

Be able to read, understand, write, and debug software programs using an appropriate programming language, tools, and software development process.

- [L6] Be able to read, understand, write, and debug simple software programs
 - Be able to develop a simple program using variables, expressions, selection, and loops.
-

Textbooks

- Wikipedia has a section on 'Educational Programming Language' that classifies programming languages used in education with other real world programming languages. Many educational programming languages position themselves inside a learning path that is a sequence of languages each designed to build on the others moving a student from easy to understand and entertaining environments to full professional environments. Some of the better known are presented in this page
http://en.wikipedia.org/wiki/Educational_programming_language

- An introduction to problem solving and programming is covered in chapters 1 and 2 of the textbook 'AQA Computing' by Kevin Bond and Sylvia Langfield.

This book can be purchased for \$62 from fishpond.co.nz at

http://www.fishpond.co.nz/Books/Computers/General/product_info/12010070/

- 'Hello World, computer programming for kids and other beginners' by Warren and Carter Sande. [Publisher's site](#) also lists reviews on this book. This book uses **Python** as the programming language.

This book can be purchased directly from the author's publisher Manning at <http://www.manning.com/sande/> and it costs \$US 35 for the hard copy (with e-book), and \$22.50 for the e-book only.

- A complete online textbook in **Java** 'Introduction to Programming Using Java', Fifth Edition by [David J. Eck](#) can be downloaded at <http://math.hws.edu/javanotes/index.html>
- A complete online textbook in **Python** 'How to Think Like a Computer Scientist: Learning with Python' by Jeffrey Elkner, Allen B. Downey, and Chris Meyers can be downloaded at <http://openbookproject.net//thinkCSpy/>
- 'Snake Wrangling for Kids – Learning to program with **Python**' by [Jason R Briggs](#) is a free e-book that can be downloaded at <http://www.briggs.net.nz/log/writing/snake-wrangling-for-kids/>

A printed version of this book can be purchased from Lulu Self Publishing at <https://www.lulu.com/commerce/index.php?fBuyContent=6372170> and it costs approximately \$US 10

- Wikibooks has a **Scratch** Open-Content textbook at <http://en.wikibooks.org/wiki/Scratch>

Online Guides

- [Jeannette Wing](#) has an article and presentation on Computational Thinking at locations below:
 1. [Computational Thinking Presentation](#)
 2. [ACM article on Computational Thinking](#)
- [Nishant Gupta](#) has lessons on [Curriki](#) on algorithms that explains the following concepts:
 1. [Lesson 1 - Introduction to Algorithms](#)
 2. [Lesson 2 - Divide and Conquer Method](#)
 3. [Lesson 3 - Greedy Method](#)
 4. [Lesson 4 - Dynamic Programming](#)
 5. [Lesson 5 - Different Types of Sorting](#)
- A collection of everyday algorithms are collated by [Jessen Havill](#) below:
 1. [Car seat installation instructions](#)
 2. [Shampoo instructions](#)
 3. [Washing machine instructions](#)
 4. [Fire alarm instructions](#)
 5. [Fire extinguisher instructions](#)
- A guide to writing **Pseudocode** by [Tim Whitfort](#) can be downloaded here http://ironbark.bendigo.latrobe.edu.au/subjects/PE/2005s1/other_resources/pseudocode_guide.html
- A guide to doing **desk checks** by Tim Whitfort can be downloaded here http://ironbark.bendigo.latrobe.edu.au/subjects/PE/2005s1/other_resources/desk_check_guide.html
- A free online flowcharting tool is at <http://www.drawanywhere.com/> which requires free registration to use

