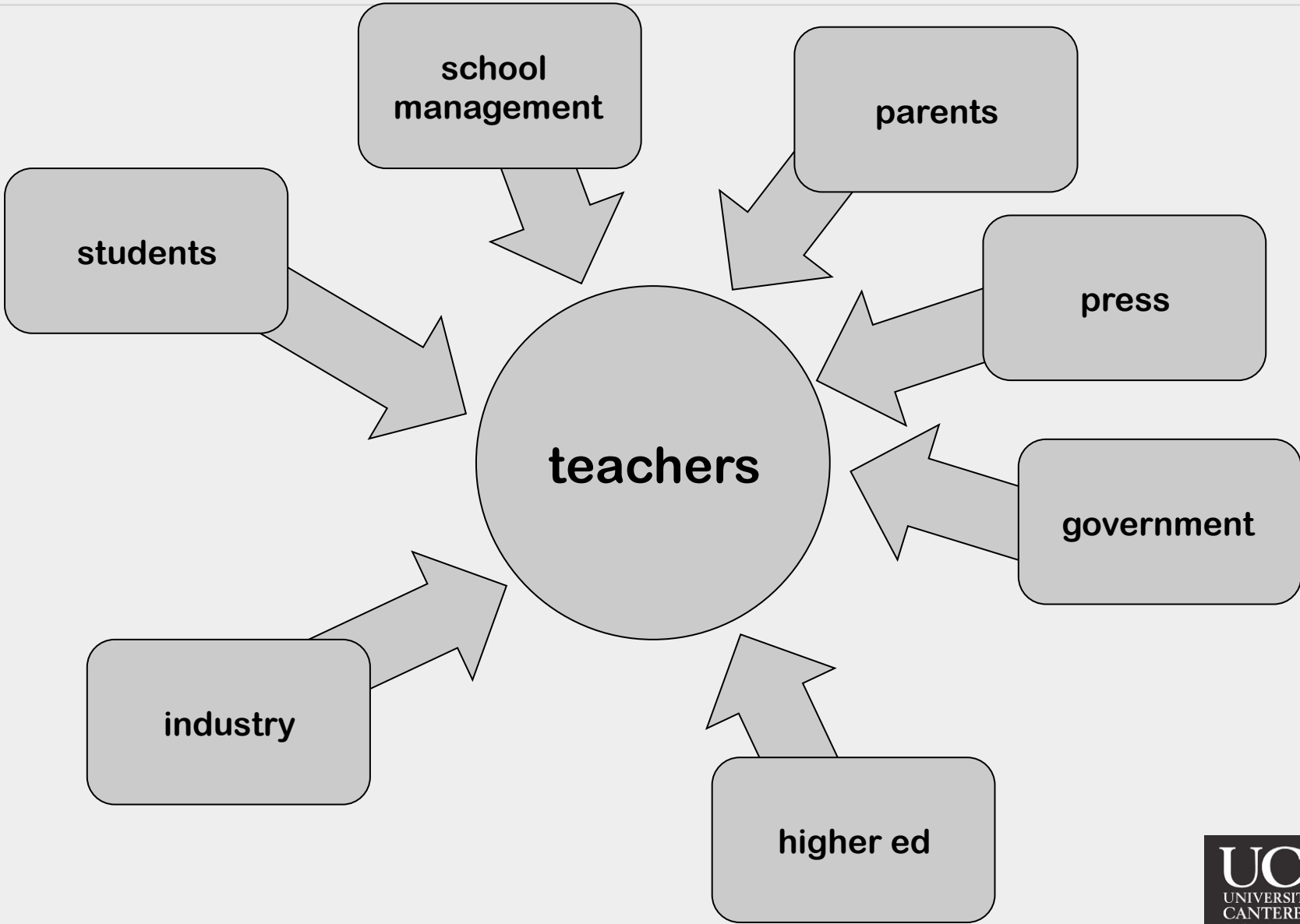


# Adoption of new Computer Science high school standards by New Zealand teachers

David Thompson, Tim Bell  
University of Canterbury  
New Zealand

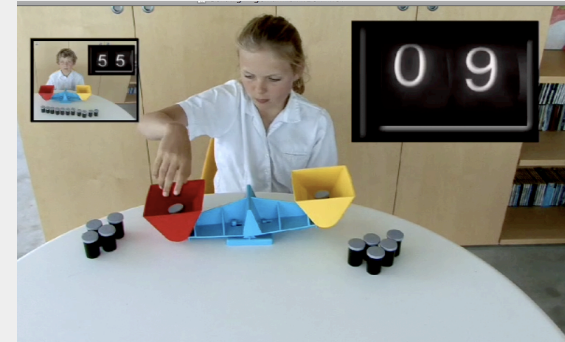
# Teachers are the lynchpin



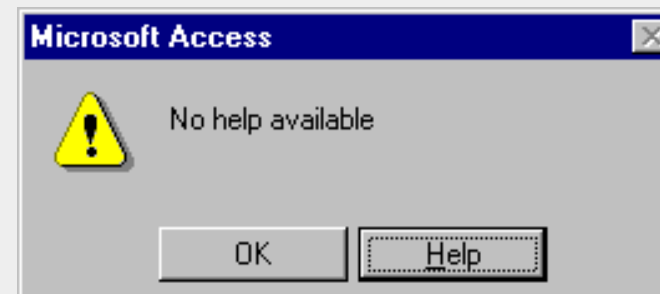
Computer Science in NZ Schools – teacher adoption

# Level 1 Computer science

- Algorithm comparisons
- Programming languages
- Human Computer Interaction



```
user1$ gcc selectionSort.c
user1$ ./a.out
Setting up array...
Selection sort demo with 50000 items...
Finished test
user1$
```



# AS91074

Computer Science in NZ Schools – teacher adoption

## Achievement Standard

**Subject Reference** Digital Technologies 1.44

**Title** Demonstrate understanding of basic computer science

**Level** 1 **Credits** 3 **Area** A

**Subfield** Technology

**Domain** Digital Technologies

**Status** Registered **Status date**

**Planned review date** 31 December 2014 **Date version published**

This achievement standard requires demonstrating understanding of basic computer science.

