Vanja Venrooy

Unit Title: Computer Science Concepts

Year 11 NCEA Level 1
Digital Achievement Standard: 1.44
External, Credits: 3
No of Lessons: 18-20 plus assessment time

Introduction

This unit plan is designed for a Year 11 class, and aims to cover the basic concepts from Computer Science to ensure students gain an understanding of the three key areas of Algorithms, Programming Languages and Human-Computer Interaction, and to have an appreciation of what the discipline involves.

From the Digital Technology Guidelines

TCKS Objectives

- Demonstrate an understanding of the distinguishing concepts of algorithms and programming languages from Computer Science and Software Engineering
- Understand the concept of an algorithm (vs. a program), and that there are different costs for different algorithms for the same task.
- Understand the Programming Language concepts of high level languages, machine languages, interpretation and compilation, and the idea that programming languages are precise.
- Understand what user interfaces are and the importance of usability and therefore be able to evaluate different interfaces based on different criteria

Key Areas of Knowledge

- Concepts of algorithms, and some common examples
- Compare and contrast different algorithms
- Programming languages and their roles
- Compare and contrast different languages
- High & low level languages
- Machine language, compilers and translators
- The role of Human User Interface
- Evaluating Human User Interfaces
- Compare and contrast different user interfaces

NZ Curriculum

Curriculum Focus
Technological Knowledge – Level 6

- Technological systems – understand the implications of subsystems for the design, development, and maintenance of technological systems.

Nature of Technology

- *Characteristics of technology* – understand the interdisciplinary nature of technology and the implications of this for maximising possibilities through collaborative practice.
- *Characteristics of technological outcomes* – understand that some technological outcomes can be perceived as both product and system. Understand how these outcomes impact on other outcomes and practices and on people's views of themselves and possible futures.

Key Competencies

- Thinking skills – actively seek, use and create knowledge
- Using Language, symbols and text – representing and communicating information and ideas
- Managing self - setting goals, manage projects, working independently
- Relating to others – students interact with others in the classroom (teacher and students) making informed decisions using feedback from teacher and/or peers.
- Participating and contributing – students participate fully in each lesson, sharing knowledge and creating learning opportunities for others.

Assessment opportunities

Students will be assessed using Level 1 Digital Technology Achievement Standard 1.44, External: 3 Credits

Student prior knowledge

Students should be confident computer users, have experience using the internet for research and in using a variety of applications for collecting and displaying data.

Resources required

A computer with overhead projector and sound for teaching use
Computers with internet access for student use
Various applications for collecting and displaying data
Software for YouTube Videos, Java, Flash, Adobe, Word.
Printer

Notes:

1. This unit plan was developed as my assessment task for Senior ICT towards my Graduate Diploma in Teaching (Secondary). Any feedback would be greatly appreciated.
   My contact details are:
   Vanja Venrooy Email: v.venrooy@xtra.co.nz

2. This is a selection of resources I would choose to use for this topic. They are not all "Classroom-Ready", but could be easily adapted for use. They are also not necessarily in the strict order I would use, as I would try to have a mixture of hands-on, computer-based and discussion based activities in each lesson.
   Number of lessons is a guide only. Rough guide I was working on is 1 credit = 10 hours including assessment time, three major topics given equal time.

Lesson Topic	Lesson Outline	Resources / Activities
Algorithms: Overview **2 lessons**	**Teaching Material:** Introduction, Definitions, Specifying an Algorithm Sorting algorithm examples and Analysing algorithms	http://www.youtube.com/watch?v=Zybl598sK24 A fun video where a 19 month old girl tries to sort out her toy boxes in order of the next largest one. Could be used as a good 'Catch' to start a lesson. http://courses.cs.vt.edu/~csonline/Algorithms/Lessons/index.html An entire module of introductory lessons on Algorithms Very good basis for developing ALL lessons for this topic, or could be used by students to work through as-is Includes Review Tests and animated demonstrations for students to work through. I would use it as a teaching basis, and mix it up with some of the other activities here.
	Student work: Create an algorithm for a simple task and carry out basic testing Includes activities that could be adapted for assessment.	These brain-teasers are computer-based activities but could easily be done using the white-board and pen and paper instead of the computer. 5 – 10 minutes for each puzzle, and I would only use one per lesson. Could be printed on cards and handed out at start of lesson. • http://gwydir.demon.co.uk/jo/games/puzzles/goat.htm The Goat, Wolf, Hay puzzle – get each across the river without them eating each other • http://gwydir.demon.co.uk/jo/games/puzzles/bridge.htm The Seven Bridges puzzle – cross each bridge only once • Swap Puzzle - http://www.cs4fn.org/algorithms/swappuzzle/ Fun little animated game. Students can challenge each other to find least number of moves **Non Computer-based activities** – some may be able to be used for assessment 45-60 minutes each, depending on how much write-up students are required to do. I would only choose one of these, and adapt it by creating a worksheet and more structure for student write-up. • Students first think how they would search for a song on an MP3 player, and then they learn about binary search. Zip file containing pdfs: Download here. • Students think of an algorithm to shelve books in a library, and calculate the cost to sort books using the algorithm. Zip file containing pdfs: Download here. • Students work on a Sudoku puzzle and a maze puzzle, and figure out algorithms to solve them. Zip file containing pdfs: Download here. • http://csi.dcs.gla.ac.uk/workshop-view.php?workshopID=12 – close to being classroom ready. Requires registration for CSInside at no cost. A really good 50 minute workshop on developing and testing algorithms. This workshop pack includes Instructions, Power point file, Hints and tips for use, Curriculum Links (to the Scottish curriculum, but could easily be adapted for New Zealand) and the Aims of the workshop. This workshop also contains elements useful for AS 1.45 – writing an algorithm.

