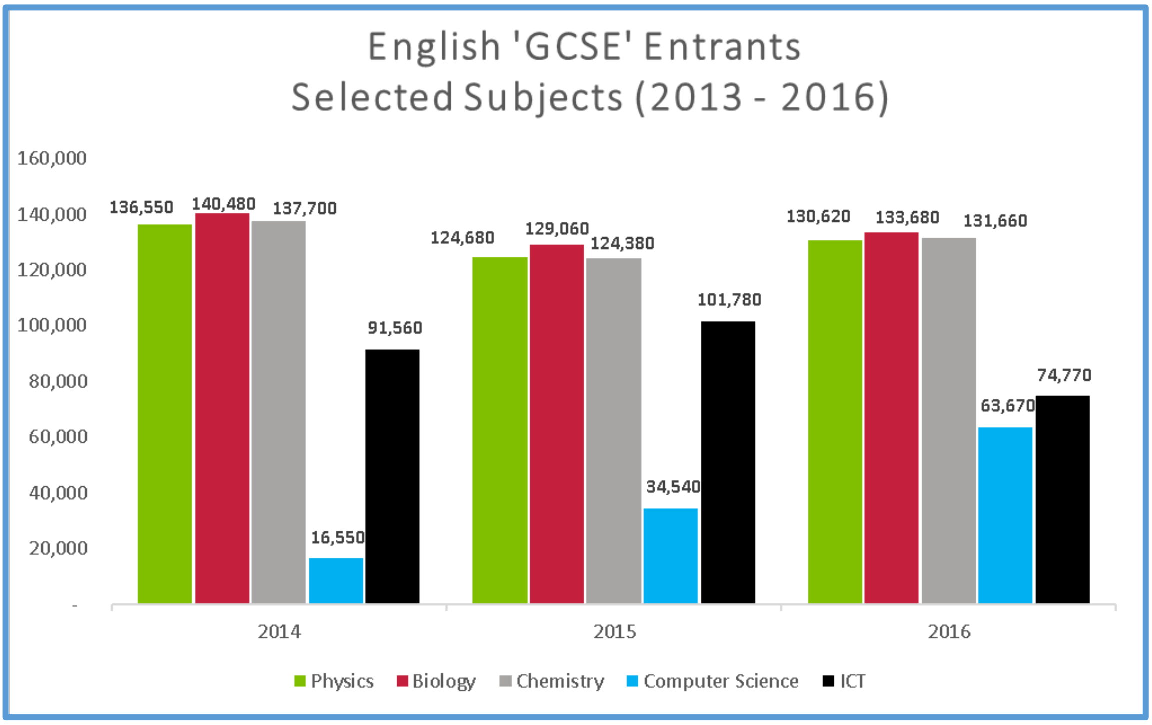

Figure 3: English 'GCSE' Entrants Selected Subjects

Figure 3 shows the percentage of students taking the new CS 'GCSE', the number of students taking the new qualification has a positive upward trend from albeit from a small base (Ofqual, 2016a).

Scotland. The number of students taking Computing Science(CS) in their upper second level qualifications of Advanced Highers and Highers are relatively low when compared to the natural sciences (Figure 4 and Figure 5)(Data from: Scottish Qualifications Authority, 2014; 2016b).

As the revision of the curriculum to specify the new subject of CS is very recent, it is too early to tell what impact on student numbers this revision may have in the medium to longer term.

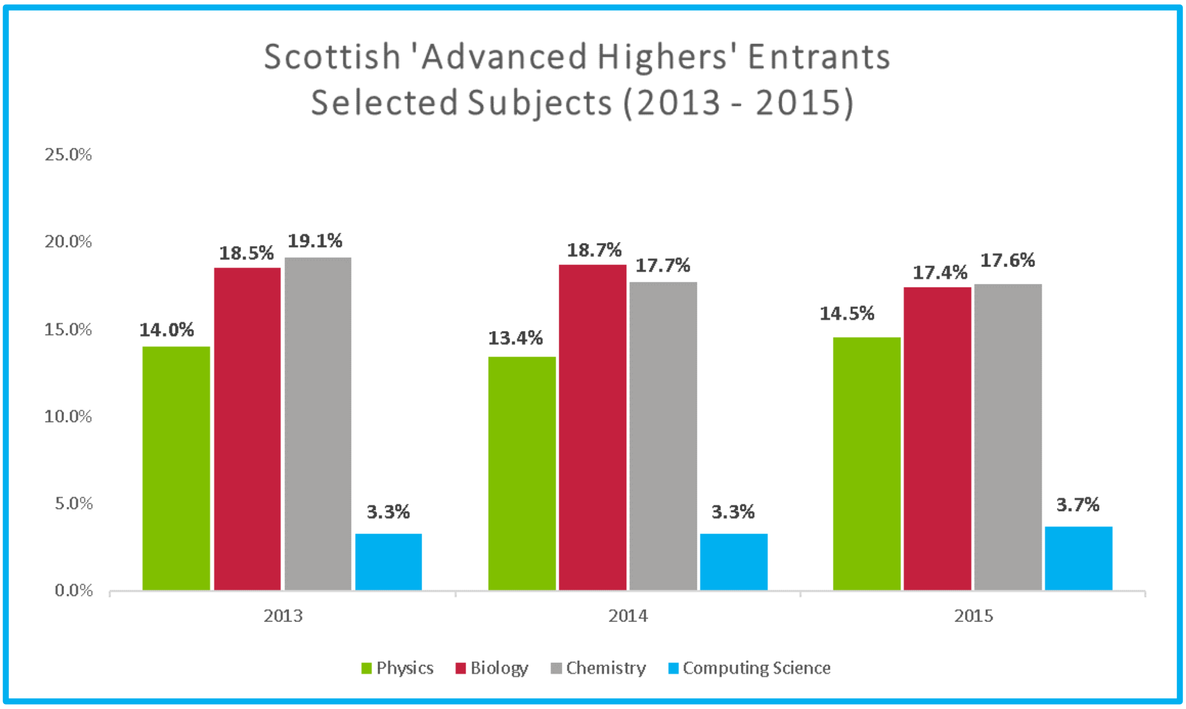

Figure 4: Scottish 'Advanced Highers' % of Total Entrants Selected Subjects

A 2015 briefing document by Computers at Schools Scotland (CASS) reports that the subject discipline is experiencing several difficulties which largely relate to structural and resourcing problems within Scottish schools, many of which predate the introduction of CS onto the curriculum. These difficulties include the closing of school CS departments, the existence of negative student perceptions of CS based on the previous curriculum as boring (Mitchell,

Purchase, & Hamer, 2009), number of CS teachers in schools dropping, and the failure to attract new CS teacher candidates. All of these factors are restricting the students opportunity to take the subject(Computing at School Scotland, 2015).

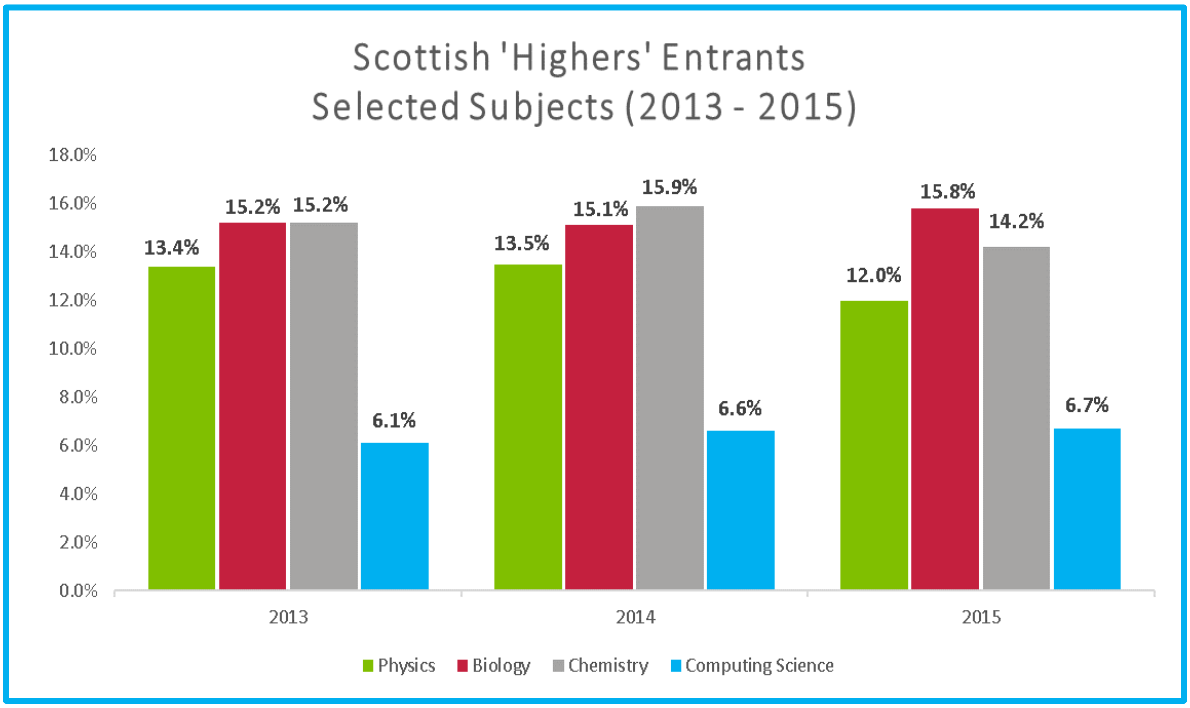

Figure 5: Scottish ‘Highers’ % of Total Entrants Selected Subjects

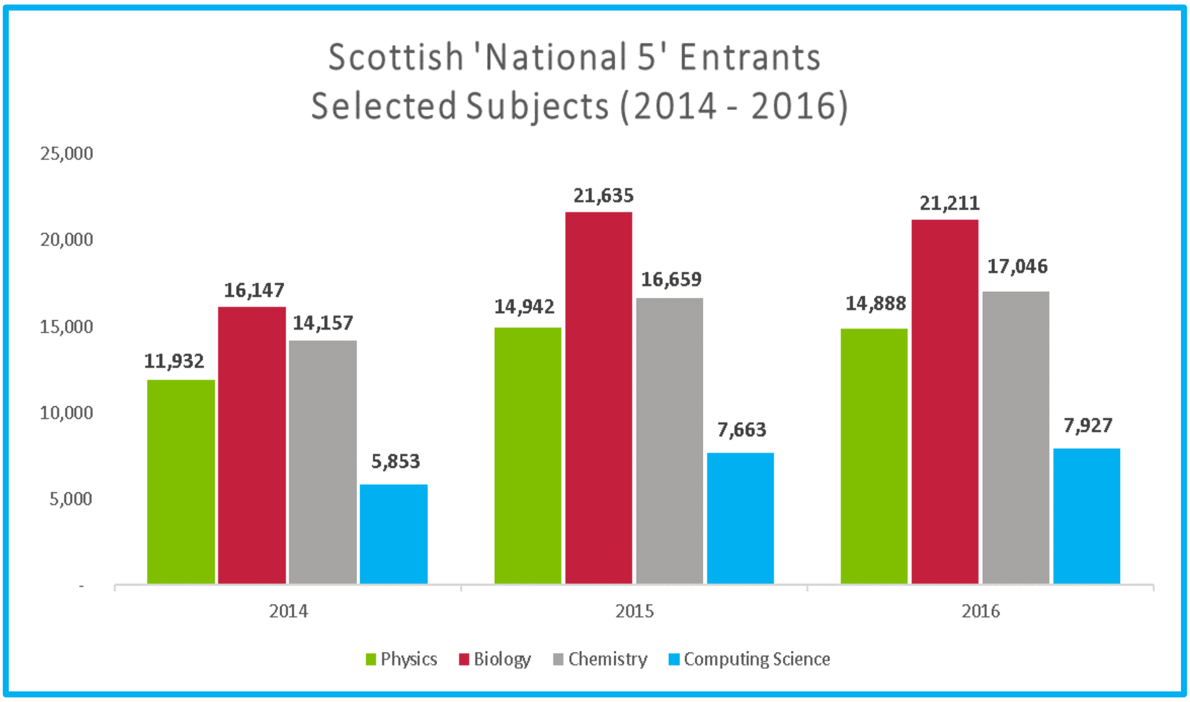

Figure 6: Scottish 'National 5's' Entrants Selected Subjects[19]

The introduction of CS as a subject in 2014 into the Scottish lower secondary curriculum (ages 12 -15) may in time increase the numbers taking CS in its upper second level (Figure 6) (Data from: Scottish Qualifications Authority, 2014; 2016b).

By providing students with CS instruction in lower second level, it is hoped that the steep learning curve and workload often faced by students taking the subject for the first time will be reduced and that it will encourage more to students to take CS as they enter their upper second level. Introduced in 2014, it is too recent to observe if the inclusion of the CS 'national 5' qualification into Scottish secondary schools will in time provide the scaffolding for CS in upper second level to thrive.

[19] Not expressed in percentages as total number of students taking 'National 5's' not avail for 2016

New Zealand. Our analysis of student participation in the Programming and CS standards introduced into the New Zealand(NZ) senior cycle curriculum in 2014 is limited by the data sets made publicly available by the New Zealand Qualifications Authority (NZQA[20]). Each standard is worth between three and six credits (1 credit = ten hours), with schools combining standards of between eighteen and twenty-four credits of related areas to construct courses. The NZQA provides details on students taking each of the standards in a year, but does not provide data on student choices between each standard. Given that there are 10,335 different standards over three years in the NCEA achievement standards of which the Digital Technologies number 35, this limitation is understandable.

When a school adopts the Digital Technologies programme, a teacher can choose from the 35 available standards to comprise a course of instruction, so whether or not the CS and/or programming standards are introduced will depend a lot on whether a teacher chooses to include them. (Thompson, Bell, Andreae, & Robins, 2013) As the standards are not compulsory, if a teacher lacks confidence in their instruction (especially CS), he or she can avoid its inclusion and still run a computing course.

[20] www.nzqa.govt.nz

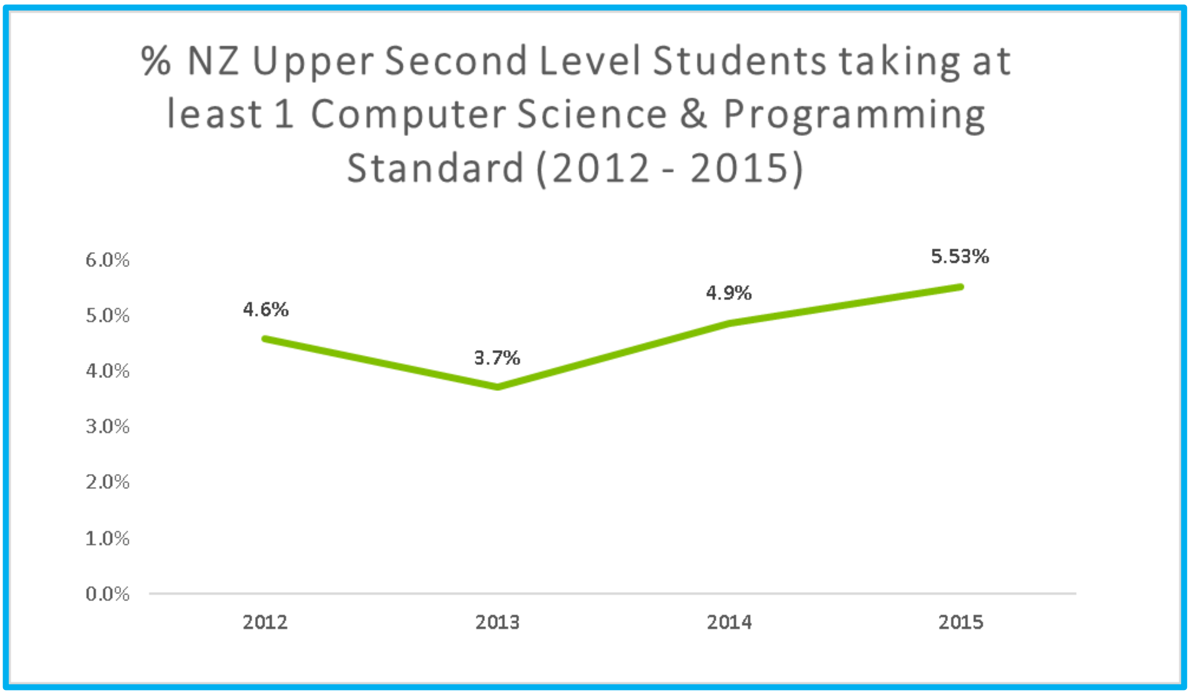

Figure 7: NZ Upper Second Level Students Taking at least 1 CS or Programming Standard

An analysis of the adoption of the new CS and Programming Standards (modules) in NZ, shows a general trend of increasing numbers of students taking at least one CS or Programming Standard since their inclusion in the curriculum in 2012 (Figure 7) (Data from: New Zealand Qualifications Authority, 2012, 2013, 2014c, 2015f).