### Achievement Criteria

Achievement	Achievement with Merit
<ul style="list-style-type: none"> <li>Demonstrate understanding of basic concepts from computer science.</li> </ul>	<ul style="list-style-type: none"> <li>Demonstrate in-depth understanding of basic concepts from computer science.</li> </ul>

### Explanatory Notes

- This achievement standard is derived from the Level 6 achievement standard in the Technology learning area in *The New Zealand Curriculum* (Ministry of Education, 2007), and is related to the material in *Learning Guide for Technology*, Ministry of Education, 2007, available at <http://seniorsecondary.tki.org.nz>.  
Further information can be found at <http://www.techlink.org.nz>.
- Demonstrate understanding of basic concepts from computer science* involves:
  - describing the key characteristics and roles of algorithm instructions
  - describing an algorithm for a task, showing understanding of how that can be in an algorithm, and determining the cost of a problem of a particular size

- describing the role and characteristics of programming languages, including the different roles and characteristics of high level languages and low level (or machine) languages, and the function of a compiler
- describing the role of a user interface and factors that affect its usability

*Demonstrate in-depth understanding of basic concepts from computer science* involves:

- explaining how algorithms are distinct from related concepts such as procedures and informal instructions
- showing understanding of the way steps in an algorithm are combined in sequential, conditional, and iterative structures, and the cost of an iterative algorithm for a problem of size  $n$
- explaining how the characteristics of programming languages at different levels (high level and low level) are different and important for their roles
- explaining the need for programs to translate between high level and low level languages
- explaining how different factors of a user interface affect its usability

*Demonstrate comprehensive understanding of basic concepts from computer science* involves:

- comparing and contrasting the concepts of algorithm instructions
- determining and comparing the costs of two different algorithms for the same problem of size  $n$
- comparing and contrasting high level and low level (or machine) languages, explaining different ways in which programs in a high level language are translated into a machine language
- discussing how different factors of a user interface affect its usability, comparing and contrasting related interfaces.

- The *basic concepts from computer science* are: the concept of a programming language; and the concept of usability.
- An algorithm is a precise unambiguous specification of how to solve a computational task in a finite number of well-defined steps that can be followed from a computer program. An algorithm has a cost (the number of instructions performed) for a task. Different algorithms for the same task have different costs.
- A programming language is a precise, formal language that can be run on a computer; it is distinct from pseudocode and is different from machine level to low level (or machine) languages.
- A user interface is the part of a computer or electronic system that a user interacts with to control the system. The usability of an interface is a characteristic for evaluating an interface.
- Assessment Specifications for this achievement standard are available on the Technology Resources page found at <http://www.nzqa.govt.nz/assessment-specifications/>.

### Quality Assurance

- Providers and Industry Training Organisations must be accredited by NZQA before they can register credits from assessment against achievement standards.
- Accredited providers and Industry Training Organisations assessing against achievement standards must engage with the moderation system that applies to those achievement standards.

Accreditation and Moderation Action Plan (AMAP) reference 0233

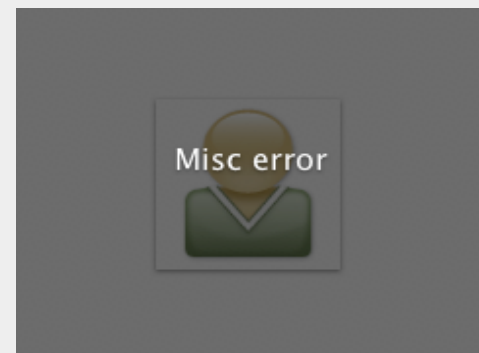
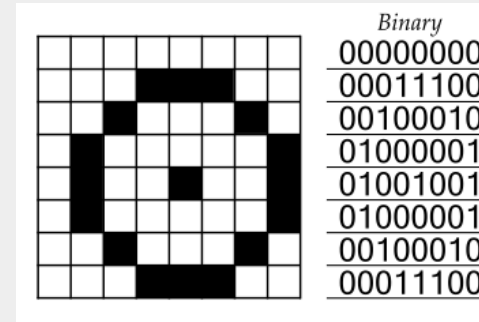
# Excellence

*Demonstrate comprehensive understanding of basic computer science concepts from computer science involves:*

- comparing and contrasting the concepts of algorithms, programs, and informal instructions
- determining and comparing the costs of two different iterative algorithms for the same problem of size  $n$
- comparing and contrasting high level and low level (or machine) languages, and explaining different ways in which programs in a high level programming language are translated into a machine language
- discussing how different factors of a user interface contribute to its usability by comparing and contrasting related interfaces

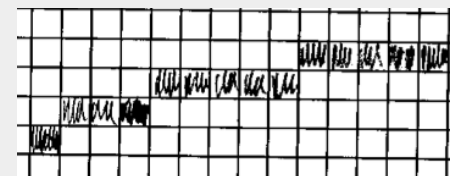
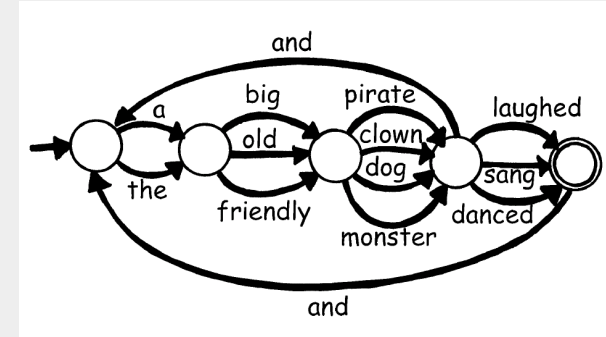
# Level 2 Computer Science

- representing data using bits
- encoding: compression, error, encryption
- HCI and usability heuristics



# Level 3 Computer Science

- formal languages
- network communication protocols
- complexity and tractability
- intelligent systems
- software engineering
- graphics and visual computing



# nzacditt.org.nz



Log out

- Home
- About Us
- Events
- News
- Regions
- Resources

[Home](#) > Welcome to the NZACDITT website

[View](#) [Edit](#) [Revisions](#)

## Welcome to NZACDITT

The **New Zealand Association for Computing, Digital and Information Technology Teachers** is an association with the goal of advocating for our subjects. The aim of the association is to create a community of teachers where we can share resources, communicate and speak with one voice to get our subject area recognised and supported.