- [RobotProg](http://www.physicsbox.com/indexrobotprogen.html) is a free virtual robot with a flowchart: first you draw the flowchart, and then you run the program and watch the robot executing your program. With RobotProg you can learn programming bases by means of gradual levels. At the last level, you may program several robots playing games on the same ground. For tutorial and downloads visit: <http://www.physicsbox.com/indexrobotprogen.html>
 - [NRICH](#) has activities to test student understanding of flowcharts, programming logic and algorithms at:
 1. [Flow Chart](#)
 2. [Procedure Solver](#)
 3. [Travelling Salesman](#)
 4. [Only Connect](#)
 - A case study [How Do You Introduce Computing in an Engaging Way? Teaching Programming and Language Concepts Using LEGO](#) by [NCWIT](#) gives an insight into using LEGO bricks to teach language specification and other abstract concepts.
 - A tutorial for understanding the Object Oriented Paradigm designed by [developers from University of Saskatchewan](#) is at http://www.cs.usask.ca/content/resources/tutorials/csconcepts/2000_10/w ebTutorial/html/Tut-index.html
 - A complete course in programming (runs in Internet Explorer only) with **VBA** (Visual Basic for Applications) developed by [King George V School](#) at <http://www.kgv.net/ict-ks4/DevProgramming/programming.htm>
- Important: Please be aware the above course is copyrighted material and it is recommended for reference purposes only. You will need to get prior permission from igc@kgv.edu.hk to be permitted to use the materials.
- Tutorials in **VBA** (Visual Basic for Applications) for Excel by [John Lacher and Associates](#) are [VBA Fundamentals Tutorial](#) and [Advanced VBA Tutorial](#)
 - A case study [How Do You Introduce Computing in an Engaging Way? Storytelling](#) by [NCWIT](#) gives an insight into using **Alice** to introduce fundamental concepts in programming, problem solving and logical thinking.
 - A presentation 'Alice: A Free 3D Animation World for Teaching Programming' by [Barb Ericson](#) can be downloaded at <http://acacomputerlab.com/Documents/Alice%20Tutorial.ppt>

- Some materials developed to explain some programming concepts when using **Java** language developed by [Barb Ericson](#) are below:
 1. 'Introduction to Programming' at <http://coweb.cc.gatech.edu/ice-gt/uploads/297/Intro-Prog-Mod1-part2b.ppt>
 2. 'Declaring Variables' at <http://coweb.cc.gatech.edu/ice-gt/uploads/297/Variables-Mod4.ppt>
 3. 'Variables Worksheet' at <http://coweb.cc.gatech.edu/ice-gt/uploads/297/Variable%20Worksheet.doc>
 4. 'Loops Review' at <http://coweb.cc.gatech.edu/ice-gt/uploads/297/LoopsReview.ppt>
 5. 'Teaching Java using Turtles' at <http://coweb.cc.gatech.edu/ice-gt/uploads/297/JavaTurtles-Mod3-part2.ppt>
- [Sun Microsystems](#), creators of **Java** has materials for learning BlueJ, a simpler variation of Java aimed at beginners
 1. [An Introduction to Java](#)
 2. [Java Fundamentals](#)
 3. [Understanding Conditionals](#)
 4. [Understanding Loops](#)
- [Object-Oriented Programming Concepts](#) by [Sun Microsystems](#), creators of **Java** teaches you the core concepts behind object-oriented programming: objects, messages, classes, and inheritance. This lesson ends by showing you how these concepts translate into code.
- [Sun Microsystems](#), creators of **Java** has tutorials on the following programming concepts with regards to Java language:
 1. [Variables](#)
 2. [Operators](#)
 3. [Expressions, Statements, and Blocks](#)
 4. [Control Flow Statements](#)
- Beginner Developer Learning Centre from Microsoft called 'Kid's Corner' is at <http://msdn.microsoft.com/en-gb/beginner/bb308754.aspx> which contains teacher materials, student lessons, quizzes etc.

These are aimed at the beginner and intermediate learners of programming with Microsoft programming environments such as Visual Basic and .NET; it also covers HTML web design and Windows Application programming.