Lesson Topic	Lesson Outline	Resources / Activities
		Kinaesthetic Activities Good for a warm-up: • http://www.cs.duke.edu/csed/pltl/exercises/lessons/14/sorting.html Students work on algorithms for making S'mores. Requires crackers, chocolate spread, marshmallows and some plates and cutlery. Good for hands-on, and they get to eat the results! These could take at least 30 minutes each, and I would only choose one of these. • http://www.rigb.org/christmaslectures08/html/activities/get-it-sorted.pdf#page=1 Activities for Bubble and Quick sorting algorithms, using the students to sort themselves by height. • http://mathforum.org/alejandre/frisbie/jam.html Traffic Jam which is a problem solving activity that requires you to develop an algorithm Includes group and individual kinaesthetic activities, as well as computer based activity Also includes **extension** and algebra exercises. This is similar to the Swap puzzle, but more extensive, and covers topics for AS 1.45.
Algorithms: Comparisons **4 lessons**	**Teaching Material:** Comparing speed and space required for different algorithms *Complexity, Infeasibility, Computability of algorithms – **more for Level 2, or extension***	• http://www.youtube.com/watch?v=2HjspVV0jK4 Fun musical introduction into comparing Sorting algorithms based on the song "The Gambler" Could be used as a good 'Catch' to start a lesson. 3 ½ mins • http://www.csi.ucd.ie/staff/jcarthy/home/FirstYear/Comp1001-L13.pdf Good plain English notes on comparing Algorithms i.e. Complexity Theory Includes algorithm performance in terms of time and memory requirements For Level 2 this also includes: Simple explanations of Big-O notation, Infeasible Algorithms and Computability Some material is a bit more advanced than required for Year 11, but can be used as an overview or extension. • http://www.sorting-algorithms.com/ Includes some really good summarising discussion points on sorting, as well as visualisations.
	Student Work: Examples and activities for comparing different sorting algorithms	• http://csta.villanova.edu/CstaRepository/bitstream/2378/337/1/Activity_cardSorting.pdf A good starter! Requires jumbo cards for demonstration purposes. Allow 15 minutes • http://mathsite.math.berkeley.edu/sorting/brick.html. Sorting Bricks. A good site that explains sorting algorithms and also lets students do the sorting with instructions on each step of the algorithm. Includes step-by-step instructions as well as some challenges to be solved. Easily allow 30 minutes

Lesson Topic	Lesson Outline	Resources / Activities
		• http://www.cs.usfca.edu/~galles/visualization/download.html A complete range of visualisations in one package of several sort algorithms. Java must be installed for opening this JAR file. Very clear to see how each algorithm is working. 10 mins • http://www.mundayweb.com/progs/applets/saas/ My favourite visualisation especially for the Algorithm race. You can set the speed and number of records to be sorted. Includes time taken and number of swaps needed. Requires Java. Could use this tool as part of an assessment task where students run a series of tests, and submit a write-up of results, observations, and conclusions. Allow 2 lessons • http://www.math.ucla.edu/~rcompton/musical_sorting_algorithms/musical_sorting_algorithms.html A cool tool for demonstrating Sorting using music as well as visually. I would use this as a 'Catch' to start a lesson, or bring attention back after a work session. 5 minutes only
Programming Languages: Overview	**Teaching Material:** Overview, introduction, history of programming languages	• http://www.teach-ict.com/gcse/software/programming_languages/miniweb/index.htm Mini website covering introduction, types, low-level, assembly, high-level and interpreter topics. Good basic notes to use as basis for teaching.
	Student Work: Research and share with class	• http://james-iry.blogspot.com/2009/05/brief-incomplete-and-mostly-wrong.html Humorous blog on the history of programming languages. Allocate one or two events per student and research the official version, make notes and share with the rest of the class. Approx 20 minutes
Programming Languages: Different types	**Teaching Material:** Programming languages, compilers, interpreters.	• http://cplus.about.com/od/introductiontoprogramming/p/programming.htm What is a programming language? About compilers and interpreters. • http://www.triond.com/users/Sameer+Shrestha 2 articles of interest here: Clear and easy to understand, and includes a good diagram. 1. Compiler vs Interpreter 2. Compilation and Execution.
	Student Work: Following and giving instructions	• http://csunplugged.org/programming-languages Hands-on activity using pen, paper, verbal instructions, focusing on the need for the use of precise language. 20 mins • http://csunplugged.org/sites/default/files/tellAndDraw1.5.pdf This is an engaging extension of the CSUnplugged "Marching Orders" activity. It exposes students to the idea of open-source development, and also programming language design.