There is so much happening in the Digital Technologies space at the moment, read about some of them in the latest [T-News Magazine](#). This is a regular Newsletter that has loads of information for us.

## Changes to NZACDITT Google Group - Feb 19 2013

This is an advanced warning that the NZACDITT Google groups will only be available to paid-up members of NZACDITT. The cross over will happen on the 1st of April 2013. NZACDITT (Your subject Association) manages the Google Group. It is the main vehicle for discussion and resources, (and was open to all people subject to approval). NZACDITT want to make this resource available to paid members only so that we can start to provide more dedicated support to our membership.

## Join your Region when you Join NZACDITT

Join the region you represent when you are a member of NZACDITT. We want to build up the Professional Capabilities of the regions. We need to start organising at a regional level to get collaboration and sharing happening.

## Latest News

NZQA Digital Technologies External Assessment Workshops - 91070  
06 March, 2013

[Read more ...](#)

National Newsletter Technology - February  
11 February, 2013

Kia ora, ATTENTION: The Principal, PD Coordinator and Secondary Teachers of Technology Please find attached the National Technology Newsletter for...

[Read more ...](#)

Digital Information - Oracle  
10 February, 2013

[Read more ...](#)

[View all news](#)

## Coming up ...



# Surveys

	Feb 2012	May 2013
Respondents	89	109
Size of mailing list	404	216
Response rate	22.0%	50.5%
Number of different schools	69	87
Male/Female	48.9%/51.1%	50.5%/49.5%
50+ years old	60.4%	56.0%

- Comparing only teachers from same schools (40 in common)

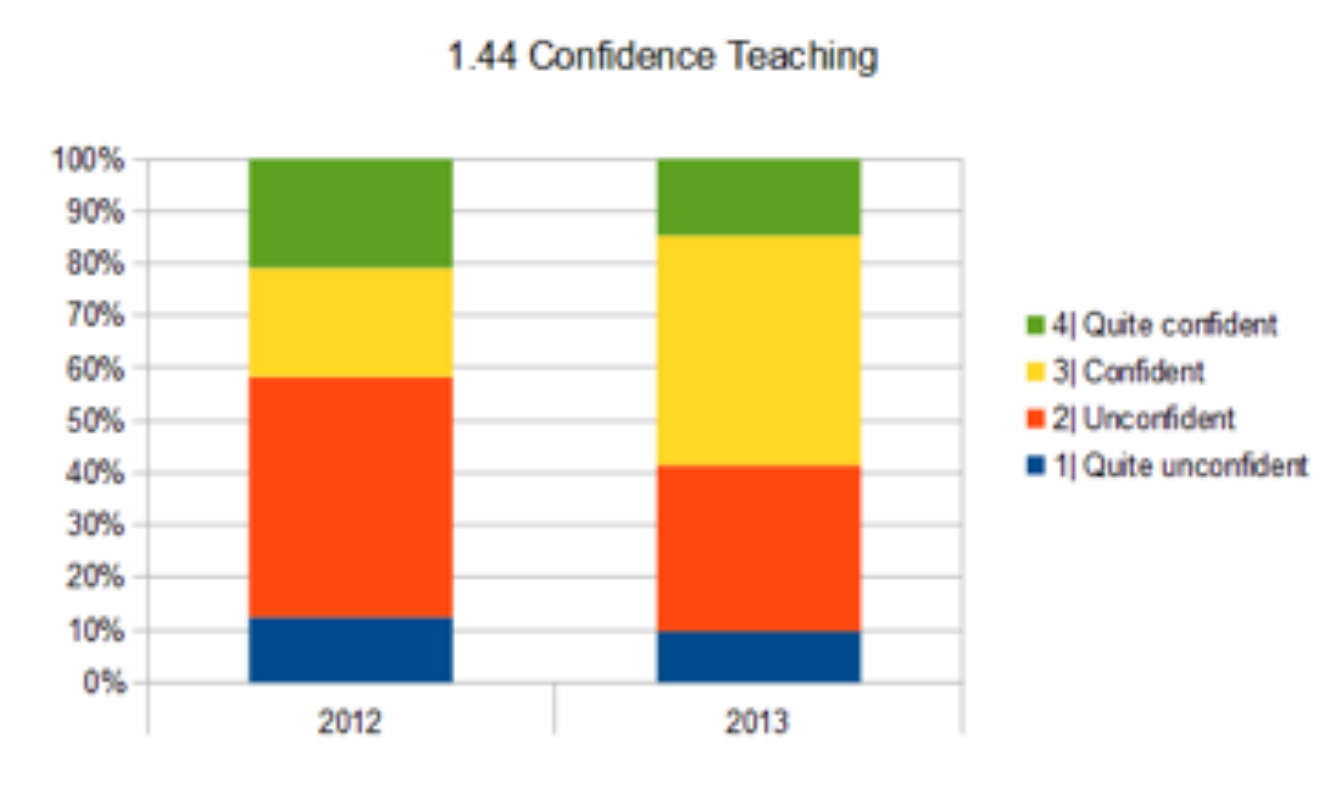
# Qualification (2012)