You can also check some other related links at <http://msdn.microsoft.com/en-gb/beginner/cc979165.aspx>

See also Windows Development for Beginners at <http://msdn.microsoft.com/en-us/beginner/dd435692.aspx>

- Follow the Wikiversity course in Introduction to Programming at [http://en.wikiversity.org/wiki/Introduction to Programming](http://en.wikiversity.org/wiki/Introduction_to_Programming)

Some other Wikiversity courses for learning popular languages are below:

1. [http://en.wikiversity.org/wiki/Introduction to Programming in Java](http://en.wikiversity.org/wiki/Introduction_to_Programming_in_Java)
 2. <http://en.wikiversity.org/wiki/Topic:Python>
 3. [http://en.wikiversity.org/wiki/Visual Basic](http://en.wikiversity.org/wiki/Visual_Basic)
 4. [http://en.wikiversity.org/wiki/Learning to program with Alice](http://en.wikiversity.org/wiki/Learning_to_program_with_Alice)
- [Sunraysia Mallee School Network](#) teachers have developed a complete free Moodle course in Scratch with guest access called [Multimedia Programming with Scratch](#) that has teacher resources, videos, student work etc.
 - [Susan Rodger](#) has a growing repository of free **Alice** resources including tutorials, videos and Alice World examples below:
 1. [Alice Tutorials](#)
 2. [Alice Videos](#)
 3. [Alice Example Worlds](#)
 4. [Alice materials](#) for Duke High School Girls Event Materials
 5. [Alice materials](#) for Duke Femmes Event
 - 'Adventures in **Alice** Programming' is a large repository of Alice tutorials submitted by various authors mentioned on the site. View the entire repository at the following locations:
 1. [Tutorials for Grades 5-12](#)
 2. [Alice Examples for Middle School and High School](#)
 3. [Teacher Lesson Plans](#)
 - [Steven Huss-Lederman](#) has several resources in **Alice** that were used in workshops for promoting Computer Science to female students. You can view them at <http://beloit.edu/~huss/BECSMaterials.html>.

For solutions to the exercises and activities in the above resources, you can contact Steven at huss@beloit.edu

- [Computer Studies Educators at Waterloo](#) has a presentation on teaching a course using **Scratch** and **Alice**. [Peter McAsh](#) explains basic concepts of sequence, selection, and repetition in addition to more complex Object Oriented concepts such as methods, objects, functions, parameters, abstraction, inheritance, and event handling.

1. [How to teach a course in Scratch and Alice](#)
 2. Programming Concepts – [presentation](#), [document](#)
 3. [Programming Tasks](#)
 4. [Resource File](#)
- **Scratch:** A step-by-step '[How to get started with Scratch Software](#)'

You can also view the [Scratch support website](#) with many documents, videos and resources to get started with Scratch

- A case study [How Do You Introduce Computing in an Engaging Way? Snap, Create, and Share with Scratch](#) by [NCWIT](#) gives an insight into using **Scratch** to algorithmic thinking.
- Introduction to **Scratch** at <http://coweb.cc.gatech.edu/ice-gt/uploads/629/IntroScratch-short.ppt> by [Michelle Venable-Foster](#) & [Barb Ericson](#)
- [ICT Cover Lessons](#) has a set of tutorial videos (created using [Camtasia](#)) for **Scratch** that teaches how to create parrots and make them fly at random, then shoot them and make them disappear. View at http://www.ictcoverlessons.com/cover_lessons/scratch/scratchhome.html
- Lessons created for Computer Science outreach sessions with presentations and activities in Scratch are shared by [Ben Schafer](#) at locations below:
 1. [Slides](#)
 2. [Introduction Lesson "Plan"](#)
 3. [Irish game](#)
 4. [Zipped directory of all materials suitable for distribution on USB drive](#)
 - 5.
- [Richard Baldwin](#) has well written tutorials for beginners in the following languages:
 1. [The New Face of Computer Science Education - The Scratch Generation](#)
 2. [Introductory Java Tutorial](#)
 3. [Intermediate Java Tutorial](#)
 4. [Advanced Java Tutorial](#)
 5. [Test Your Java Knowledge](#)
 6. [Python Programming Tutorial](#)
 7. [Learn to Program using Alice](#)
 8. [Computer Programming for Homeschool Students and Other Beginners](#)

- Another online textbook on **Greenfoot** is 'Getting Started with the Greenfoot Java IDE' by [Richard Baldwin](#) at <http://www.developer.com/java/other/article.php/3761811/Getting-Started-with-the-Greenfoot-Java-IDE.htm>
- [KIDware Software](#) has a **Visual Basic Express** tutorial CD-ROM that can be purchased online either via download or post at <http://www.kidwaresoftware.com/vbekids.htm>. At the time of writing this, the price was US\$20 (Credit card or [Paypal](#) account required for making payment). There is a free trial version of the tutorial which can be downloaded at <http://www.kidwaresoftware.com/downloads/vbekid10.zip>.