Lesson Topic	Lesson Outline	Resources / Activities
Programming Languages: Comparisons	**Teaching Material:** Different types of language Strengths, weaknesses, which language for which task.	• http://programming.dojo.net.nz/welcome/index Good comparisons and full explanation of many different languages. • http://cplus.about.com/od/introductiontoprogramming/a/comparelangs.htm Compares different types of popular languages, giving strengths and weaknesses.
	Student Work: Compare different languages in their ease of readability, size, speed etc	• http://www.teach-ict.com/gcse/software/programming_languages/starters_plenaries/what%20am%20I%20_%20programminglanguage.pdf A poster that can be used as a simple activity to guess which programming language called 'What am I'. 10 mins • http://programming.dojo.net.nz/resources/programming-language-posters/index Posters and information comparing the same FizzBuzz algorithm, programmed in 6 different languages. Python, Ruby, C, Basic, Scheme and Java. Includes ideas of discussion topics, and can be used as a starting point for further research. • http://99-bottles-of-beer.net/ Same program (99 bottles of beer) in Alice, Java, Python, Scratch, QBasic and too many others. Can be used as a comparison activity and assessment task. 2 or 3 lessons
Human Computer Interaction: Overview	**Teaching Material:** Overview, what is good design, bad design	• http://www.adaptivepath.com/blog/2010/03/12/explaining-user-experience-design-to-high-schoolers-and-other-new-audiences/ Good starting place – The User Experience, and what you need for a good experience. • http://www.fatmax.org/AS/419/index.html A great collection of teaching resources here including really good power points on HCI.
	Student Work: Identifying bad design, suggesting improvements	• http://www.baddesigns.com/examples.html#intro http://www.goodexperience.com/tib/archives/product_design/ Lots of bad designs to admire here. Look for suggestions from students at ways to improve the design. • http://csunplugged.org/human-interface-design Classroom ready activity, aimed at raising awareness of human interface design issues. Allow 1 hour or more.
Human Computer Interaction: Principles	**Teaching Material:** Usability principles	• http://www.teach-ict.com/as_a2/topics/user_interfaces/as_userinterface/index.htm Good explanations of several User Interfaces, including Pros and Cons of each. • http://library.gnome.org/devel/hig-book/stable/index.html.en An online text book covering topics in Usability Principles. Easy to understand language, and comprehensive coverage of topics. Use as teaching notes, or online text book for student use.

Lesson Topic	Lesson Outline	Resources / Activities
	Student Work:	• Develop an instruction manual for a simple task on a simple interface, such as a digital watch, video/TV recorder, air conditioning remote control, or cell phone.
Human Computer Interaction: Evaluating	**Teaching Material:** Methods for evaluating interfaces	• http://www.useit.com/papers/heuristic/heuristic_list.html This list is used to evaluate different user interfaces. Heuristic based evaluation is more suited for Level 2, but students can still get an introduction to what they are here.
	Student Work: Evaluating various user interfaces	• http://www.cs4fn.org/fundamentals/hci.php Good magazine-style website, interesting articles and links to activities. These are some good ideas for activities or assessment tasks. • Identify a frustrating user interface, and explain why it is difficult to use. • Compare the interfaces for two different devices, such as two cell phones, two mp3 players, or even two car alarm remote controls. What are the relative merits of the different interfaces for the same task? • Have a parent or grandparent try to perform a simple task on an unfamiliar digital device (such as send a text message, or set the time on a microwave oven clock), noting down every key press they make and noting any problems they have. Write a report on specific problems with the design of the interface for this user.