---

- 56% have a computing qualification
- 11% have a CS degree

# Confidence teaching?

Computer Science in NZ Schools – teacher adoption



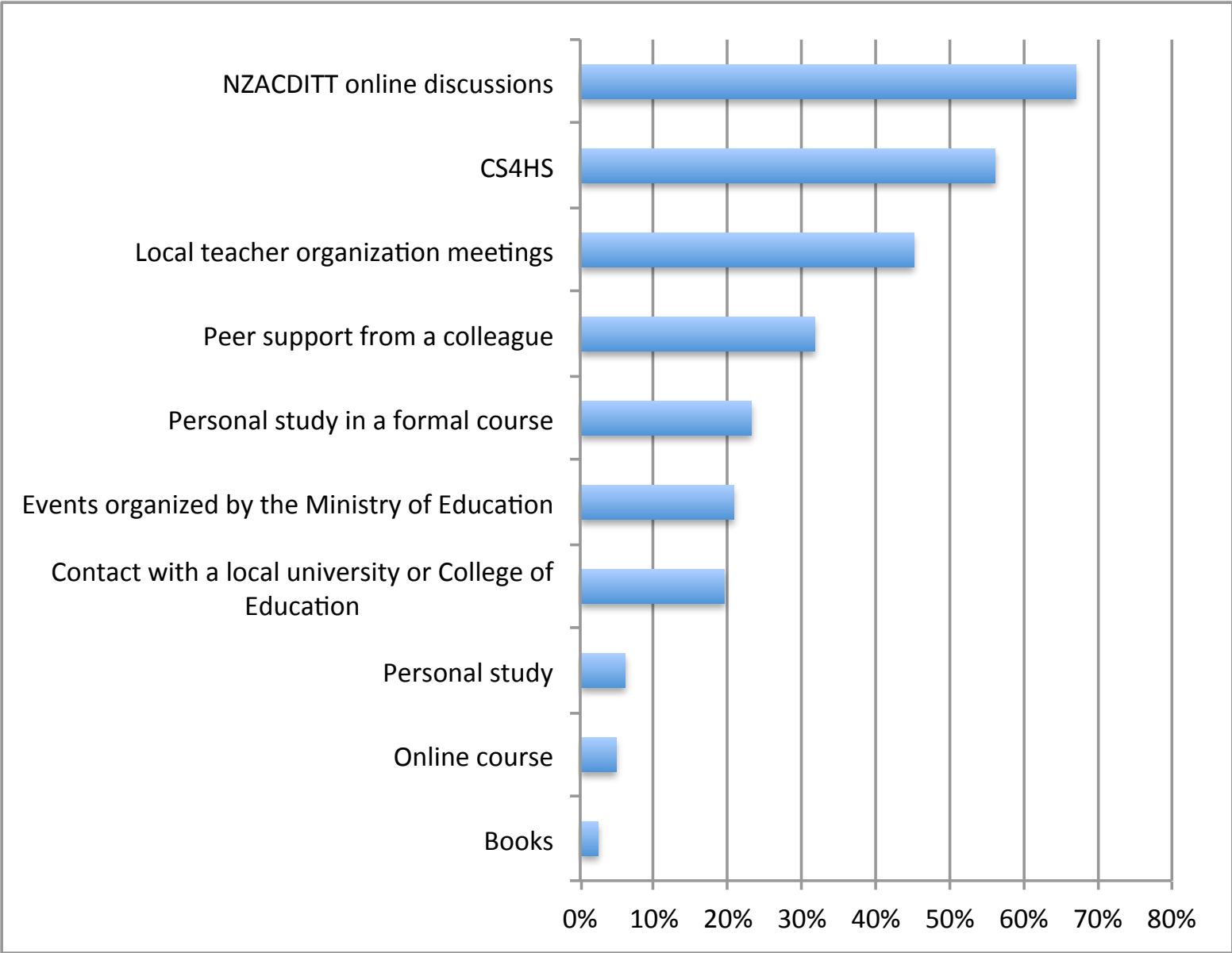
# Fair assessment

---

- 2.75 (2012) to 2.86 (2013)
- 5% (2012) to 0 (2013)  
strongly disagree

# Sources of Professional Development

Computer Science in NZ Schools – teacher adoption



# Adoption of 1.44 standard

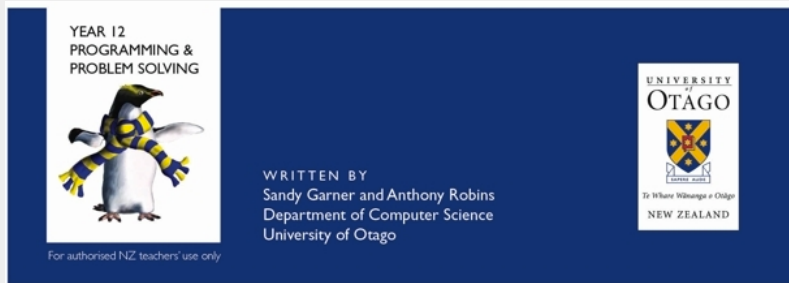
---

- Local support or formal study:  
60% did not adopt
- Peer support, university contact and CS4HS:  
30% did not adopt

# Financial support for PD

- 42% good support
- 50% partial support
- 7% self funded
- Biggest issues:
  - Time
  - Opportunities
  - Overwhelmed/exhausted

# Resources used



### Computer Science Field Guide

Student Version

#### Table Of Contents

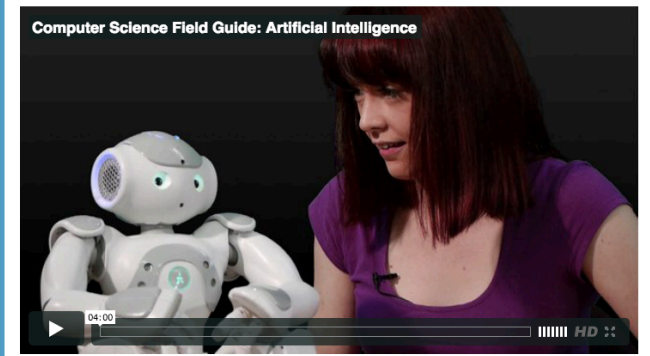
- 1. Introduction
- 2. Data representation
- 3. Algorithms
- 4. Human-computer interaction and usability
- 5. Programming languages
- 6. Coding - Introduction
- 7. Compression coding
- 8. Encryption coding
- 9. Error control coding
- 10. Artificial Intelligence**
  - 10.1. What's the big picture?
  - 10.2. Chatterbots and The Turing Test
  - 10.3. The whole story!
  - 10.4. Further reading
- 11. Complexity and tractability
- 12. Formal languages
- 13. Computer graphics
- 14. Computer vision
- 15. Network communication protocols
- 16. Software engineering
- 17. Appendices

Quick search

Feedback [Please place your feedback here](#)

## 10. Artificial Intelligence

**Warning:** This chapter is still in development; the current version has been released for the 2013 school year with enough material to enable students to complete the NZ 3.44 achievement standard, but more sections may be added during the year to give more options for student projects.



### 10.1. What's the big picture?

Artificial Intelligence conjures up all sorts of images — perhaps you think of friendly systems that can talk to you and solve tough problems; or maniac robots that are bent on world domination? There's the promise of driverless cars that are safer than human drivers, and the worry of medical advice systems that hold peoples lives in their virtual hands. The field of Artificial Intelligence is a part of computer science that has a lot of promise and also raises a lot of concerns. It can be used to make decisions in systems as large as an aeroplane or an *autonomous dump truck*, or as small as a mobile phone that accurately predicts text being typed into it. What they have in common is that they try to mimic aspects of human intelligence. And importantly, such systems can be of significant help in people's everyday lives.