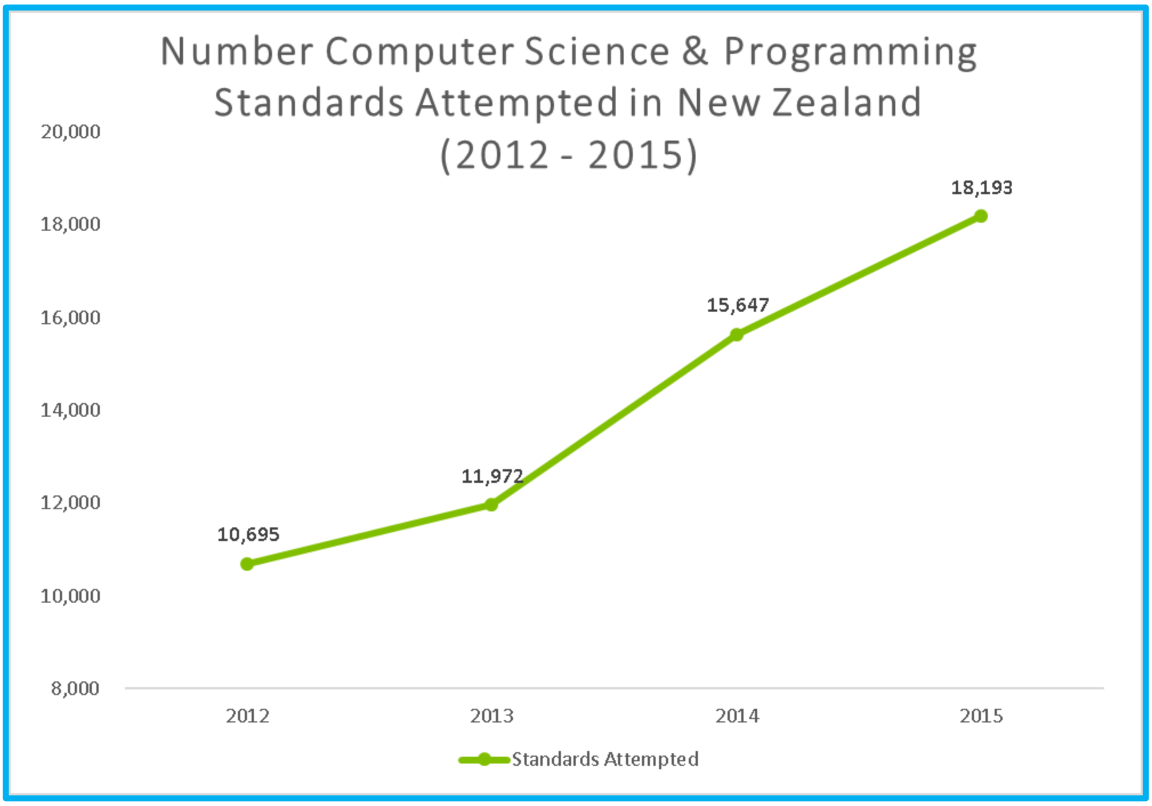

Figure 8: Number Standards Attempted New Zealand Upper Second Level

Figure 8 details the total number CS and Programming standards attempted each year since 2012 (Data from: New Zealand Qualifications Authority, 2012, 2013, 2014c, 2015f). Given the wide variety and crossover of standards in various programmes in NZ secondary schools it was not possible to present a graphic which allowed comparison between CS related subjects and the natural sciences.

Israel. CS is not mandatory in Israeli high schools. Upper second level students can elect to take CS as well as an extensive list of choices. Generally, students preparing for their matriculation exams (Bagrut) decide to focus either on humanities, social sciences, or the sciences. Students who choose to focus on sciences, will normally take an extended mathematics programme (4-5 unit) and then choose one or two scientific courses. (Armoni & Gal-Ezer, 2014). It is notable that while taking physics is a requirement for pursuing

scientific or engineering degrees in third level institutions, CS is not required to pursue its counterpart's degrees in further education.

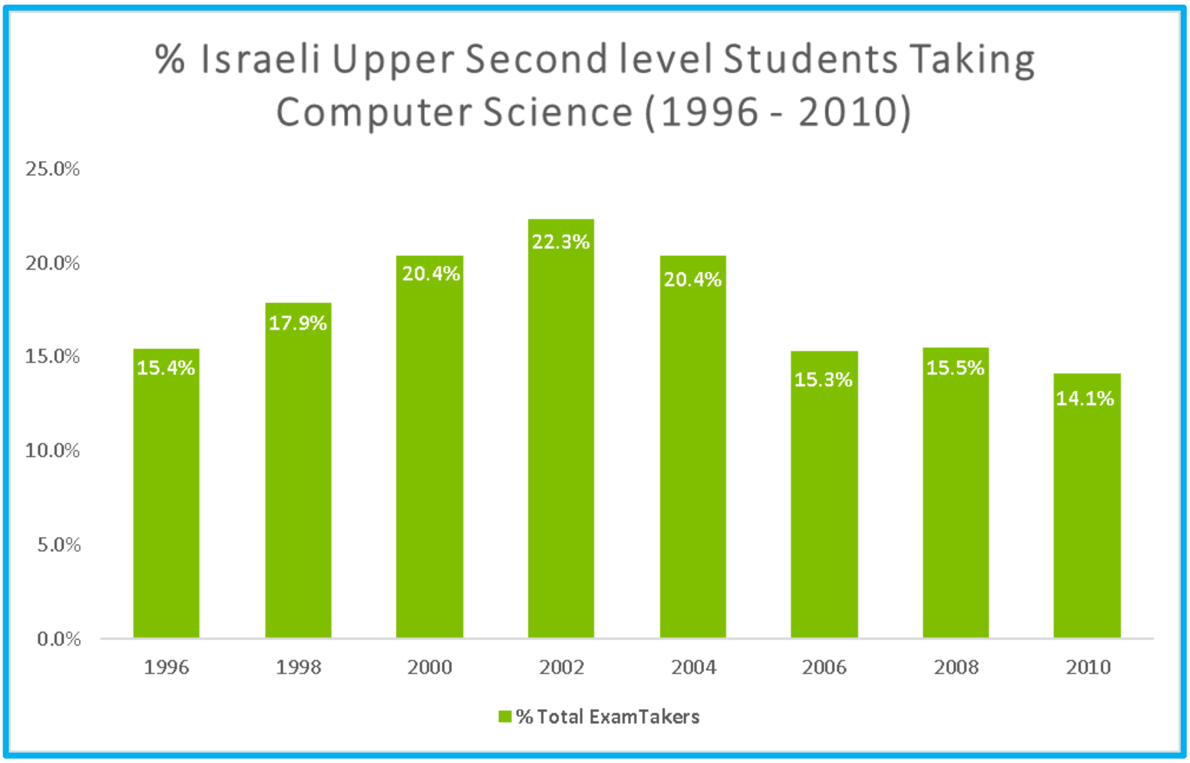

Figure 9: Computer Science Exam Takers of All Exam Takers in Israel

Figure 9 shows the percentage of Israeli students that have taken a CS examination of the total exam taking population (Adapted from: Armoni & Gal-Ezer, 2014, P.107). The average over this sixteen-year period is 18% of the total exam population. Armoni and Gal-Ezer note that the decrease in participation rates of Israel students in CS from 2002 "*may be due to the "dot com bubble" collapse, which no doubt had an effect on what students chose to learn*" (2014 , P.107).

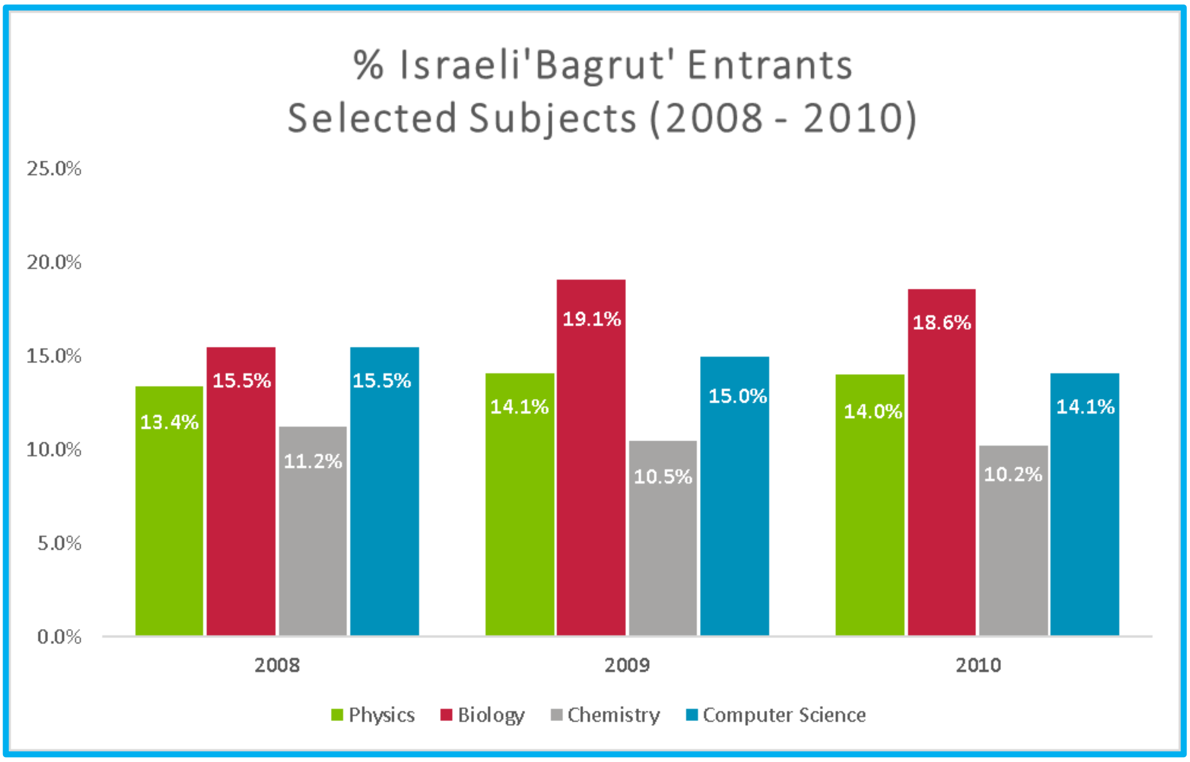

Figure 10: Israeli 'Bagrut' Entrants Selected Subjects

A comparison of the examination participation of upper second level students in CS with their taking of exams in the natural sciences of Physics, Biology and Chemistry (Figure 10), shows that CS is the second most popular subject of the four science subject with Israeli students(Adapted from: Armoni & Gal-Ezer, 2014, P.109).

Summary of Findings. Computer Science is offered as a discrete (except New Zealand) and optional subject across each of our jurisdictions.

The student participation rates in CS in upper second level within the recently revised jurisdictions (England, Scotland & New Zealand) have been low, with early figures indicating an upward trend in many instances. England and Scotland both introduced CS programmes into their lower second level (ages 12 – 15) which have attracted favourable student participation.

As these jurisdictions are still in the process of transitioning from legacy ICT to CS programmes, it is too early to draw conclusions on the longer-term rate of participation of students in the subject.

The longest established CS programme in our report Israel, has an average participation rate of 18% (1996 – 2010), and compares very favourably with student participation in the natural sciences.

Participation by Gender

Participation: What is the gender breakdown?

Introduction. The low level of female participation in technology and computing related fields across industry, academia and society is recognised internationally as an issue, not only from an equality perspective but from an educational, societal and economic viewpoint also. A 2014 Accenture report detailed that within the European Union only 6 to 7 percent of technical careers are being filled by women (2014). Commentators point to a self-perpetuating cycle of smaller percentages of girls in comparison to boys taking technology related subjects in secondary education. This feeds into low participation rates in related third level under graduate and post graduate qualifications, and on into industry. A lack of visible role models in industry feeds into a perception of younger girls that ‘technology & computing’ is a male domain and so the cycle continues.

In this section, an analysis of the participation rate of males and females in upper second level Computer Science (CS) programmes across our research jurisdictions is conducted. This analysis is limited to where the relevant primary or secondary data was publicly available, with two jurisdictions offering only limited data (Israel & New Zealand) and another none (Ontario).

England. English schools are currently transitioning from an ICT curriculum which is being phased out to introduce a new curriculum for CS in their upper second level (Examination boards stopped offering ICT A-levels in September 2016). Accordingly, it is not prudent to draw any conclusions from the data available for this period, but it is possible to observe general trends.

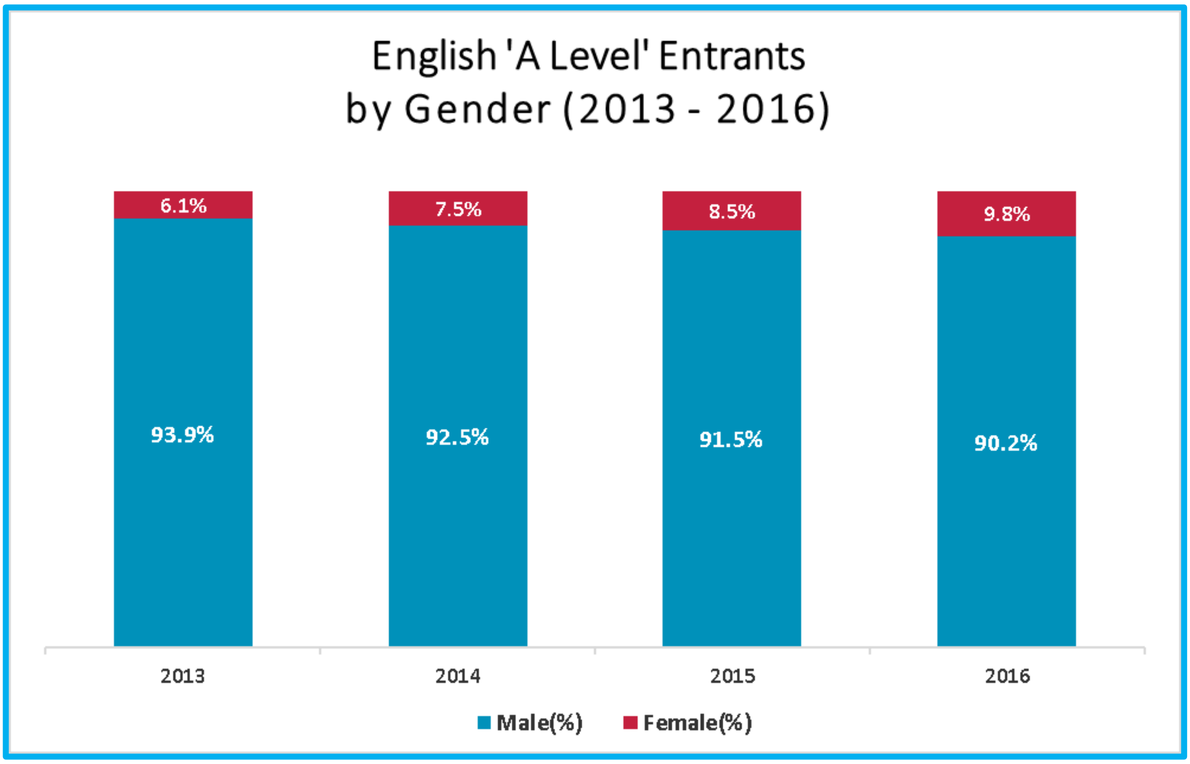

Figure 11: Participation English Computer Science ‘A Levels’ by Gender

Since the introduction of the new CS A-Level qualification, a positive trend is visible in the percentage of females taking the subject albeit from a very low starting point (Figure 11) (Joint Council for Qualifications, 2011, 2012, 2013, 2014a, 2015a, 2016a).