AI (also known as intelligent systems) is primarily a branch of computer science but it has borrowed a lot of concepts and ideas from other fields, especially mathematics (particularly logic, combinatorics, statistics, probability and optimisation theory), biology, psychology, linguistics, neuroscience and philosophy.

In this chapter we'll explore a range of these intelligent systems. Inevitably this will mean dealing with ethical and philosophical issues too — do we really want machines to take over some of our jobs? Can we trust them? Might it all go too far one day? What do we really mean by a computer being intelligent? While we won't address these questions directly in this chapter, gaining some technical knowledge about AI will enable you to make more informed decisions about the deeper issues.

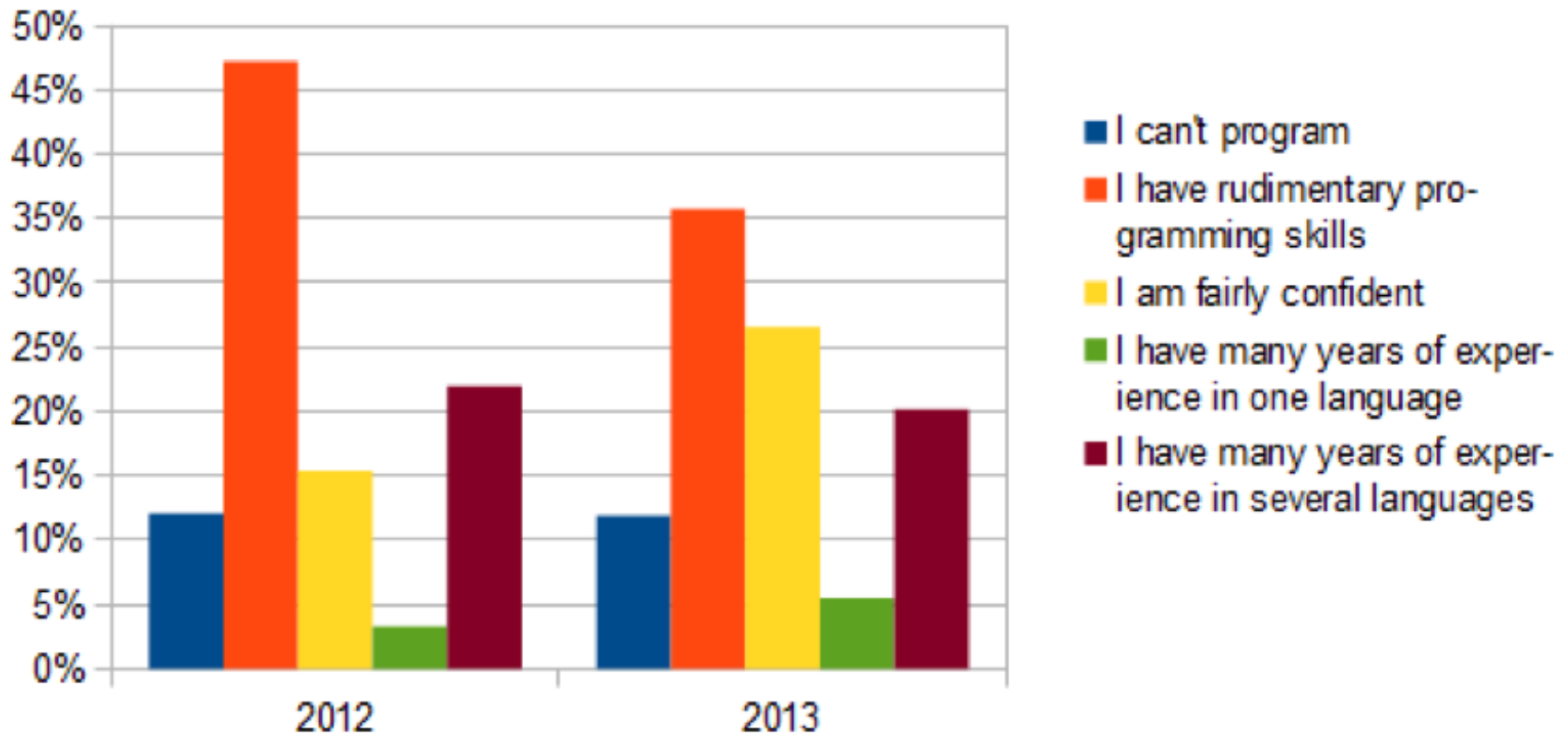
### 10.2. Chatterbots and The Turing Test





# Programming ability (2012)

What is your level of programming experience?



# Student achievement

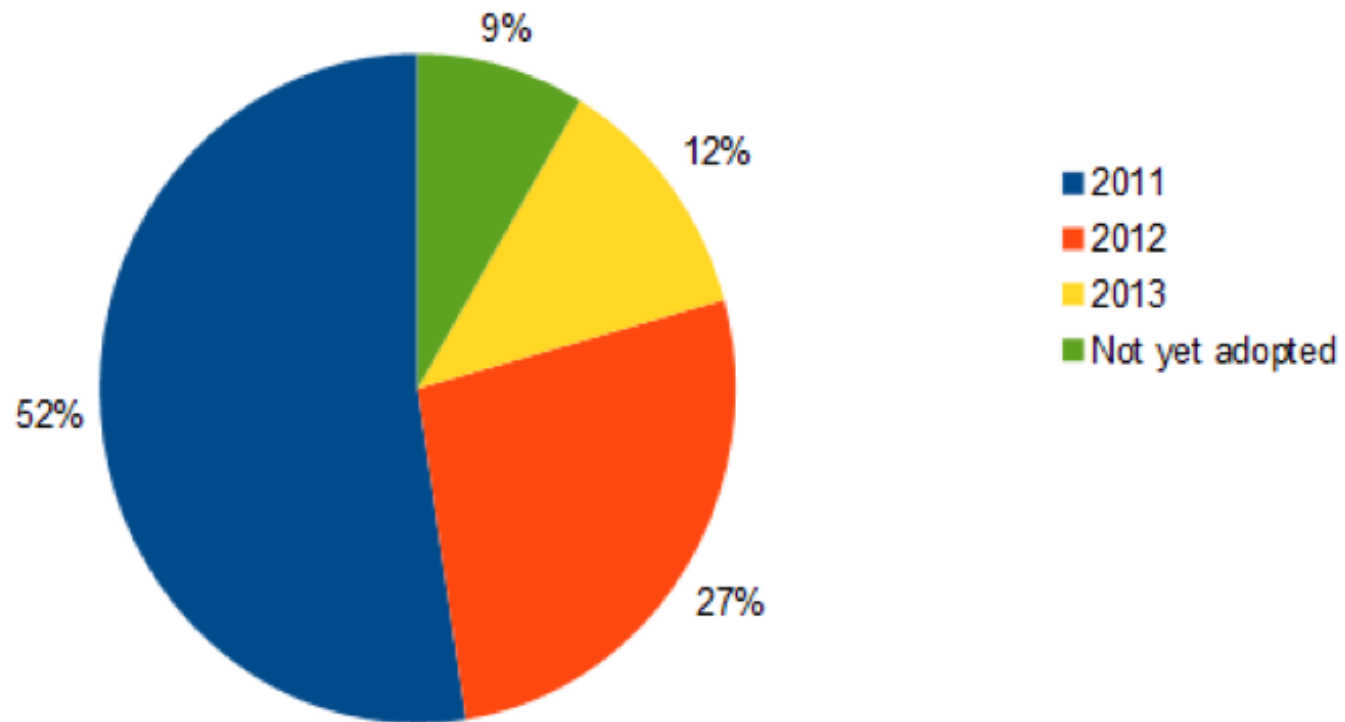
---

- 21% to 30% female
- Female achievement: 2.23
- Male achievement: 2.06

# First adopted standards

Computer Science in NZ Schools – teacher adoption

When did you first adopt any of the new programming and computer science standards?

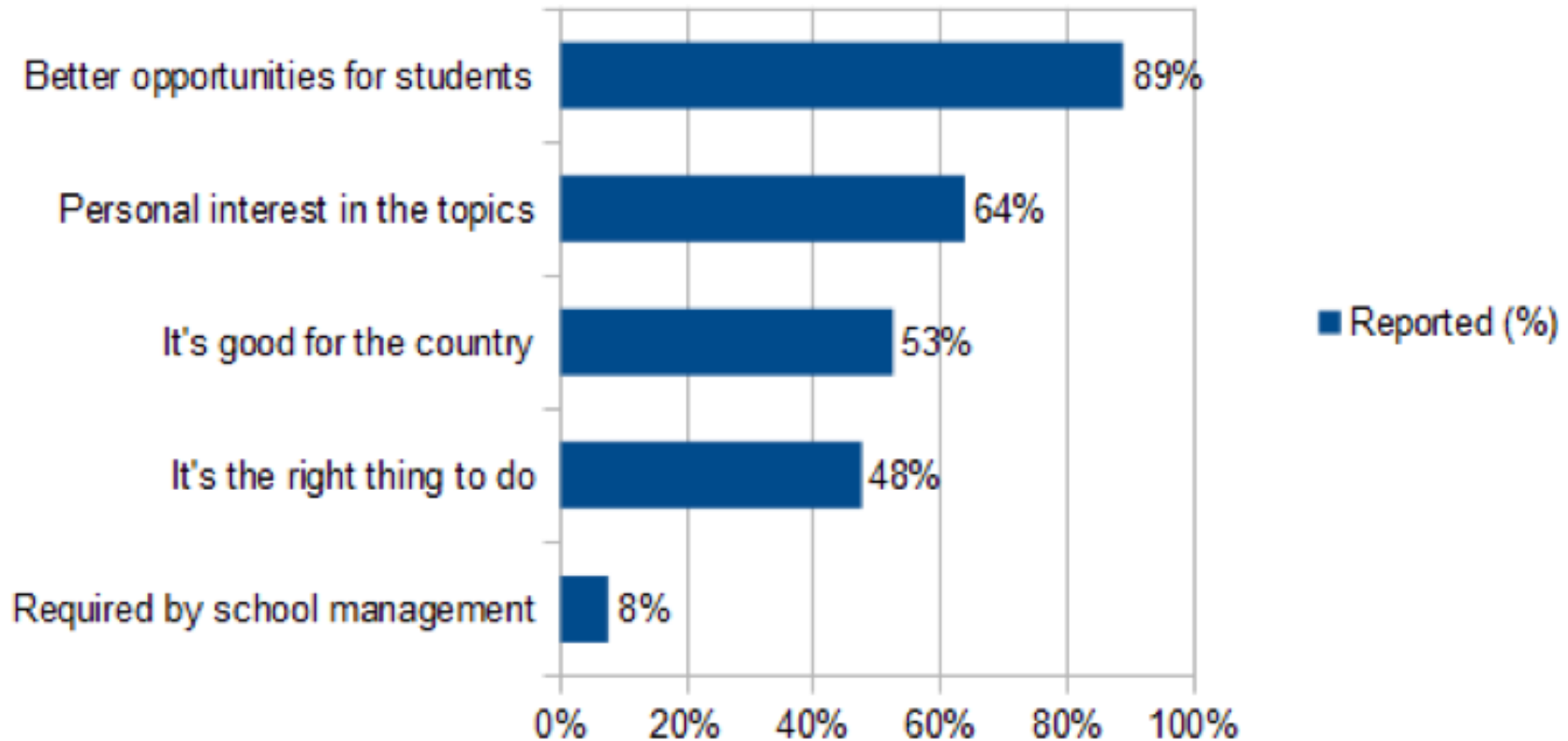


# Areas of Digital Technologies

	Feb 2012	May 2013	change
Digital information	84.3%	84.8%	+0.5%
Digital infrastructure	41.2%	32.6%	-8.6%
Digital media	90.2%	93.5%	+3.3%
Electronics	9.8%	15.2%	+5.4%
<b>Programming and computer science</b>	62.7%	82.6%	<b>+19.9%</b>