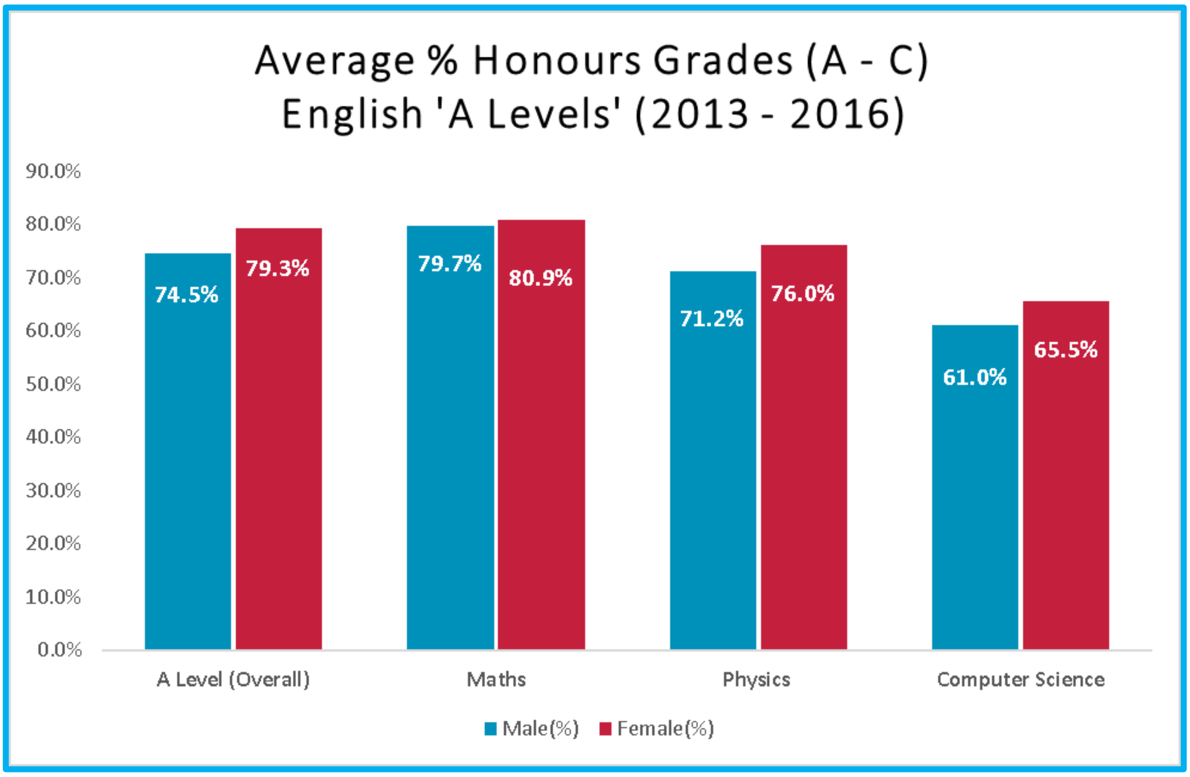

Figure 12: English Students Honours Grades in 'A Levels' by Gender

Figure 12 shows that when females participated in CS A-Levels in the years between 2013 and 2016 they achieved a higher number of honours grades than their male counterparts.

It is also notable in Figure 12 than the percentage of honours grades for both males and females in CS is considerably lower than the percentage for ‘A Level’s’ overall. Recent analysis published by researchers from Roehampton University find that the reasons behind this relative under performance (given student academic profiles) are yet unclear, and may be “due to the relative difficulty of the qualification, teaching in the subject or some combination of these and other factors” (Kemp et al., 2016 , P.3).

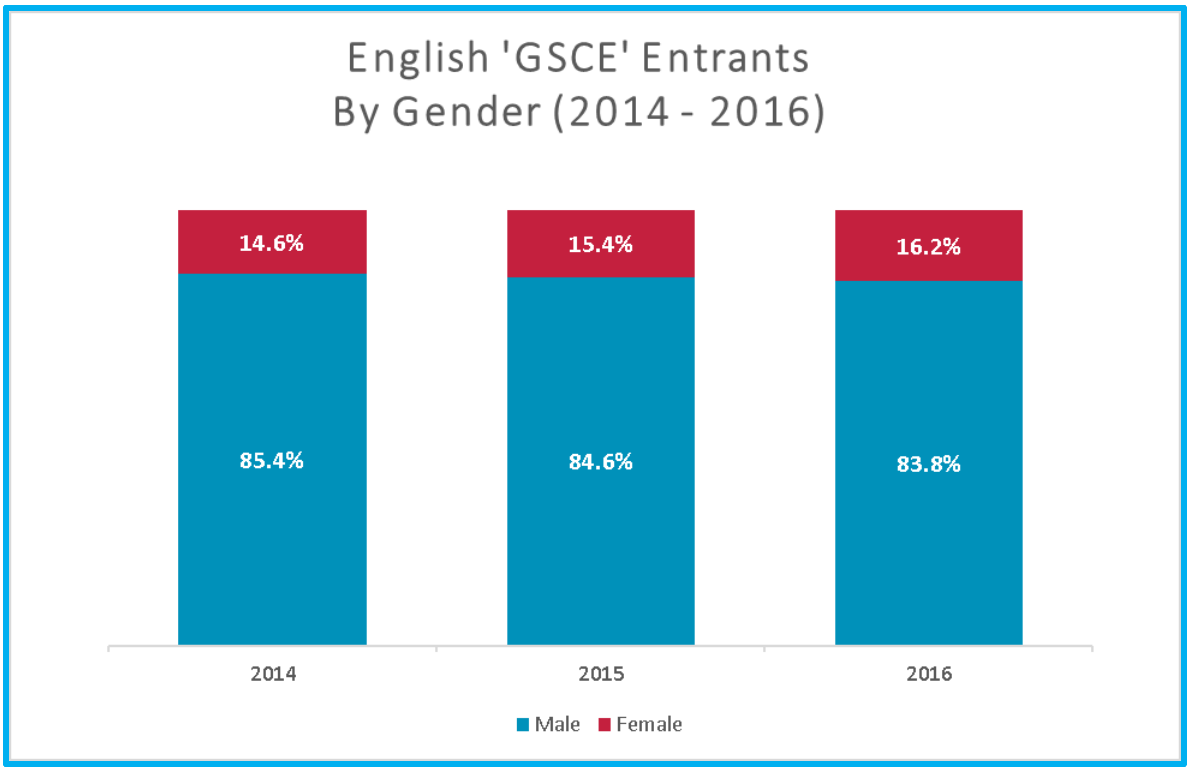

Figure 13: Participation English Computer Science 'GCSE' by Gender

As part of the curriculum revision in 'Computing', a new GCSE in CS has been introduced since 2014. Early indications on the level of participation of females in the new qualification suggest a slight upwardly trend since its introduction, with an average of approximately 15.5% of the entrants being female over the first three years (Figure 13). (Joint Council for Qualifications, 2014b, 2015b, 2016b).

Scotland. The new CS curriculum was only introduced into the Scottish upper second level in 2012, so it is not possible to draw any conclusions from the available data, however early trends may be observed.

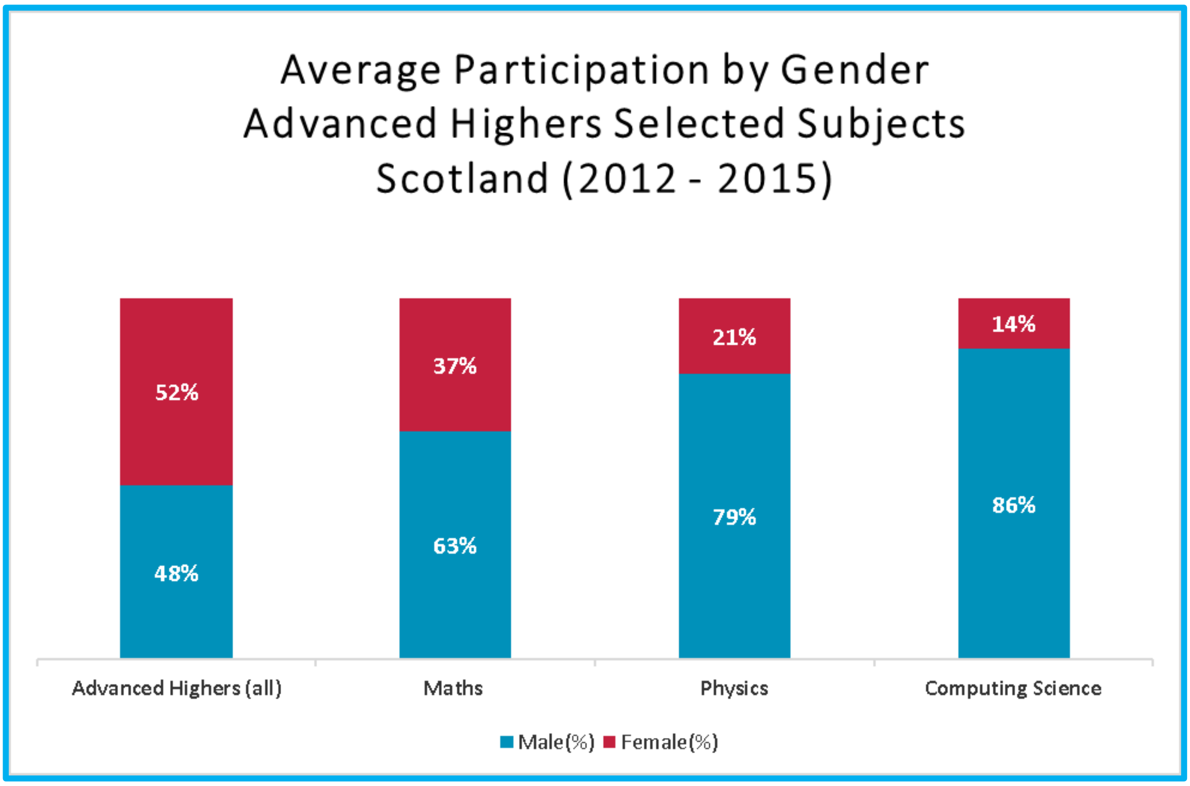

Figure 14: Participation Scottish ‘Advanced Highers’ by Gender

The average participation rate of females attempting an Advanced Highers qualification in CS was 14% over the four years from 2012 to 2015. Figure 14 highlights how this level of participation compares to Mathematics (37%) and Physics (21%), with overall females making up over half (52%) of the students taking Advanced Highers in these years. Figure 15 shows a slightly higher participation rate by females in the Scottish ‘Highers’ qualification with an average of 21% over the years 2012 to 2015 (Scottish Qualifications Authority, 2013a, 2013b, 2015b, 2015c).

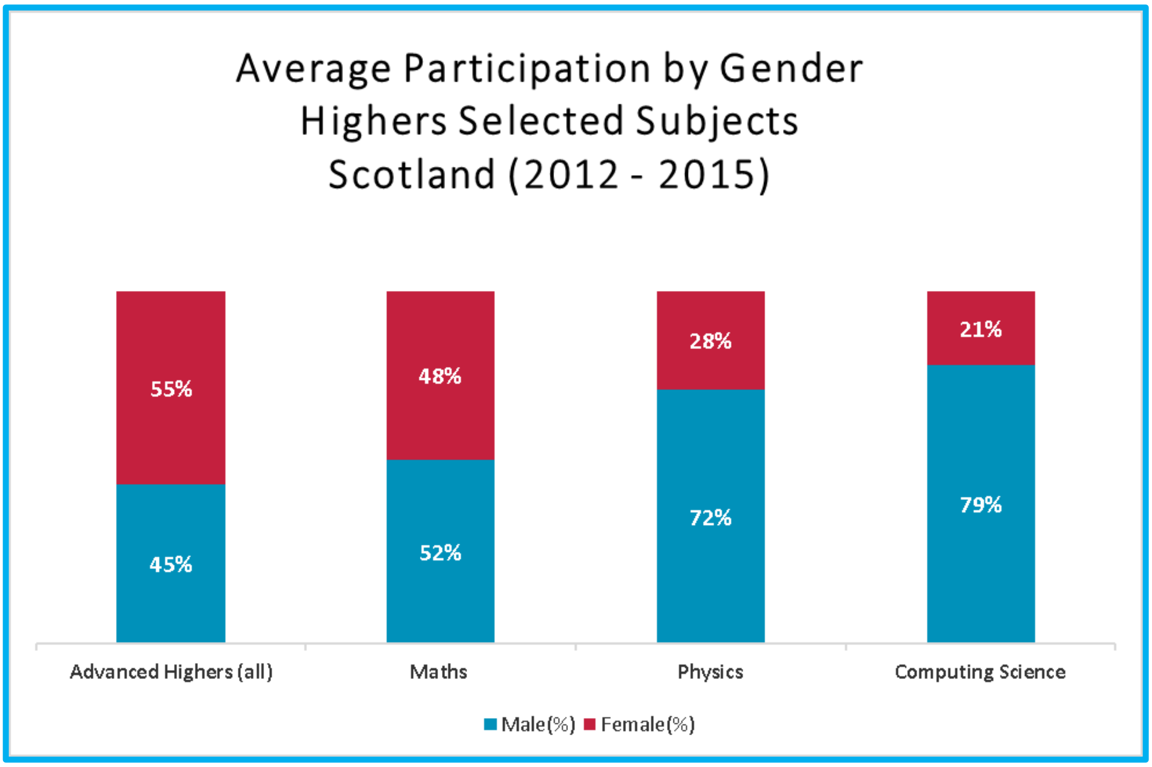

Figure 15: Participation Scottish 'Highers' by Gender

Interestingly the analysis of the data available from the Scottish Qualifications Authority, also reported that a higher percentage of females achieve honour grades than boys across Advanced Highers Qualifications, and do particularly well in CS achieving an honours grade 90% of the time over the four years (Figure 16)(2013a, 2013b, 2015b, 2015c).

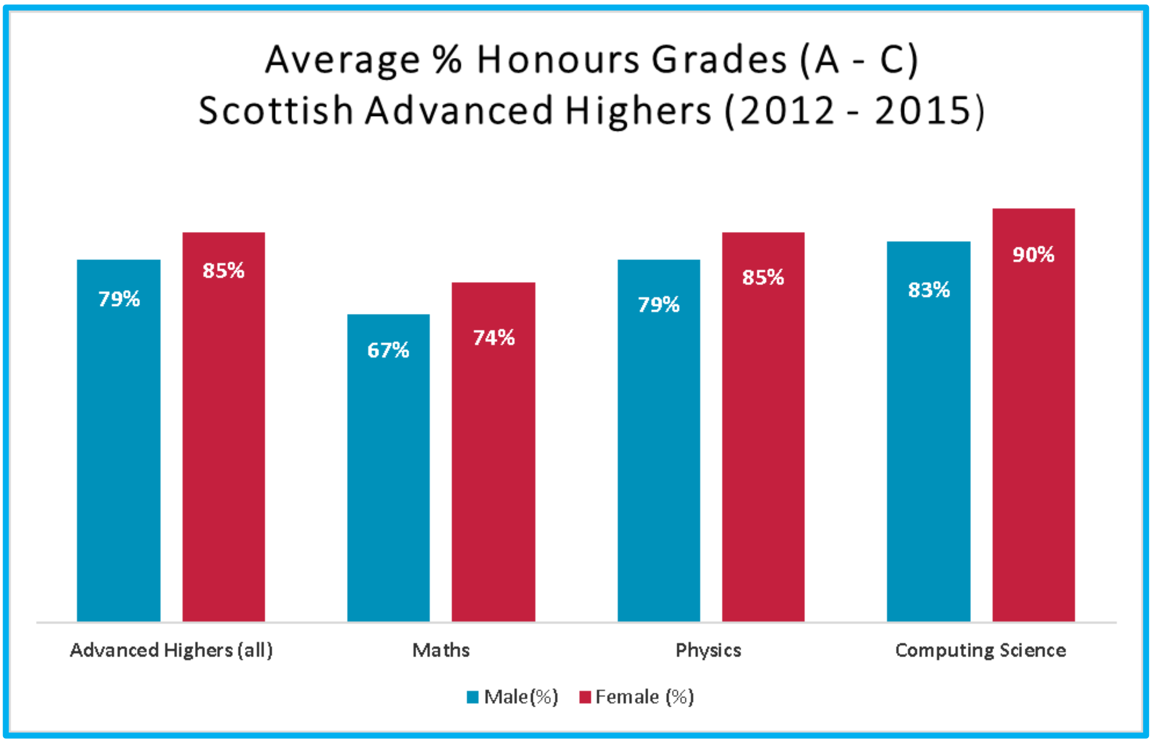

Figure 16: Scottish Students Honours Grades in 'Advanced Highers' by Gender

The first two years (2014-15) of the introduction of a new National 5 qualification in CS which introduces students to the subject before upper second level has an average female participation rate of 22%, a statistic that provides hope for a potential stronger pipeline for female participation in CS in upper second level (Scottish Qualifications Authority, 2015b, 2015c).

New Zealand. The New Zealand Qualifications Authority(NZQA) provide detailed yearly statistics for its national qualifications, however outside of providing aggregated grade results for each standard by gender, it does not provide data to the public on the participation rates of students across the various standards(modules) by gender.

	Candidates	Gender
AS91074 Demonstrate understanding of basic concepts from computer science	654	
Male	475	72.6%
Female	179	27.4%
AS91075 Construct an algorithmic structure for a basic task	2131	
Male	1493	70.1%
Female	638	29.9%
AS91076 Construct a basic computer program for a specified task	3245	
Male	2082	64.2%
Female	1163	35.8%

Figure 17: Participation New Zealand CS & Programming Standards by Gender

As the curriculum changes in New Zealand took place in 2012, it is very early to draw conclusions from the available academic research. Research conducted by Bell et al. in the first year of the introduction of the programming and CS standards found that the ratio was two males to every female in level 1 (Figure 17) (2012).

A 2013 study finding that "*In terms of student gender balance, between 21% and 30% of students enrolled in Level 1 standards are female. While low, it is still above the 10–20% female enrolment rate for CS courses in NZ universities. Achievement levels between genders is slightly better for females, with a mean achievement level of 2.23 for females versus 2.06 for males*"(Thompson & Bell, 2013, P.3).

Further research conducted in 2014 shows a bleaker picture of female participation across the new standards, with a substantial fall off in the numbers of girls continuing from level 1 modules into level 2 & 3 (less than 5% that attempted level 1 programming, subsequently made an attempt at its level 3 equivalent) (Tim Bell et al., 2014). More positively, this research observes a 23% jump in numbers taking a key class required to major in CS or

Software Engineering in Canterbury University in 2014, the first year of university entry for the cohort of students to have had access to the new standards in secondary school. Within this number the number of females taking this class increased to 13% from an average of 7.5 – 7.9% in the previous three years.

Israel. The longest established upper second level CS programme within our report jurisdictions, also has the most balanced male female participation rates with an average ratio of 56% male to 44% female over a sixteen-year period (Figure 18) (Armoni & Gal-Ezer, 2014, P.110).

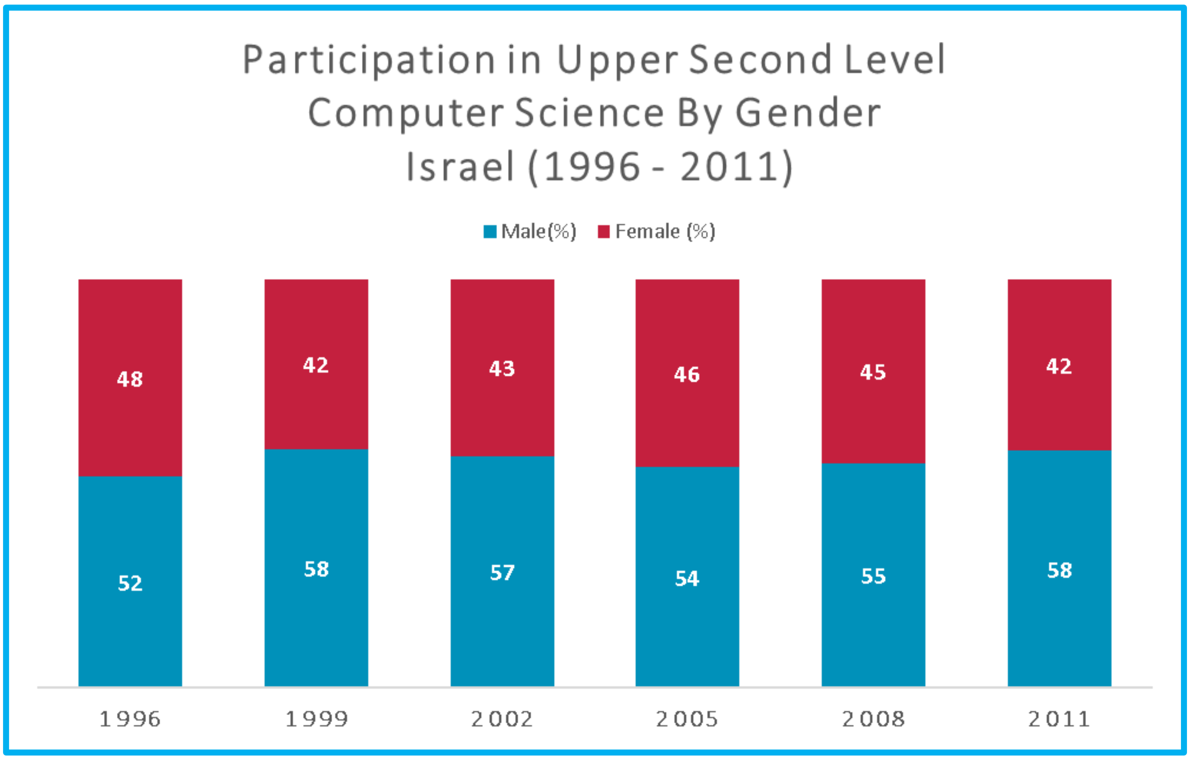

Figure 18: Participation Israeli Computer Science 'Bagrut' by Gender

An analysis of the Israeli upper second level CS programme by Armoni et al. in a 2014 research paper entitled "*High school computer science education paves the way for higher*

education: The Israeli case" found that female CS students achieved marginally better grades than their male counterparts over a seven-year period (2005 – 2011). This research continues to show that where male and female students that studied CS in high school they are disproportionately more likely to pursue third level studies in computing than those who had not (Figure 19)(Adapted from: Armoni & Gal-Ezer, 2014).

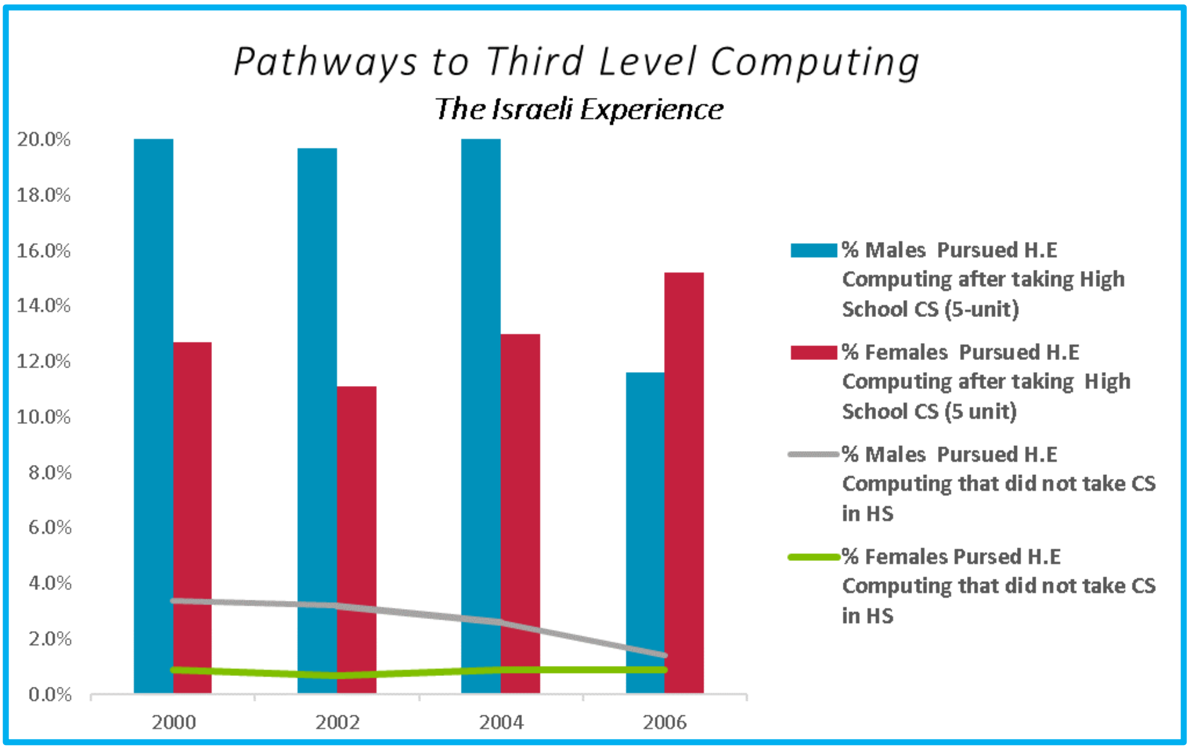

Figure 19: Israeli Student Pathways to Third Level Computing

As an extension of this analysis, 2013 data from the Israeli Central Bureau of Statistics show that females make up 38% of its ICT workforce (Israel Central Bureau of Statistics, 2013), against an average of 6 to 7% across Europe (Accenture, 2014). Whereas Israel has many attributes in its society that are distinctive and may account for several these differences, it is still interesting to note that it's robust participation rates of females in CS in high schools is extending into third level education and onwards into industry, and thus, helping to break the

self-perpetuating cycle of low participation of females in technology and computing mentioned earlier.

Summary of Findings. Participation by females in upper second level Computer Science is low across the reports jurisdictions (excluding Israel). This finding reflects the low levels of female participation in technology and computing related fields across industry, academia and society in general. This is an important issue, not only from an equality perspective but also from an educational, societal and economic viewpoint.

The research indicates that when female students do participate in CS programmes in upper second level education, they achieve on average a higher percentage of honour's grades than their male counterparts.

Pre-requisites

Are there pre-requisite studies needed – subjects that must be taken in lower second level?

Introduction. An analysis of the documentation in each of the report jurisdictions was conducted to determine whether pre-requisite studies were required prior to a student entering an upper second level Computer Science(CS) programme.

England. The examination boards in England do not specify strict pre-requisites before a student undertakes one of their upper second level CS programmes, but make recommendations to suitable prior learning. A final decision on entry to a programme is at the discretion of each individual school.

Eduqas suggests that it would be "*reasonable to assume that many learners will have achieved qualifications equivalent to Level 2 at KS4. Skills in Numeracy/Mathematics, Literacy/English and Information Communication Technology will provide a good basis for progression*" ,With some learners already having "*gained knowledge, understanding and skills through their study of Computer Science at GCSE or AS*".(2014 ,P.5). The AQA[21] specify that "there are no previous learning requirements" but "*recommend that students should have the skills and knowledge associated with a GCSE Computer Science course or*

[21] Assessment and Qualifications Alliance

equivalent" (2016, P.111). The OCR[22] state that learners need "no prior qualification" but will have "*likely to have followed a Key Stage 4 programme of study*"(2014, P.15).

Scotland. In Scotland the admission of students into the 'Higher' CS course "is at the discretion of the centre", but students would "*normally be expected to have attained the skills, knowledge and understanding required by the following or equivalent qualifications and/or experience: National 5 Computing Science Course or relevant Units*" (Scottish Qualifications Authority, 2015d, P.2). These expectations reflect changes introduced by the Curriculum for Excellence (CFE) in 2010. The CFE specified a new subject 'Computing Science', offered on the curriculum as a core entitlement and discrete subject commencing in the first three years of secondary school (leading to National 3, 4, & 5 Qualifications in Computing Science).

Students wishing to pursue an 'Advanced Higher' qualification would normally be expected to have completed the 'Higher' CS course, with the decision whether to admit a student (or not) left to the discretion of each school (Scottish Qualifications Authority, 2016a).

New Zealand. There are no pre-requisites subjects for students to complete before taking the first senior (level 1) standards in Programming and CS in New Zealand secondary schools. In fact, there are no strict pre-requisite structure in place between level 1 and level 2, with students able to take level two standards without completing level one first, a decision which is at the discretion of each individual school (Tim Bell et al., 2014). However, as each of the standards over the three years in upper second level build on the knowledge and learning

[22] Oxford, Cambridge and RSA

from the previous year, the most common progression through the Programming and CS standards is sequential.

Ontario. Entering senior grade 10, students wishing to study ‘Computing’ take the ‘Introduction to Computer Studies’ course which has no pre-requisite studies requirement. This entry level course is described in the curriculum as ‘Open’, “*designed to broaden students’ knowledge and skills in subjects that reflect their interests and to prepare them for active and rewarding participation in society. They are not designed with the specific requirements of universities, colleges, or the workplace in mind*”(Ministry of Education, Ontario, 2008, P.7).

On completion of this introductory course at the end of grade 10, students are then required to choose to pursue studies relevant to university pathway by taking the ‘Introduction to Computer Science’ course or choose a more vocational path by taking ‘Introduction to Computer Programming’. In their final year, grade 12, students continue their chosen path with a course in ‘Computer Science’ offered to those students that took ‘Introduction to Computer Science’ in grade 11, and similarly students that chose computer programming continue the Computer Programming stream (Figure 20) (Sturman & Sizmur, 2011, P.11).