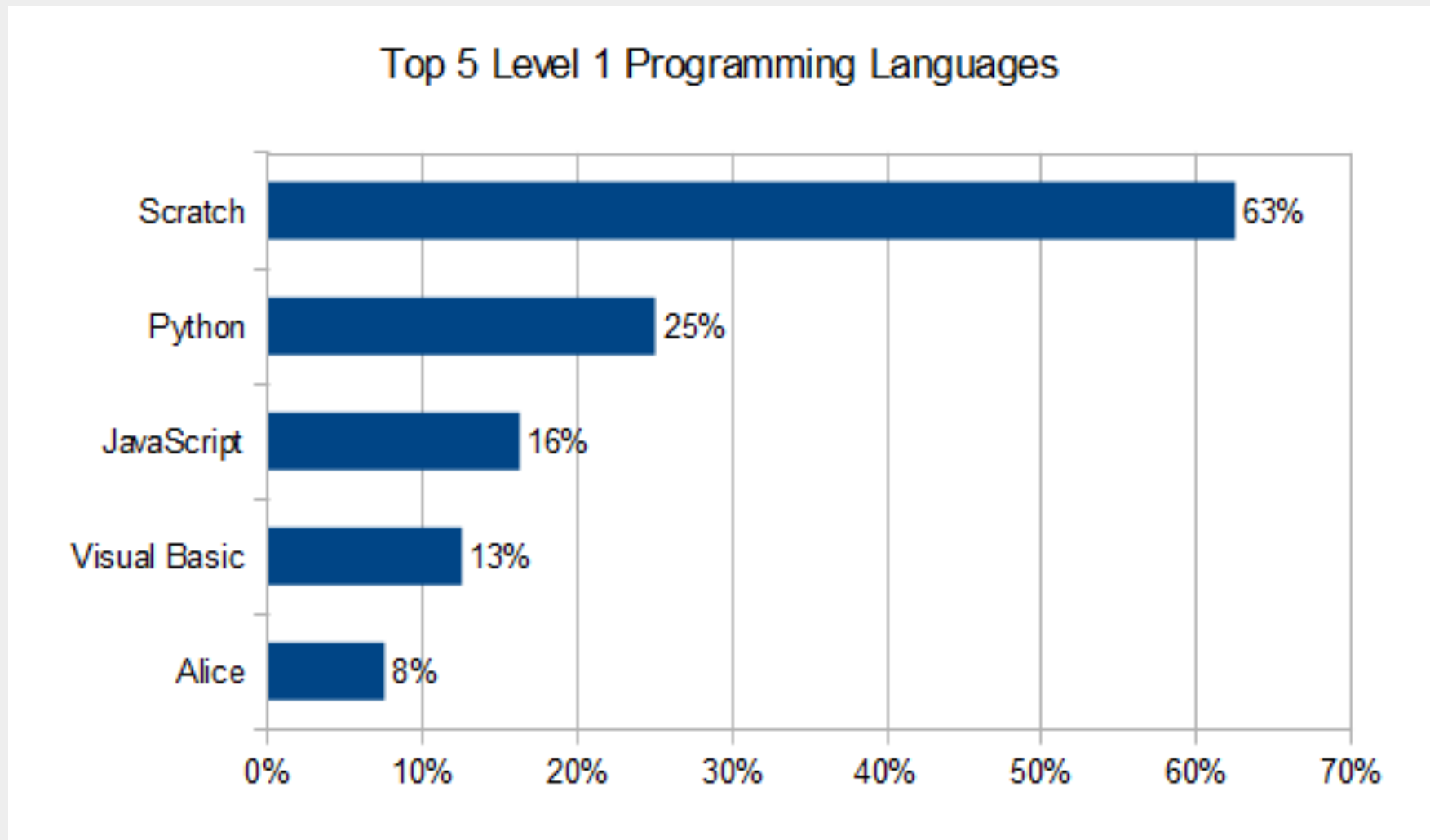
# Motivation for change

Motivations for adopting Programming and Computer Science Standards (2012)



# Programming languages: Level 1

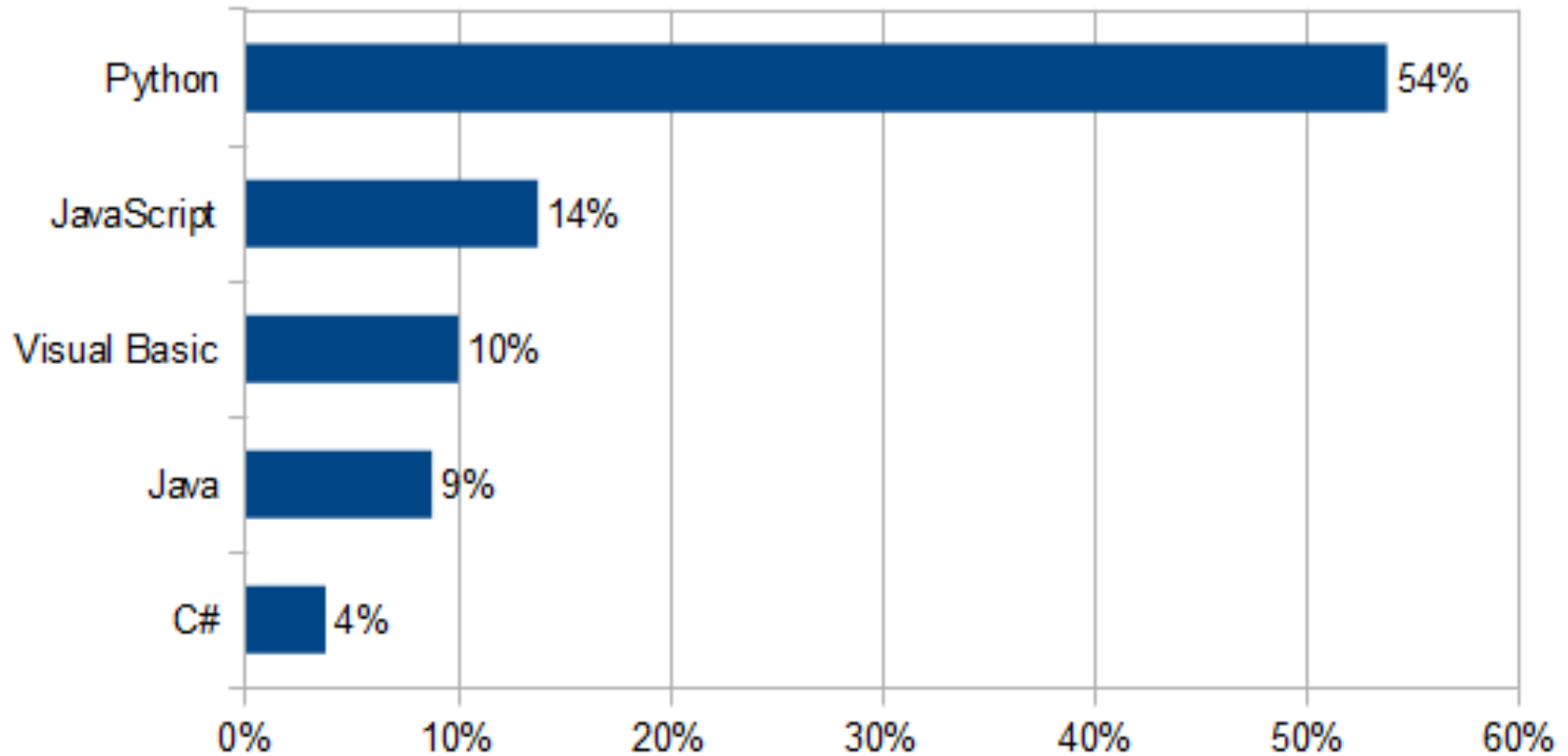
Computer Science in NZ Schools – teacher adoption