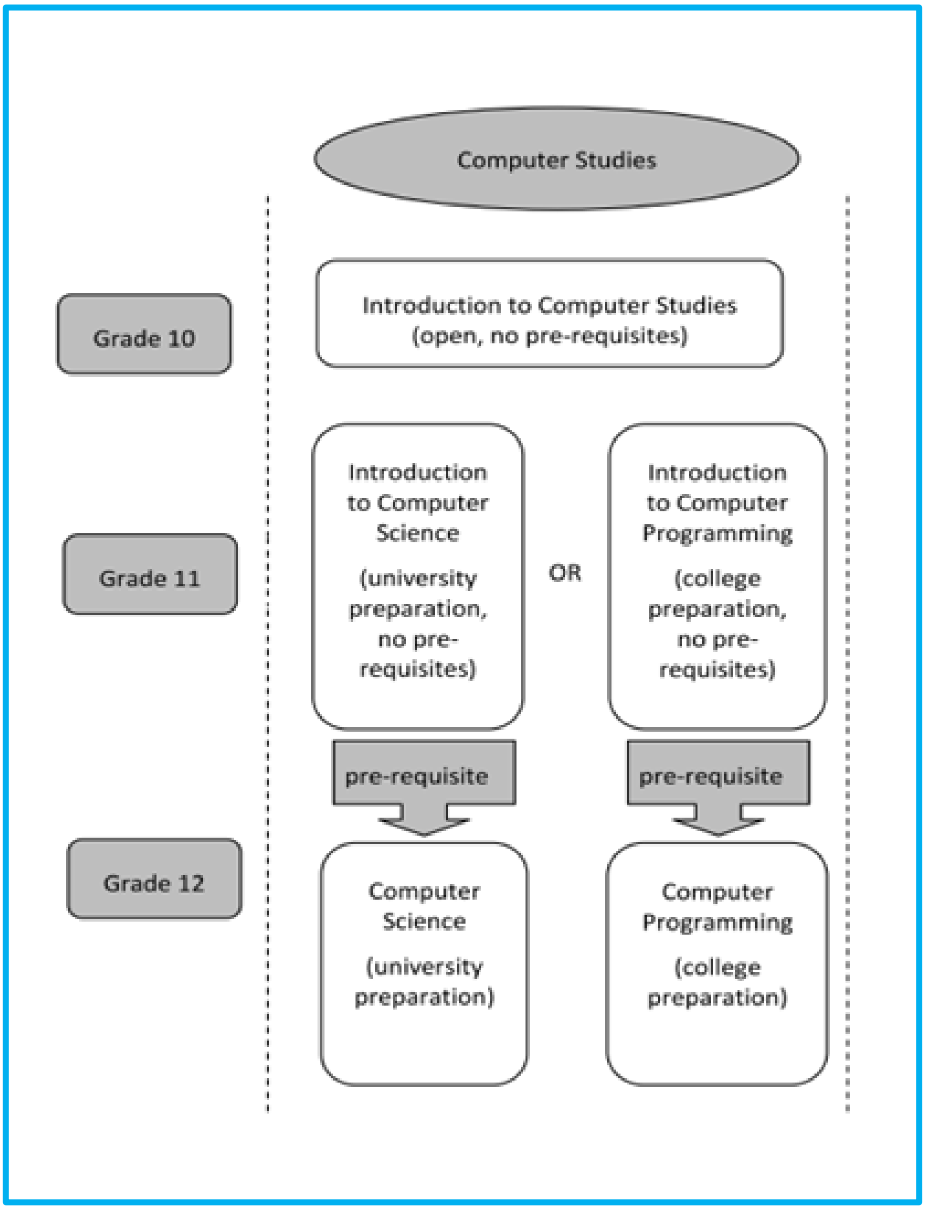

Figure 20: Pre-requisites in Ontario Computer Studies Programme

Israel. Israeli students entering upper second level, can choose to take CS as an elective subject without any requirement to have completed pre-requisite studies.

Students are given a choice of two programmes which they select depending on their level of interest in CS. The 3 unit programme is design to appeal to those students with a general

interest in the subject, and the 5 unit programme which is deeper and broader, is designed to appeal to those students with a greater interest in CS.

Prior to the introduction of the Science and Technology Excellence Program (STEP) into the lower second level in 2011, Israeli students outside of a number of short term and local educational initiatives, had few opportunities to learn CS before entering upper second level education (Zur Bargury et al., 2012).

Summary of Findings. Apart from Scotland, none of the report jurisdictions have strict requirements for pre-requisite studies prior to a student undertaking an upper second level CS programme. In England, the examination boards make recommendations to the extent of a student's prior studies, the remaining jurisdictions have no expectations of prior learning. In all jurisdictions, the decision to admit a student to a CS programme was at the ultimate discretion of school management.

Assessment

How is the subject assessed? Is it included in any formal end-of-school examination?

Introduction. In addressing this research question, each jurisdiction was analysed to (a) identify if a formal Computer Science(CS) qualification was offered which was recognised for purposes of entry into further and higher education, and (b) to determine what form the assessment of the subject took.

England. Upper second level students studying CS in English schools take part in assessments that lead to a formal national qualification award of an A-level. Assessment is conducted externally by three examination boards (AQA, OCR& Eduqas) which are audited by a government agency the Office of Qualifications and Examinations Regulation (Ofqual). Each student wishing to pursue higher education on completion of second level schooling takes several A levels, which universities and colleges recognise for admission purposes.

The Office of Qualifications and Examinations Regulation (Ofqual) assign assessment objectives (AOs) for each subject on the curriculum which each examination board must use to measure student achievement against. The three AOs for the CS are detailed in figure 21 (2016b).

Computer science

Objective	Requirements	Weighting
AO1	Demonstrate knowledge and understanding of the key concepts and principles of computer science	30%
AO2	Apply knowledge and understanding of key concepts and principles of computer science	40%
AO3	Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions	30%

Figure 21: Ofqual Assessment Objectives Computer Science

The three examination boards apply slightly different weightings to each of the AOs in their assessments, but they follow the same format of two examination papers (worth 40% each) and non-examination based assessment of a project where students exhibit practical problem solving skills (worth 20%). Each of the examination boards provide external assessment of the examination papers, and validation of the project component which is internally assessed by the teacher.

Weighting of assessment objectives for A-level Computer Science

Assessment objectives (AO)	Component weightings (approx %)			Overall weighting (approx %)
	Paper 1	Paper 2	NEA	
AO1	8	22	0	30
AO2	12	16	2	30
AO3	20	2	18	40
Overall weighting of components	40	40	20	100

Figure 22: AQA Assessment Objectives Weightings

For example, Figure 22 shows the AQA examinations board assessment matrix (2016, P.105). The AQA offers its first paper as an on-screen exam of 2 hr 30 min duration which contains a series of short questions and write/adapt/extend programs in an electronic answer document. The second paper is a written examination of 2 hr 30 min. The third component consists of non-exam assessment(NEA) of a project which allows students to exhibit independently learnt skills by providing a programming solution to a problem of their own choosing/ interest.

Scotland. The subject of CS is part of the national education qualification framework, with upper second level students learning assessed for the awarding of ‘Highers’ and ‘Advanced Highers’ qualifications. These qualifications meet the entry requirements of third level institutions, Scottish universities generally accepting ‘Highers ‘for entry, with English universities requiring Advanced Highers.

The Scottish Qualification Authority (SQA[23]) provides both external assessment and validation of internal assessment of the Highers and Advanced Highers qualifications depending on the specific requirements of the subject being assessed.

The ‘Highers ‘CS course assessment is made up of two components, an examination paper worth 60% of the available grade marks and an assignment worth the remaining 40%. The purpose of the question paper *“is to assess breadth of knowledge from across the Course, depth of understanding, and application of this knowledge and understanding to answer appropriately challenging questions*”(SQA, 2016c, P.5). The question paper is set by the SQA, administered in the schools under specified conditions and externally graded by SQA correctors.

[23] http://www.sqa.org.uk/

The purpose of the Highers assignment component is to assess the "*practical application of knowledge and skills from the Units to develop a solution to an appropriately challenging computing science problem. It will assess learners' skills in analysing a problem, designing, implementing and testing a solution to the problem, and reporting on that solution*"(SQA, 2016c, P.7). The assignment is set by the SQA, then conducted under open book conditions in each school under the supervision and control of the class teacher, and internally graded using SQA approved marking criteria. All grading is subsequently quality assured by the SQA to ensure the overall consistency of the assessment.

The first component of the Advanced Highers assessment is a student project worth 60% of the available grade marks in which "*Learners will apply knowledge and skills from across the Course to specify, plan, develop, implement, test and evaluate a digital solution to a significant and appropriately challenging computing-based problem*"(SQA, 2016a P.10). Each student project is set, monitored and internally marked by teachers in line with SQA marking criteria. All project grades are subject to quality assurance by the SQA.

The second Advanced Highers component is an examination paper worth 40% of the grade, with an objective to "*assess breadth of knowledge from across the Course, depth of understanding, and application of this knowledge and understanding to answer appropriately challenging questions*"(SQA, 2015a P.6) . The paper is set and marked by SQA, and is conducted in each school under conditions specified for external examinations.

New Zealand. The CS and Programming standards are part of The National Certificate in Educational Achievement (NCEA), which is the main national qualification for secondary school students in New Zealand. The NCEA is recognised by colleges and universities for admission requirements, both domestically and internationally.

Notably, prior to the introduction of the new CS & programming standards in 2012, ‘computing’ subjects were assessed as ‘unit’ standards (competency based), which are awarded on a pass or fail basis and do not count for university entrance. This led to the more academically minded students avoiding the subject, generating a perception that CS was a non- academic subject (T. Bell et al., 2014).

Each year upper second level students (years 11 to 13) choose several subjects to study, on which they are assessed against defined standards. The NCEA qualification itself is flexible, students are free to select various combinations of small modules (“standards”), contingent on the school’s provisioning. On achievement of a standard a student gains credits, with aggregation of a set number of credits leading to the award of a NCEA certificate. Certificates are awarded on three levels; achievement, merit and excellence, which mirrors the achievement levels in each standard.

Assessment of student work for the NCEA is managed and moderated by the New Zealand Quality Authority (NZQA[24]), provisioning for both external assessment and validation of internal assessment depending on the most appropriate assessment needs for each standard. Each of the five Programming standards require students to plan and/or develop computer programs of varying complexity which are assessed internally by the teacher, with external auditing (moderation by NZQA) to ensure consistency. This assessment approach reflects the most suitable pedagogical approach for the subject (active experiential learning), with each teacher then able to observe the processes that a student uses to program, including how independently a student is working.

The three CS standards are externally assessed by NZQA appointed markers. These external assessments are submitted as written reports (14-page maximum) which allow students to

[24] http://www.nzqa.govt.nz/about-us/our-role/what-we-do/

detail and reflect on their personal learning and experience ("the student voice") with the topic rather than sitting a written exam (T. Bell et al., 2014; Hubwieser, Armoni, & Giannakos, 2015). Pragmatically this assessment approach to the three CS standards helps resolve internal resource constraints and timetable issues, but also places a high demand on teachers given the bespoke nature of the assessment.

Ontario. In Canada, the education system is run separately by each of its thirteen provinces and territories. This federated approach results in a non-integrated national system of education with no national qualifications, examinations, or awarding bodies. Examinations are set, conducted and marked provincially or by individual schools in almost all subjects at the end of high school

In Ontario, a student's grades are recorded in the 'Provincial Report Card' for each subject in secondary school through levels 9 – 12. As a student finishes each course of instruction a final grade is recorded, with a credit granted in subjects in which a student's grade is 50% or higher (Ministry of Education, Ontario, 2008). On receipt of sufficient credits (30), students are awarded a high school diploma, which counts towards college or university admission in Ontario. The final grade is devised as follows:(Ministry of Education, Ontario, 2008)

- **Seventy per cent** of the grade will be based on evaluations conducted throughout the course.
- **Thirty per cent** of the grade will be based on a final evaluation in the form of a programming project and/or a written examination, the choice of which is decided locally.

Israel. In Israel, upper second level CS students take an examination as part of the national matriculation exam called the ‘Bagrut’. This external examination is essential for admission to most Israeli universities. A final school grade is calculated as the average of a student’s matriculation exam and an internal grade which is based on internal examinations and their performance throughout the year (Averbuch, Benaya, & Zur, 2011).

Students entering the Israeli upper second level can take three different programmes of study in CS (Gal-Ezer & Harel, 1999):

- An introductory One-unit program consisting of 90 hours of tuition.
- An intermediate Three-unit program consisting of 270 hours of tuition, applicable for the Bagrut examination.
- A full five-unit program consisting of 450 hours of tuition, applicable for the Bagrut examination.

Both three unit and five unit programmes are examined by a combination of written examination, and course work. This includes students preparing a project examined in the laboratory(Armoni & Gal-Ezer, 2014), see Figure 23 (Gal-Ezer & Harel, 1999, P.3).

Study units	written exam weight	lab exam weight	total weight in 3-unit program	total weight in 5-unit program
Fundamentals 1	75%	25%	30%	
Fundamentals 2	50%			50%
Second Paradigm or Applications	25%	25%	70%	
Software Design	40%	20%		50%
Theory	40%			

Figure 23: Assessment of Computer Science Programmes Israel

Summary of Findings. In each of the jurisdictions, the study of Computer Science at upper second level resulted in the awarding of national qualifications which were recognised for the purposes of entry into further and higher education.

The assessment strategies detailed combine the provision of traditional written examinations and course work such as project work that students complete independently during their school year. While the weighting assigned to each of these two components differed by jurisdiction, a key finding was the universal recognition of the importance of providing assessment that enabled each student to exhibit the practical application of skills learnt, alongside their understanding of the subject content.

	Qualification	Examination Paper (Formal)	Assessed	Course Work (Project / Lab)	Assessed
England	A - Level	80%	**External**	20%	**Internal**
Scotland	Highers	60%	**External**	40%	**Internal**
	Advanced Highers	40%	**External**	60%	**Internal**
New Zealand	NCEA: Programming	0%	**N/A**	100%	**Internal**
	NCEA: Computer Science	0%	**N/A**	100%	**External**
Ontario	Provincial Report Card	30%	**Internal**	70%	**Internal**
Israel	Bagrut	75-80%	**External**	25- 20%	**Internal**

Table 11: Summary of Assessment Strategies Across Report Jurisdictions

Section Four

Teacher Professional Development

Teacher Professional Development

> What specific supports are in place for teachers of Computer Science courses, both in continuing professional development and initial teacher training?

Introduction. The importance of teachers in the adoption, implementation and sustainability of a Computer Science (CS) curriculum is universally recognised in the CS education community (Hazzan et al., 2008; Sue Sentance, Dorling, McNicol, & Crick, 2012; Stephenson, Gal-Ezer, Haberman, & Verno, 2005; Thompson & Bell, 2013; Webb et al., 2016).

Given this noted importance, a common concern across the jurisdictions is the difficulty of ensuring that teachers have the pedagogical content knowledge (PCK) required to teach CS, and that this knowledge is refreshed and updated as needed. Therefore, the preparation (pre-service) and continuing professional development (in-service) of CS teachers is mission critical.

In the following sections, the provision of both initial training and continuing professional development of teachers in each of our research jurisdictions is reviewed. The importance placed in each jurisdiction on fostering communities of practice, is highlighted as a key component in developing a professional identify and in turn enhancing CS teachers self-perception and confidence (Ni & Guzdial, 2012).

England. The English school system had thousands of ICT teachers as the subject was disapplied in September 2012 to introduce a new Computing curriculum consisting of Computer Science, IT and Digital Literacy. This presented the Ministry of Education with a ‘grand’ challenge in how best to support these teachers in their professional development, to support them in acquiring new subject skills, curriculum and content knowledge and subject specific pedagogical approaches.

Given the short space of time since this new curriculum was introduced, the answers to many of the challenges are still yet to be fully defined. However, research by Dr. Sue Sentance et al. provide valuable insights into successes in the English system (Crick & Sentance, 2011; Sue. Sentance, 2016; Sue Sentance & Csizmadia, 2015; Sue Sentance, Dorling, & McNicol, 2013; Sue Sentance et al., 2012; Sue Sentance, Humphreys, & Dorling, 2014).

Figure 24: Model of Teacher Professional Development

In her recent article entitled "*A holistic approach to teacher professional development in Computing*"(2016), Sentance presents a model of Teacher Professional Development which combines Kennedy's (2005, 2014) transformative model of integrating multiple approaches with the notion of communities of practice (Figure 24). Dr. Sentance instantiates this model by detailing how Computing At School(CAS) supports each aspects of the model, to provide a comprehensive eco-system that offers different supports at different stages of a teachers development (2016) (Figure 25).

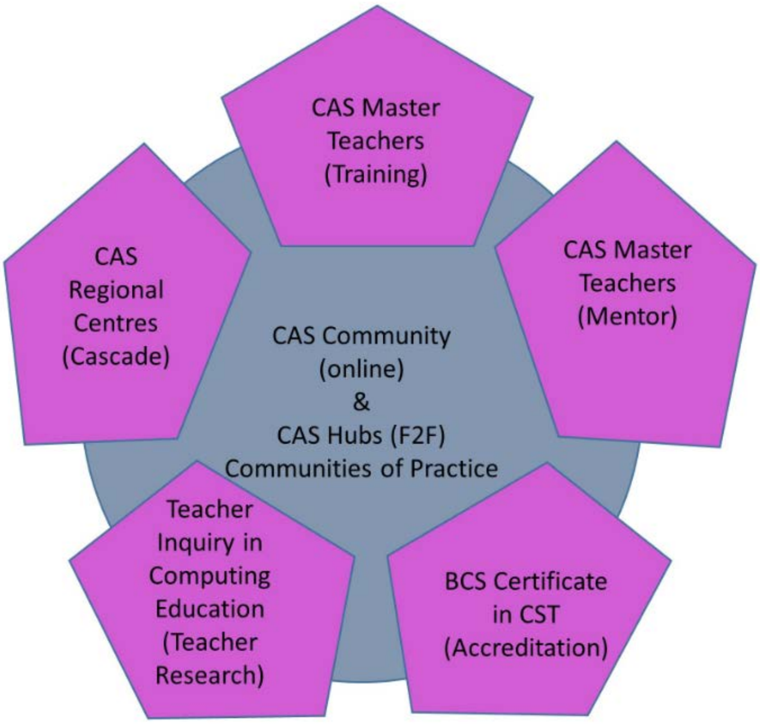

Figure 25: Instantiated Model of Teacher Professional Development

Individuals wishing to teach CS in an English secondary school can (a) enrol in one of the available undergraduate degrees in education (CS), (b) enrol in a postgraduate course in

secondary education once the individual has an under graduate degree in a related discipline, or (c) in limited cases through a process of in-school training (Brown et al., 2014).

Like other jurisdictions, there is the difficulty recruiting suitable candidates to undergo teacher training in the subject area as graduates of CS are in high labour market demand. An interesting initiative to address this supply side problem is availability of a number of generous bursaries to attract highly qualified candidates to the profession (Sue Sentance et al., 2013).

Scotland. Individuals wishing to become a CS Teacher in Scottish Secondary schools have the option of undertaking a four-year Bachelors of Education specialising in CS, or a Post Graduate Diploma in Education (PGDE) after completing a primary degree in a related subject area (such as Computer systems, software development, databases or information systems) (University of Strathclyde, ND).

A 2015 parliamentary briefing by Computing at School Scotland (CASS) , detail a number of current difficulties in recruiting new CS applicants, with only 21 students training in 2014 out of a target of 25 (maximum of 41 places)(CASS, 2015). CASS reason that a combination of supply side (competitive job market for CS graduates, absence of flexible distance and part time PGDE's), and demand side factors (lowest replenishment rate of all STEM subjects, reduction in number of schools offering the subject) are behind these low figures.

The Professional Learning and Networking for Computing project (Plan C), launched in 2014, is a CASS initiative funded by the Scottish Government to support and strengthen the teaching of CS in Scottish schools. A network of 50 'lead' teachers from a wide geographical spread of Scottish secondary schools were trained in phase one, with each lead teacher then

supported to establish 25 local teacher hubs all over Scotland in the next phases (Computing At School, 2016 ; The Chartered Institute for IT (BCS), nd).

PLAN C

Professional Learning And Networking for Computing

Figure 26: Computing at School Scotland Plan C

These local hubs provide the fulcrum for the provision of continuous professional development, with many training programmes conducted by the ‘lead’ teacher with an emphasis on peer to peer learning. The hubs provide an effective mechanism for the sharing of opinions, knowledge and resources. In this manner the human infrastructure was developed for a vibrant and knowledgeable community of practice, with at least 350 of the 650 CS secondary teachers involved in the programme(CAS, 2016 ; CASS, 2015).

New Zealand. The new Digital Technology standards in New Zealand (NZ) were introduced very rapidly, with little over two-years between the 2009 Digital Technologies Expert Panel (DTEP) report calling for their urgent introduction and the subjects being taught in high school classrooms. This accelerated approach meant that existing teachers had little time to prepare for the changes, with these preparations further hampered by the limited availability of formal training and other resources (T. Bell et al., 2014). A 3-day national symposium,

which was held three months before the introduction of the new standards in February 2011, largely represented the formal training provided prior to the changes (Thompson et al., 2013). Given the short time frame, an opportunity to train new teachers in the standards was not available.

New entrants wishing to teach a subject such as Digital Technologies in NZ are required to "have completed a subject-based degree or an NZQF Level 7 diploma with a mix of subjects relevant to their chosen teaching subjects, followed by a Graduate Diploma of Teaching" (Ministry of Education, New Zealand, 2016b). In contrast, many of the pre-existing ICT teachers started as typing or commerce teachers. A survey of digital technology teachers in the first year of the new standards, found that 60% of the 91 respondents were aged 50 or older and that only 56% had any specific degree in a computing related area, with 11% having a CS degree and a further 11% with a IS or IT degree (Thompson et al., 2013).

To further add to the challenge, the city of Christchurch experienced a series of major earthquakes which disrupted society for several months, closing high schools and other public buildings for several weeks during the first months of the introduction of the new standards.

Despite all these challenges, the majority of existing computing teachers "*embraced the change, primarily because of wanting to do the right thing for the students and the country, rather than due to any directives from management*" (Tim Bell, 2014, P.29; Thompson et al., 2013).

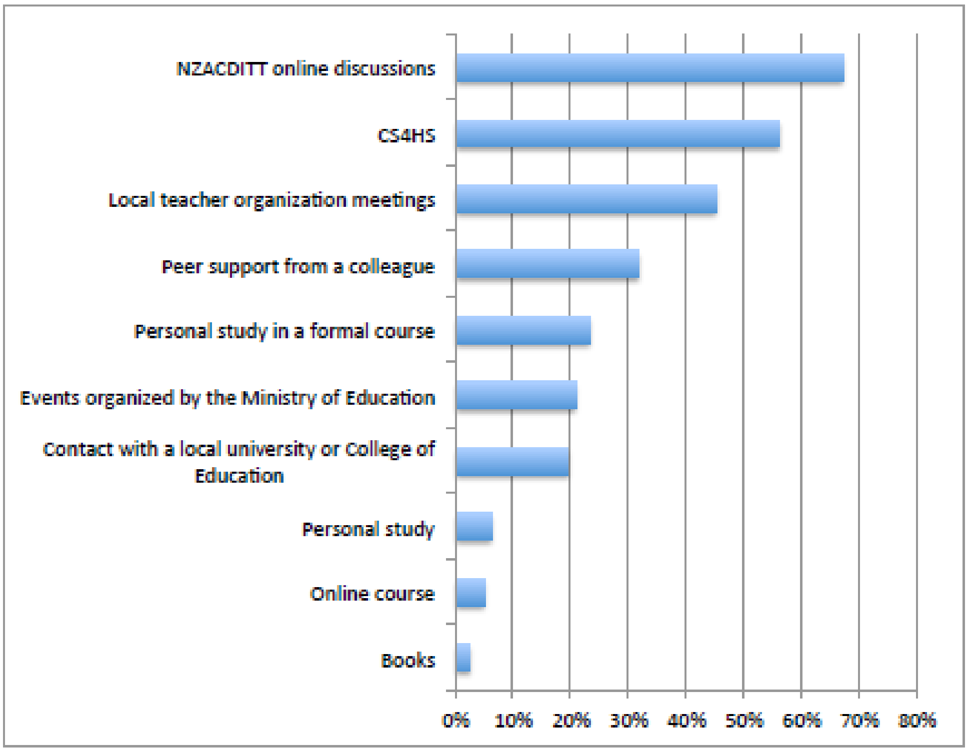

Figure 27: Sources of Teacher Professional Development New Zealand

Since the introduction of the new CS and Programming standards in 2012, considerable effort has been made to help the existing cohort of teachers to develop their competence and very importantly their confidence with these new standards. Figure 27 shows the usage by teachers of the various supports provided, from a survey conducted by David Thompson and Tim Bell (2013, P.3).

The findings exhibited in Figure 27 highlight the central importance placed by teachers on peer to peer support, learning and sharing. At the centre of this community of learning is The New Zealand Association for Computing, Digital and Information Technology Teachers (NZACDITT[25]). Founded in March 2009, the NZACDITT has been greatly supported by universities, industry and government working in partnership to contribute to initiatives such

[25] http://nzacditt.org.nz/

as the CS4HS[26] workshops, CS teacher qualifications delivered by distance learning, online open source resources (e.g. "Computer Science Field Guide[27]"), and CS teacher helplines (Tim Bell, 2014; Thompson et al., 2013).

Ontario. There are currently no specific Computer Studies(CS) teacher training programmes either provincially or nationally in Canada (Snyder, 2012). New entrants who wish to teach CS in Ontario high schools are required to have an Ontario teaching licence and a third level qualification in computer science, software engineering or equivalent professional experience. The Ontario College of Teacher's (OCT) issues licences to individuals who have completed teaching training in one of its fifty accredited full and part time teacher education programs in eighteen universities across Ontario.

OCT approves and reviews 'additional qualifications' for teachers whose objectives are to "develop the skills and knowledge needed by teachers to design, deliver and assess programs "in specific disciplines including CS (OCT, nd). Teachers on completion of these qualifications can take a 'Specialist course' that extends and reinforces their earlier learning, but with a specific focus on the development of leadership skills within a discipline (OCT, 2010).

Ontario has a professional association called 'The Association for Computer Studies Education'[28] (ACSE), which is run by educators from high schools, colleges and universities. The mandate of the ACSE is to support CS students and teachers in creating a community of practice through its provision of conferences, summer institutes, on-line resources, and by

[26] http://www.cs4hs.com/
[27] http://csfieldguide.org.nz/
[28] http://www.acse.net/home

providing a unified voice for the community with the ministry and the Ontario College of Teachers.

Israel. With the introduction of the revised Israeli curriculum in 1998, an undergraduate degree in CS coupled with a formal teacher training certification is a mandatory requirement for individuals wishing to teach CS in Israeli high schools. This stipulation of an appropriate CS teaching qualification has been acknowledged as an essential component to the success of the CS programme in Israel (Hazzan et al., 2008).

As CS had been offered in Israeli high schools since the 1970's, the school system at the time of the curriculum revision had a pre-existing cohort of CS teachers with varying levels of subject, curricular, and pedagogical knowledge (Haberman & Ginat, 1999). To address this, a large scale in-service teacher training programme was conducted in the first six months of 1998, which combined distance learning with local workshops led by experienced leading teachers who already piloted the new material (Haberman & Ginat, 1999).

For new entrants, there are two routes to becoming a CS teacher in Israel: (a) training programmes available to CS graduates without formal teacher training and to existing teachers of other scientific topics who wish to teach CS, and (b) teacher preparation programmes provided by third level institutions, involving four years of study that combine a Bachelor's degree in CS and a teacher preparation programme specific to the subject. The teacher preparation programs concentrate on three essential domains: subject matter knowledge, pedagogical content knowledge, and curricular knowledge (Armoni, 2011; Gal-Ezer & Zur, 2007). These teacher preparation programmes also function to provide the infrastructure and opportunity to offer in-service teachers ongoing training.

A notable aspect of the preparation of pre-service teachers is the inclusion of research elements into each teacher training programme, such as the undertaking of research based projects and assignments based on the synthesis of peer reviewed research papers. The intention is to familiarise each new CS teacher with the body of research in CS education, adding to the active research community contributing to the ongoing development process of CS curricula and syllabi (Hazzan, Gal-Ezer, & Ragonis, 2010).

The Israel National Center for Computer Science Teachers (‘Machshava[29]’) founded in 2000 by the Ministry of Education is the professional hub for all CS teachers in Israel. This dedicated center is responsible for the organisation and funding of in-service study programs, workshops, and guidelines to assist the teachers in their professional development. As part of this provision, Machshava provides summer seminars called “On the Frontier of Computer Science” for ‘leading teachers’ to foster their professional leadership in the CS education community. This leading teacher programme recognised the value of experienced and enthusiastic teachers as coaches in peer to peer within the community (Lapidot & Aharoni, 2008).

Machshava’s remit is not only to provide a structure for in-service CS teacher training but also critically to advance the professional identity of the profession by building a community of practice through activities such as the hosting of an annual teacher conference, arranging courses, facilitating meetings on specific issues, supporting local teacher centres, and the publication of research and learning materials (Gal-Ezer & Stephenson, 2014; Machshava, NA)

[29] http://cse.proj.ac.il

Summary of Findings. Each of the report jurisdictions recognises the central role of teachers in the adoption, implementation and sustainability of a Computer Science curriculum. As most of the jurisdictions had a large cohort of ICT teachers from legacy programmes an emphasis was placed on the professional development of these existing teachers in a manner that was cost and time effective. The resultant strategy of the fostering of communities of practice, integrated with multiple approaches professional development (such as training, mentoring, research, accreditation and peer to peer knowledge sharing) was found to be both a welcome and effective approach to these challenges.

Section Five

Implications & Recommendations

Implications and Recommendations

Introduction

In conducting our report on the provision of courses in Computer Science in upper second level education internationally, several key and interrelated learnings emerged that have important implications for the NCCA’s work in developing advice for the introduction of a Computer Science course for the Leaving Certificate. The key learnings are presented under three categories: Subject, Curriculum and Infrastructure.

Subject

Content. Across our report jurisdictions, the central aspects of the content in each Computer Science(CS) course were broadly similar. This finding reflects a strong consensus on content internationally, suggesting that those charged with devising the content for a new Leaving Certificate Computer Science subject will have an extensive and largely agreed body of content to call upon.

Assessment. Each of the assessment strategies within our report jurisdictions combine the sitting of traditional written examination papers and course work such as software programming projects that students complete independently during their school year. While the weighting assigned to each of these two components differ by jurisdiction, a key takeaway was the universal recognition of the importance of providing assessment that enabled each student to exhibit the practical application of skills learnt, alongside their understanding of the subject content.

Pedagogy. Computer Science is a subject that requires a high level of application and experimentation to allow students construct their own personal meaning of its core concepts and constructs. Ideally a pedagogical approach to the subject must not only allow students sufficient time for this important application and experimentation during school hours, but also to encourage and nurture self-directed learning at home

Curriculum

Supporting Student Learning. The experience of our report jurisdiction in provisioning upper second level Computer Science suggest the importance of supporting students learning by providing instruction in the subject earlier in their schooling.

Changes are taking place in each of the reports jurisdictions to provide ‘scaffolding’ of students CS learning in lower secondary (e.g. GCSE Computer Science in England). As Computer Science is introduced onto the Irish curriculum, an opportunity exists to act on this international experience by provisioning student learning in Computer Science prior to their entry into upper second level. The current pilot of the short course in coding in selected schools provides one such template for this important student support.

Computational Thinking. While Computational Thinking is a skill that underpins Computer Science programmes, it is increasingly important across all STEM disciplines whose boundaries are being expanded thought the use of powerful, inexpensive and available computing technologies.

As Computer Science is introduced onto the Irish curriculum, the extent to which Computational Thinking and other aspects of the subject may influence and inform other complementary subjects needs to be considered.