# Programming languages: Level 2

Computer Science in NZ Schools – teacher adoption

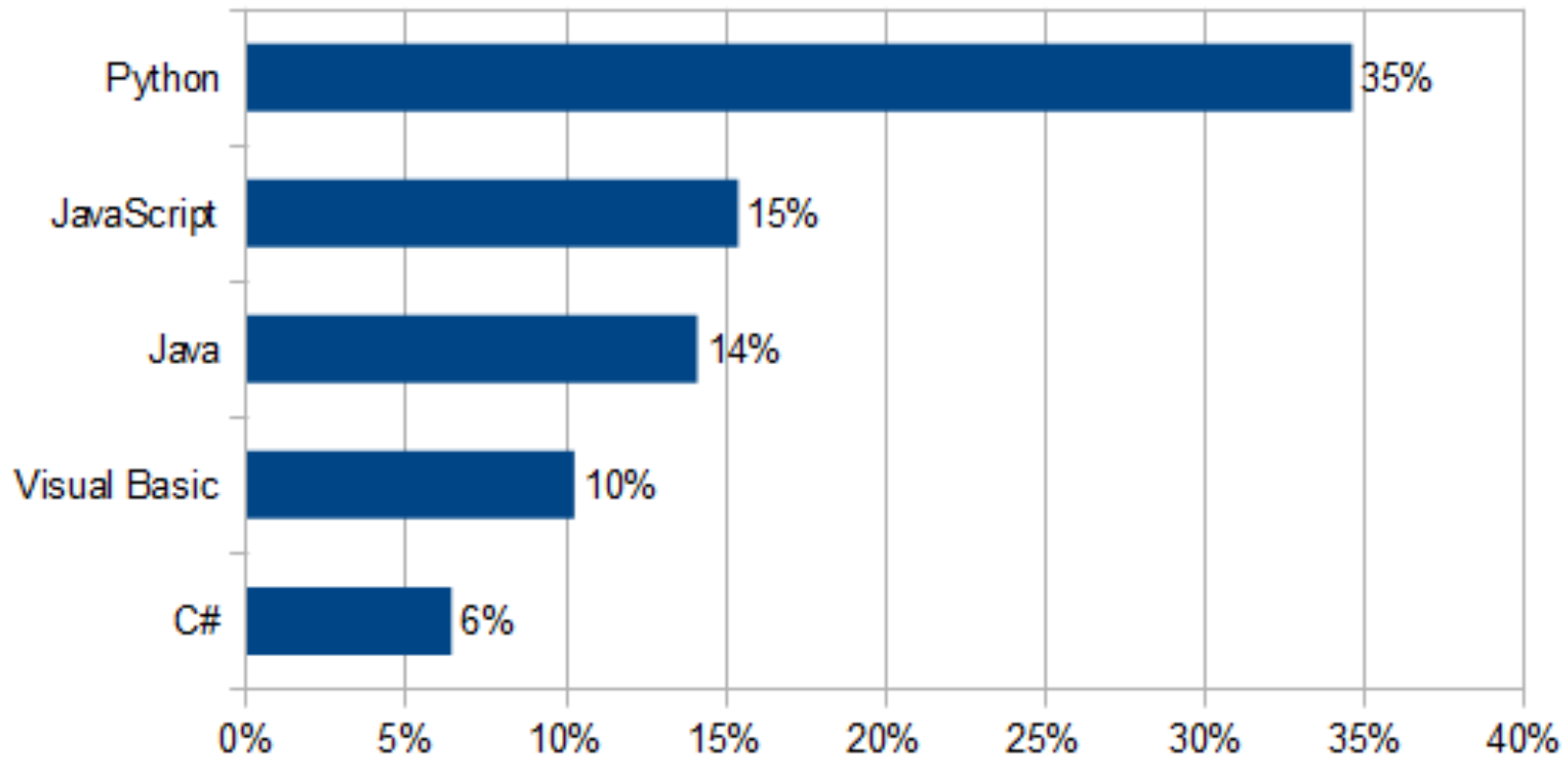
Top 5 Level 2 Programming Languages



# Programming languages: Level 3

Computer Science in NZ Schools – teacher adoption

Top 5 Level 3 Programming Languages





# Themes

- "Hopefully, the work load will ease as courses settle in"
- "I am now into my third year with the [Year 11] students and feel really confident delivering the standards"
- "Educating other staff (still) that this is not a typing class"
- Workload significantly higher than that required for other subjects
- Quality of students attracted

We had this years **DUX** in the subject this year. I have two contenders for next yrs DUX in this years yr12 programming class.... I am finding it a **bit daunting** ... We are growing, and keeping students. They love the new standards. I am now **worried about the lower ability students** ...

25% of my year 13's (65 kids) are going to be studying CS in uni next year. Very exciting.

**Almost worth ALL the work.**

**Tuesday, 12 November 2013 6:24 PM**

**csfieldguide.org.nz**  
**tim.bell@canterbury.ac.nz**