Infrastructure

Subject Sustainability. A low rate of student participation in Computer Science programmes was evident across our reports jurisdictions, with particularly low rates amongst girls (aside from Israel on both counts). The more recently implemented CS programmes (England, Scotland, and New Zealand) have positive yearly growth trends (albeit from low numbers), but it is too early to draw any conclusions.

As the Irish experience, may not necessarily be all that different from these more recent implementations, it is important to consider approaches to a CS programme implementation that can best foster a sustainable cohort of students.

Teacher Professional Development. Teachers are recognised as of pivotal importance to the adoption, implementation and sustainability of a Computer Science curriculum. A key challenge across each of the reports jurisdictions is to ensure that teachers are supported in their professional development to acquire the subject, curricular and pedagogical knowledge and skills needed to protect the student experience and rigour of the subject.

From an Irish perspective, the more recent implementations (England, Scotland & New Zealand) offer many valuable lessons on the provision of pre-and in service professional development of teachers, warranting the undertaking of further detailed research.

References

Accenture. (2014). Powering economic growth;Attracting more young women into science and technology. In Women Invent Tomorrow at Silicon Republic (Ed.). Accenture Dublin.

AQA. (2016). AS AND A-LEVEL COMPUTER SCIENCE (V1.2 ed.). AQA, Devas Street, Manchester M15 6EX.: aqa.org.uk.

Armoni, M. (2011). Looking at secondary teacher preparation through the lens of computer science. *ACM Transactions on Computing Education (TOCE), 11*(4), 23.

Armoni, M. (2016). Computing in Schools: High-school software engineering education: personal impressions of an 18-year-old student. *ACM Inroads, 7*(2), 25-27.

Armoni, M., & Gal-Ezer, J. (2014). High school computer science education paves the way for higher education: the Israeli case. *Computer Science Education, 24*(2-3), 101-122.

Averbuch, H., Benaya, T., & Zur, E. (2011). *Criteria for writing exams which reflect the k12 CS foundations study material.* Paper presented at the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.

Bell, T. (2014). Establishing a nationwide CS curriculum in New Zealand high schools. *Communications of the ACM, 57*(2), 28-30.

Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602485

Bell, T., Newton, H., Andreae, P., & Robins, A. (2012). *The introduction of computer science to NZ high schools: an analysis of student work.* Paper presented at the Proceedings of the 7th Workshop in Primary and Secondary Computing Education.

Bell, T., Newton, H., Duncan, C., & Jarman, S. (2014). *Adoption of Computer Science in NZ schools.* Paper presented at the Proceedings of ITx-New Zealand's Conference of IT.

Brown, N. C. C., Kölling, M., Crick, T., Jones, S. P., Humphreys, S., & Sentance, S. (2013). *Bringing computer science back into schools: Lessons from the UK.* Paper presented at the SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education.

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602484

Carrell, T., Gough-Jones, V., & Fahy, K. (2008). The future of Computer Science and Digital Technologies in New Zealand secondary schools: Issues of 21st teaching and learning, senior courses and suitable assessments: Technical report.

Cheema, J. R., & Zhang, B. (2013). Quantity and quality of computer use and academic achievement: Evidence from a large-scale international test program. *International Journal of Education and Development using Information and Communication Technology, 9*(2), 95.

CoderDojo. (2016). CoderDojo Movement (Vol. 2016). https://coderdojo.com/movement/: CoderDojo.

College of Teachers, O. (2010). Additional Qualification Course Guideline Computer Studies, Specialist (pp. 18). Ontario: Standards of Practice and Accreditation Department.

College of Teachers, O. (nd). Schedule D: Three-session Additonal Qualifactions (Vol. 2016, pp. 18). Ontario: Standards of Practice and Accreditation Department.

Computing At School. (2016). Switched On *Autumn 2016* (pp. 24). London: CAS.

Computing at School Scotland. (2015). Briefing on Computing Science Education in Scotland. In Kate Farrell & Professor Quintin Cutts (Eds.).

Computing At Schools Scotland. (2014). SwitchedOn. Edinburgh: CAS Scotland.

Crick, T., & Sentance, S. (2011). *Computing at school: stimulating computing education in the UK.* Paper presented at the Proceedings of the 11th Koli Calling International Conference on Computing Education Research.

Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM, 52*(6), 28-30.

Department of Education. (2013). Statutory guidance : National curriculum in England: computing programs of study. London: The Stationery Office.

Department of Education. (2014). GCE AS and A Level subject content for computer science. London: The Stationary Office.

Department of Education and Skills. (2015). Digital Strategy for Schools 2015-2020: Enhancing Teaching, Learning and Assessment. In D. o. E. a. Skills (Ed.). Dublin The Stationery Office

Department of Education and Skills, & Department of Jobs, Enterprise and Innovation,. (2014). ICT Skills Action Plan (pp. 24). Dublin: The Stationery Office

Duncan, C., Bell, T., & Tanimoto, S. (2014). *Should your 8-year-old learn coding?* Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education.

Education Scotland. (NA). Curriculum for Excellence: technologies *experiences and outcomes* (Vol. 2016). Edinburgh: Scottish Government Publications.

Eduqas. (2014). WJEC Eduqas GCE A Level in Computer Science Specifaction (pp. 42).

Fellows, M., & Parberry, I. (1993). SIGACT trying to get children excited about CS. *Computing Research News, 7*.

Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. Retrieved from London:

Gal-Ezer, J., Beeri, C., Harel, D., & Yehudai, A. (1995). A High School Program in Computer Science. *Computer, 28*(10), 73-80. doi:10.1109/2.467599

Gal-Ezer, J., & Harel, D. (1999). Curriculum and Course Syllabi for a High-School Program in Computer Science. *Computer Science Education, 9*(2), 114-147.

Gal-Ezer, J., & Stephenson, C. (2014). A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602483

Gal-Ezer, J., & Zur, E. (2007). Reaching out to CS teachers: Certification via distance learning. *Mathematics and Computer Education, 41*(3), 250.

Grimsey, G., & Phillipps, M. (2008). Evaluation of technology achievement standards for use in New Zealand secondary school computing education. *Technical Rep. New Zealand Computer Society, Wellington.* http://www. *nzcs. org. nz/news/uploads/PDFs/200805NCEAReport. pdf.*
Haberman, B., & Cohen, A. (2007). A high-school programme in software engineering. *International Journal of Engineering Education, 23*(1), 15-23.
Haberman, B., & Ginat, D. (1999). *Distance learning model with local workshop sessions applied to in-service teacher training.* Paper presented at the iTiCSE 99, Cracow, Poland.
Hagan, D., & Markham, S. (2000). *Does it help to have some programming experience before beginning a computing degree program?* Paper presented at the ITiCSE 2000, Helsinki, Finland.
Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). *A model for high school computer science education: the four key elements that make it!* Paper presented at the Conference'08, Portland, Oregon, USA.
Hazzan, O., Gal-Ezer, J., & Ragonis, N. (2010). How to Establish a Computer Science Teacher Preparation Program at Your University? The ECSTPP Workshop. *ACM Inroads, 1*(1), 35-39.
Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to implement rigorous computer science education in K-12 schools? Some answers and many questions. *ACM Transactions on Computinig Education, 15*(2). doi:10.1145/2729983
Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., . . . Jasute, E. (2015). *A global snapshot of computer science education in K-12 schools.* Paper presented at the ITiCSE-WGP 2015 - Proceedings of the 2015 ITiCSE Conference on Working Group Reports.
ICT Ireland. (2016). Making Ireland a Global Technology Powerhouse (Vol. 2016). Dublin ICT Ireland
Informatics Europe, & ACM Europe. (2013). Informatics education: Europe cannot afford to miss the boat *Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education*.
Israel Central Bureau of Statistics. (2013). *Labour Force Survey 2013*. IBS Website (http://www.cbs.gov.il/) Retrieved from http://www.cbs.gov.il/webpub/pub/text_page_eng.html?publ=68&CYear=2013&CMonth=1#2.
Joint Council for Qualifications. (2011). A, As and AEA Results, Summer 2011. London: JCQ.
Joint Council for Qualifications. (2012). A, As and AEA Results, Summer 2012. London: JCQ.
Joint Council for Qualifications. (2013). A, As and AEA Results, Summer 2013. London: JCQ.
Joint Council for Qualifications. (2014a). A, As and AEA Results, Summer 2014. London: JCQ.
Joint Council for Qualifications. (2014b). GCSE and Entry Level Certificate Results Summer 2014. On-LIne: JCQ.

Joint Council for Qualifications. (2015a). A, As and AEA Results, Summer 2015. London: JCQ.

Joint Council for Qualifications. (2015b). GCSE and Entry Level Certificate Results Summer 2015. On-LIne: JCQ.

Joint Council for Qualifications. (2016a). A, As and AEA Results, Summer 2016. London: JCQ.

Joint Council for Qualifications. (2016b). GCSE and Entry Level Certificate Results Summer 2016. On-LIne: JCQ.

Jones, S. P. (2010). Computing at School: the state of the nation A report of the Computing at School Working Group. London: UK Computing Research Committee.

Jones, S. P. (2011). Computing at School *International comparisons* (Vol. 7, pp. 2013). London: Computing at School

Kemp, P. E., Wong, B., & Berry, M. G. (2016). The Roehampton Annual Computing Education Report.

Kennedy, A. (2005). Models of continuing professional development: a framework for analysis. *Journal of in-service education, 31*(2), 235-250.

Kennedy, A. (2014). Understanding continuing professional development: the need for theory to impact on policy and practice. *Professional development in education, 40*(5), 688-697.

Lapidot, T., & Aharoni, D. (2008). On the frontier of computer science: Israeli summer seminars. *ACM SIGCSE Bulletin, 40*(4), 72-74.

Machshava. (NA). Machshava – Israel National Center for Computer Science Teachers. Retrieved from http://cse.proj.ac.il/index-en.htm

Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education, 16*(3), 211-227.

Ministry of Education, New Zealand. (2009). Technolical Context and Skills Wellington: Learning Media/Crown.

Ministry of Education, New Zealand. (2012). Digital Technologies: Learning objectives, indicators of progression, and teacher guidance. Wellington: Learning Media/Crown.

Ministry of Education, New Zealand. (2016a). Digital technology to become part of the New Zealand Curriculum and Te Marautanga o Aotearoa. Wellington: Learning Media/Crown.

Ministry of Education, New Zealand. (2016b). Secondary Teacher Education Qualifactions 2017 (pp. 27). Wellington: Learning Media/Crown.

Ministry of Education, New Zealand. (2016c). Senior Cycle Technology Acheivement & Learning Objectives.

Ministry of Education, Ontario. (2008). The Ontario Curriculum, Grades 10 to 12: Computer Studies (revised). Toronto: Queen's Printer for Ontario.

Mitchell, A. (Producer). (2004). Computing Science: what do pupils think? Retrieved from http://www.powershow.com/view/4faa3-NTQyM/Computing_Science_What_do_pupils_think_powerpoint_ppt_presentation

Mitchell, A., Purchase, H. C., & Hamer, J. (2009). Computing science: what do pupils think? *SIGCSE Bull., 41*(3), 353-353. doi:10.1145/1595496.1562995

National Council for Curriculum and Assessment (NCCA). (2004a). Curriculum Assessment and ICT in the Irish Context: A discussion paper. Dublin: The Stationary Office,.

National Council for Curriculum and Assessment (NCCA). (2004b). Information and Communications Technology (ICT) in the Primary School Curriculum *Guidelines for teachers*. Dublin: The Stationary Office.

National Council for Curriculum and Assessment (NCCA). (2005). Proposals for the future development of senior cycle education in Ireland. Dublin: The Stationary Office

New Zealand Qualifications Authority. (2011). Achievement Standard: 2.44 Demonstrate understanding of advanced concepts from computer. Wellington: NZQA.

New Zealand Qualifications Authority. (2012). NCEA Qualification Endorsement Statistics Files *National*. On-Line: NZQA.

New Zealand Qualifications Authority. (2013). NCEA Qualification Endorsement Statistics Files *National*. On-Line: NZQA.

New Zealand Qualifications Authority. (2014a). Achievement Standard: 3.44 Demonstrate understanding of areas of computer science. Wellington: NZQA.

New Zealand Qualifications Authority. (2014b). Achievement Standard: 3.46 Develop a complex computer program for a specified task. Wellington: NZQA.

New Zealand Qualifications Authority. (2014c). NCEA Qualification Endorsement Statistics Files *National*. On-Line: NZQA.

New Zealand Qualifications Authority. (2015a). Achievement Standard: 1.44 Demonstrate understanding of basic concepts from computer science. Wellington: NZQA.

New Zealand Qualifications Authority. (2015b). Achievement Standard: 1.45 Construct a plan for a basic computer program for a specified task. Wellington: NZQA.

New Zealand Qualifications Authority. (2015c). Achievement Standard: 1.46 Construct a basic computer program for a specified task. Wellington: NZQA.

New Zealand Qualifications Authority. (2015d). Achievement Standard: 2.45 Construct a plan for an advanced computer program for a specified. Wellington: NZQA.

New Zealand Qualifications Authority. (2015e). Achievement Standard: 2.46 Construct an advanced computer program for a specified task. Wellington: NZQA.

New Zealand Qualifications Authority. (2015f). NCEA Qualification Endorsement Statistics Files *National*. On-Line: NZQA.

Ni, L., & Guzdial, M. (2012). *Who AM I?: understanding high school computer science teachers' professional identity*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.

O'Doherty, T., Gleeson, J., Johnston, J., McGarr, O., & Moody, J. (2004). Computers and curriculum: difficulties and dichotomies. *Report commissioned for the National Council for Curriculum and Assessment*.

OCR. (2014). A Level Specifaction: Computer Science (pp. 52). Cambridge: OCR Publications.

Sagan, C. (1997). *The Demon-Haunted World: Science as a Candle in the Dark.*: Ballantine Books.

Scott, J., & Bundy, A. (2015). Creating a new generation of computational thinkers. *Communications of the ACM, 58*(12), 37-40.

Scottish Qualifications Authority. (2012). Course Aims and Structure. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2013a). Pre-Appeals National Course and Awards Result Statistics 6 August 2013 - Female Candidates. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2013b). Pre-Appeals National Course and Awards Result Statistics 6 August 2013 - Male Candidates. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2014). Attainment Statistics (December) 2014. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2015a). Advanced Higher Computing Science Course Assessment Specification (C716 77) (pp. 11). Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2015b). Attainment Statistics (August) 2015 Female. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2015c). Attainment Statistics (August) 2015 Male. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2015d). Higher Computing Science Course Specifaction (C716 76). Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2015e). Information Systems Design and Development (Advanced Higher) Unit. Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2015f). Information Systems Design and Development (Highers) Unit. Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2015g). Software Design and Development (Advanced Higher)Unit. Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2015h). Software Design and Development (Highers) Unit. Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2016a). Advanced Higher Computing Science Course Specification (C716 77) (pp. 12). Glasgow: SQA Publications,.

Scottish Qualifications Authority. (2016b). Attainment Statistics (August) 2016. On-Line: SQA Publications,.

Scottish Qualifications Authority. (2016c). Higher Computing Science Course Assessment Specification (C716 76). Glasgow: SQA Publications,.

Sentance, S. (2016). A holistic approach to teacher professional development in Computing (Vol. 2016). http://blogs.kcl.ac.uk/: Computer Science Education @ King's.

Sentance, S., & Csizmadia, A. (2015). Teachers’ perspectives on successful strategies for teaching Computing in school. *Paper presented at IFIP TCS, 2015*.

Sentance, S., Dorling, M., & McNicol, A. (2013). *Computer science in secondary schools in the UK: Ways to empower teachers.* Paper presented at the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.

Sentance, S., Dorling, M., McNicol, A., & Crick, T. (2012). *Grand challenges for the UK: upskilling teachers to teach computer science within the secondary curriculum.* Paper presented at the Proceedings of the 7th Workshop in Primary and Secondary Computing Education.

Sentance, S., Humphreys, S., & Dorling, M. (2014). *The network of teaching excellence in computer science and master teachers.* Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education.

Snyder, L. (2012). Status update: high school CS internationally. *ACM Inroads, 3*(2), 82-85. doi:10.1145/2189835.2189861

Stephenson, C., Gal-Ezer, J., Haberman, B., & Verno, A. (2005). The New Educational Imperative: Improving High School Computer Science Education. Using Worldwide Research and Professional Experience to Improve U. S. Schools. *Final Report of the CSTA Curriculum Improvement Task Force*.

Sturman, L., & Sizmur, J. (2011). International comparison of computing in schools. *Slough: NFER*.

The Chartered Institute for IT (BCS). (nd). Plan C *Professional Learning and Networking for Computing*: The Chartered Institute for IT.

The Office of Qualifications and Examinations Regulation. (2016a). Official Statistics: Sumamer 2016 exam entries *GCSEs,Level 1/2 certificates, AS and A levels in England*. On-lIne: Ofqual.

The Office of Qualifications and Examinations Regulation. (2016b). Statutory guidance : GCSE, AS and A level Assessment Objectives. On-Line: Ofqual.

Thompson, D., & Bell, T. (2013). *Adoption of new computer science high school standards by New Zealand teachers*. Paper presented at the Proceedings of the 8th Workshop in Primary and Secondary Computing Education.

Thompson, D., Bell, T., Andreae, P., & Robins, A. (2013). *The role of teachers in implementing curriculum changes*. Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education.

University of Strathclyde. (ND). Computing: General Regulations and Special Requirements. Retrieved from http://www.strath.ac.uk/courses/postgraduatetaught/secondaryeducation/computing/

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, 1-24.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). Running on Empty *Running on Empty*. CSTA: The Computer Science Teachers Association,The Association for Computing Machinery.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Zur Bargury, I. (2012). *A new curriculum for junior-high in computer science*. Paper presented at the Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education.

Zur Bargury, I., Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., & Hotoveli, R. (2012). *Implementing a new Computer Science Curriculum for middle school in Israel*. Paper presented at the 2012 Frontiers in Education Conference Proceedings.

Appendix

Associated Research & Bibliography

Introduction. During this research effort, an analysis of the available research on Computer Science Education(CSE) in the upper second level of Secondary Schools was conducted within the report jurisdictions.

An important finding from this analysis was the relative strength of a research tradition in CSE in each of our jurisdictions, as evaluations and studies are important for reflective practice to enhance and advance the implementation and management of educational programmes. In total, forty-four research papers in peer reviewed academic publications were reviewed. Israel had the greatest breadth and depth of research of the jurisdictions with twenty-four publications. This is partly due to being the longest established programme. But it is also due to an acknowledgement and focus within the CSE community on the importance of active research that contributes to the vibrancy and development of the subject. A search for research on Ontario's programme returned only one paper of relevance, with jurisdictions that have revised their curriculum more recently, New Zealand, Scotland and England yielding ten and nine papers respectively.

In tables 12 and13, the lead author of each publication, along with the key topics which the author covers, in their research to bring focus on the prominent researchers and topics in each jurisdiction is listed. This simplifies the portrayal of the research as many of the authors have co-written multiple papers, but as a device serves to highlight notable bodies of research.

Israel	**Judith Gal-Ezer et al.**	**Orit Hazzan et al.**	**Michal Armoni et al.**	**Noa Ragonis et al.**	**Tami Lapidot et al.**
	6	**4**	**3**	**3**	**3**
	Algorithmics, Computer Science, **CS Curriculum**, Education, High-School, Secondary Computing Education, **Underlying Principles**, Zipper Principle, Methods of Teaching, Practicum, **Teacher Preparation,** Certification, Distance Learning	High School, **CS Teacher Preparation**, Methods of Teacher Training, **Practicum**, 3 Day Seminars, Curriculum, **Success factors**, CS Education Research, Teaching Certificate, Methods	Software Engineering, Student Experience, High School, **Pipeline Higher Education**, Gender, CS Secondary Teacher Education, **Pre-service Teachers,** Didactics	Computer Science Education, Prospective Teachers, **Pre-service Training**, Teacher Preparation Curriculum, K-12 Instruction, **Methods of Teaching CS**, Practicum	**Leading Teachers**, Summer Seminars, CS education, Teacher Professional Development, **Teacher Training**, Teaching Methods, Pre-service, Professional Leadership
	Bruria Haberman et al.	**Iris Zur Bargury et al.**	**Haim Averbuch et al.**	**Ofra Brandes et al.**	
	2	**1**	**1**	**1**	
	Distance Learning, In-service Training, Local Workshops, **Teacher Training**	Algorithmic Thinking, Computational Thinking, CS, Curriculum, **Middle School**, Teachers Professional Development	Computer Science Education, **Assessment**, K12	**Leading Teachers**, In-Service Teachers	

Table 12: Lead Author, Number Publications (CSE Upper Second Level)

Ontario	**Jacob Slonim et al.**	***New Zealand***	**Tim Bell et al.**	**David Thompson et al.**	**Sumant Murugesh et al.**
	1		**7**	**2**	**1**
	Pipeline Higher Education, CSE		**Computer Science Curriculum**, High School, CS Field Guide, **Teachers Role**, Challenges, Implementation Challenges, Darmstadt, Constructivism, Open Source, Assessment, Algorithms, Programming Languages, NCEA, Engagement,	CS Education, Assessment, Teachers Supports, Confidence, **Role of Teachers, Teacher Professional Development,** Curriculum Implementation	Computer Science, Pedagogies, Programming, Kinaesthetic Activities, Information Technology Education
England & Scotland	**Sue Sentence et al.**	**Neil Brown et al.**	**Tom Crick et al.**	**Jeremy Scott et al.**	
	5	**2**	**1**	**1**	
	Computer science education, Pedagogy, Primary computing education, Secondary, computing education, **Teachers' perspectives**, Computational thinking, **Teacher Professional Development,** CS Teacher Education, **Community of Practice, CAS, Continuing Professional Development,** Curriculum, Master Teachers	Computer Science Education, High School, **Teachers**, Curriculum, Challenges, **CAS, Teacher Professional Development**	**CAS**, Computer Science Education, STEM, UK education Policy, Computing	Curriculum for Excellence, Exemplification Project, Pedagogy	

Table 13: Lead Author, Number Publications (CSE Upper Second Level) (Cont.)

Bibliography

England

Peer Reviewed Research

Brown, N. C. C., Kölling, M., Crick, T., Jones, S. P., Humphreys, S., & Sentance, S. (2013). *Bringing computer science back into schools: Lessons from the UK*. Paper presented at the SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education.

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602484

Crick, T., & Sentance, S. (2011). *Computing at school: stimulating computing education in the UK*. Paper presented at the Proceedings of the 11th Koli Calling International Conference on Computing Education Research.

Sentance, S. (2016). A holistic approach to teacher professional development in Computing (Vol. 2016). http://blogs.kcl.ac.uk/: Computer Science Education @ King's.

Sentance, S., & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching Computing in school. *Paper presented at IFIP TCS, 2015*.

Sentance, S., Dorling, M., & McNicol, A. (2013). *Computer science in secondary schools in the UK: Ways to empower teachers*. Paper presented at the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.

Sentance, S., Dorling, M., McNicol, A., & Crick, T. (2012). *Grand challenges for the UK: upskilling teachers to teach computer science within the secondary curriculum.* Paper presented at the Proceedings of the 7th Workshop in Primary and Secondary Computing Education.

Sentance, S., Humphreys, S., & Dorling, M. (2014). *The network of teaching excellence in computer science and master teachers*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education.

Israel

Peer Reviewed Research

Armoni, M. (2011). Looking at secondary teacher preparation through the lens of computer science.

ACM Transactions on Computing Education (TOCE), 11(4), 23.

Armoni, M. (2016). Computing in Schools: High-school software engineering education: personal impressions of an 18-year-old student. *ACM Inroads, 7*(2), 25-27.

Armoni, M., & Gal-Ezer, J. (2014). High school computer science education paves the way for higher education: the Israeli case. *Computer Science Education, 24*(2-3), 101-122.

Averbuch, H., Benaya, T., & Zur, E. (2011). *Criteria for writing exams which reflect the k12 CS foundations study material.* Paper presented at the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.

Brandes, O., Vilner, T., & Zur, E. (2010). *Software design course for leading CS in-service teachers.* Paper presented at the International Conference on Informatics in Secondary Schools-Evolution and Perspectives.

Gal-Ezer, J. (2012). *Grand challenges in primary and secondary computing education.* Paper presented at the Proceedings of the 7th Workshop in Primary and Secondary Computing Education.

Gal-Ezer, J., Beeri, C., Harel, D., & Yehudai, A. (1995). A High School Program in Computer Science. *Computer, 28*(10), 73-80. doi:10.1109/2.467599

Gal-Ezer, J., & Harel, D. (1999). Curriculum and Course Syllabi for a High-School Program in Computer Science. *Computer Science Education, 9*(2), 114-147.

Gal-Ezer, J., Hazzan, O., & Ragonis, N. (2009). *Preparation of high school computer science teachers: the Israeli perspective.* Paper presented at the ACM SIGCSE Bulletin.

Gal-Ezer, J., Shahak, D., & Zur, E. (2009). *Computer science issues in high school: gender and more.* Paper presented at the ACM SIGCSE Bulletin.

Gal-Ezer, J., & Stephenson, C. (2014). A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602483

Haberman, B., & Cohen, A. (2007). A high-school programme in software engineering. *International Journal of Engineering Education, 23*(1), 15-23.

Haberman, B., & Ginat, D. (1999). *Distance learning model with local workshop sessions applied to in-service teacher training.* Paper presented at the iTiCSE 99, Cracow, Poland.

Hazzan, O. High School CS Teacher Preparation: Key Elements, Structure and Challenges.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). *A model for high school computer science education: the four key elements that make it!* Paper presented at the Conference'08, Portland, Oregon, USA.

Hazzan, O., Gal-Ezer, J., & Ragonis, N. (2010). How to establish a computer science teacher preparation program at your university? The ECSTPP workshop. *ACM Inroads, 1*(1), 35-39.

Hazzan, O., & Lapidot, T. (2004). The practicum in computer science education: bridging gaps between theoretical knowledge and actual performance. *ACM SIGCSE Bulletin, 36*(4), 47-51.

Lapidot, T., & Aharoni, D. (2007). The Israeli summer seminars for CS leading teachers. *ACM SIGCSE Bulletin, 39*(3), 318-318.

Lapidot, T., & Aharoni, D. (2008). On the frontier of computer science: Israeli summer seminars. *ACM SIGCSE Bulletin, 40*(4), 72-74.

Lapidot, T., & Hazzan, O. (2003). Methods of teaching a computer science course for prospective teachers. *ACM SIGCSE Bulletin, 35*(4), 29-34.

Ragonis, N., & Hazzan, O. (2008). *Disciplinary-pedagogical teacher preparation for pre-service Computer Science teachers: Rational and implementation.* Paper presented at the International Conference on Informatics in Secondary Schools-Evolution and Perspectives.

Ragonis, N., Hazzan, O., & Gal-Ezer, J. (2010). *A survey of computer science teacher preparation programs in Israel tells us: computer science deserves a designated high school teacher preparation!* Paper presented at the Proceedings of the 41st ACM technical symposium on Computer science education.

Ragonis, N., Hazzan, O., & Gal-Ezer, J. (2011). *A study on attitudes and emphases in computer science teacher preparation.* Paper presented at the Proceedings of the 42nd ACM technical symposium on Computer science education.

Zur Bargury, I., Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., & Hotoveli, R. (2012). *Implementing a new Computer Science Curriculum for middle school in Israel.* Paper presented at the 2012 Frontiers in Education Conference Proceedings.

New Zealand

Peer Reviewed Research

Bell, T. (2014). Establishing a nationwide CS curriculum in New Zealand high schools. *Communications of the ACM, 57*(2), 28-30.

Bell, T., Andreae, P., & Lambert, L. (2010). Computer science in New Zealand high schools. *Proceedings of the Twelfth Australasian Conference on Computing Education, 103*, 15-22.

Bell, T., Andreae, P., & Robins, A. (2012). *Computer science in NZ high schools: the first year of the new standards.* Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.

Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education, 14*(2). doi:10.1145/2602485

Bell, T., Duncan, C., Jarman, S., & Newton, H. (2014). Presenting computer science concepts to high

school students.

Bell, T., Newton, H., Andreae, P., & Robins, A. (2012). *The introduction of computer science to NZ high schools: an analysis of student work.* Paper presented at the Proceedings of the 7th Workshop in Primary and Secondary Computing Education.

Bell, T., Newton, H., Duncan, C., & Jarman, S. (2014). *Adoption of Computer Science in NZ schools.* Paper presented at the Proceedings of ITx-New Zealand's Conference of IT.

Murugesh, S., Bell, T., & McGrath, A. (2010). *A review of computer science resources to support NCEA.* Paper presented at the First annual conference of Computing and Information Technology Research and Education NZ (CITRENZ2010).

Thompson, D., & Bell, T. (2013). *Adoption of new computer science high school standards by New Zealand teachers.* Paper presented at the Proceedings of the 8th Workshop in Primary and Secondary Computing Education.

Thompson, D., Bell, T., Andreae, P., & Robins, A. (2013). *The role of teachers in implementing curriculum changes.* Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education.

Ontario

Peer Reviewed Research

Slonim, J., Scully, S., & McAllister, M. (2008). Crossroads for Canadian CS enrolment. *Communications of the ACM, 51*(10), 66-70.

Scotland

Peer Reviewed Research

Scott, J., & Bundy, A. (2015). Creating a new generation of computational thinkers. *Communications of the ACM, 58*(12), 37-40.

International

Peer Reviewed Research

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for Computer Science in the School Curriculum.

Hubwieser, P. (2013). The darmstadt model: A first step towards a research framework for computer science education in schools. Keynote talk. *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2013*, 1-14.

Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M. N., . . . Schubert, S. (2011). *Computer Science/informatics in secondary education.* Paper presented at the Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE.

Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to implement rigorous computer science education in K-12 schools? Some answers and many questions. *ACM Transactions on Computing Education, 15*(2). doi:10.1145/2729983

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., . . . Jasute, E. (2015). *A global snapshot of computer science education in K-12 schools.* Paper presented at the ITiCSE-WGP 2015 - Proceedings of the 2015 ITiCSE Conference on Working Group Reports.

Informatics Europe, & ACM Europe. (2013). Informatics education: Europe cannot afford to miss the boat *Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education*.

Jones, S. P. (2011). Computing at School *International comparisons* (Vol. 7, pp. 2013). London: Computing at School

Levin, B., & Segedin, L. (2011). *International Approaches to Secondary Education*. Retrieved from Ontario:

Schulte, C., Hornung, M., Sentence, S., Dagiene, V., Jevsikova, T., Thota, N., . . . Peters, A. K. (2012). *Computer science at school/CS teacher education - Koli working-group report on CS at school.* Paper presented at the Proceedings - 12th Koli Calling International Conference on Computing Education Research, Tanko, Finland.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., . . . Verno, A. (2011). CSTA K--12 Computer Science Standards: Revised 2011.

Snyder, L. (2012). Status update: high school CS internationally. *ACM Inroads, 3*(2), 82-85. doi:10.1145/2189835.2189861

Stephenson, C., Gal-Ezer, J., Haberman, B., & Verno, A. (2005). The New Educational Imperative:

Improving High School Computer Science Education. Using Worldwide Research and Professional Experience to Improve U. S. Schools. *Final Report of the CSTA Curriculum Improvement Task Force*.

Sturman, L., & Sizmur, J. (2011). International comparison of computing in schools. *Slough: NFER*.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, 1-24.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). Running on Empty *Running on Empty*. CSTA: The Computer Science Teachers Association, The Association for Computing Machinery.