The tutorial consisting of 10 chapters explaining (in simple, easy-to-follow terms) how to build a Visual Basic Express Windows application. Students learn about project design, the Visual Basic Express toolbox, and many elements of the BASIC language. Numerous examples are used to demonstrate every step in the building process.

- A collection of notes and resources developed at [Universiti Malaysia Pahang – Faculty of Civil & Earth Resources](#) with **Visual Basic** in mind are below:
 1. [An Introduction to Programming Languages](#)
 2. [Writing your First Program](#)
 3. [Understanding Programming Concepts](#)
 4. [An Introduction to Visual Basic](#)
 5. [The Importance of Planning your Program](#)
 6. [Programming Tools](#)
- [Nigel Ward](#) has projects in **Visual Basic** with full instructions for the teacher at http://www.morpheus.cc/ict/vb/vb_net/lessons.htm ordered in increasing complexity.
- A tutorial in **Visual Basic** developed by [FunctionX](#) is at <http://www.functionx.com/visualbasic/index.htm>

This web site provides lessons and topics on how to use the Visual Basic language. The lessons teach Visual Basic as a computer language, not as a programming environment. We use our step-by-step approach to learning with practical examples and guided sections.

- A complete tutorial in **Visual Basic** by [Kardi Teknomo](#) is at <http://people.revoledu.com/kardi/tutorial/VB/index.html>

- [KIDware Software](http://www.kidwaresoftware.com/javakids.htm) has a **Java** for Kids tutorial CD-ROM that can be purchased online either via download or post at <http://www.kidwaresoftware.com/javakids.htm>. At the time of writing this, the price was US\$20 (Credit card or [Paypal](#) account required for making payment). There is a free trial version of the tutorial which can be downloaded at <http://www.kidwaresoftware.com/downloads/javak20.zip>.

The tutorial consisting of 10 chapters explaining (in simple, easy-to-follow terms) how to build a Java application. Students learn about project design, object-oriented programming, console applications, graphics applications and many elements of the Java language. Numerous examples are used to demonstrate every step in the building process. The tutorial also includes several detailed computer projects for students to build and try.

-
- [BBC-GCSE Bitesize](#) has resources (revisions and review tests) on the following:
 1. [Logo Control](#), [Test](#)
 2. [System Flow charts](#), [Test](#)

Visual Simulations & Demonstrations

- A nice demonstration of the common programming principles such as [while](#) loops, [if](#) statements, and [arrays](#) all programmed in Java can be seen at <http://www.ics.heacademy.ac.uk/resources/rlos/introprog/examples.php>
- A simple flash animation on creating a class in **Java** can be viewed at <http://nhsweb.calvertnet.k12.md.us/Stroh/JAVA/FlashTut/index.htm>

Classroom Activities & Games

- [CS Unplugged Activity 12](#) – Programming Languages
- [CS Unplugged Activity 10](#) – Routing and Deadlock (teaches algorithms)
- [MaTHmaniaCS Lesson 16](#): Deadlock (teaches algorithm concepts)
- [CS4FN](#) has the following online and offline activities/articles that demonstrates concepts in computational and logical thinking:
 1. [What is computational and logical thinking?](#)
 2. [Kakuro, Logic and Computer Science](#) activity
 3. [Pixels and Puzzles](#) activity
 4. [Sudoku](#) activity
 5. [The Intelligent Piece of Paper](#) activity

6. [Winning at Noughts and Crosses](#) activity
 7. [Create a face](#) activity which also introduces object orientation
 8. [Computer Magician](#), a mind reading activity
 9. [Programming Fundamentals](#)
- [Puzzles, Maths & More](#) site has 2 different puzzles students can try out to improve problem solving skills below:
 1. [Rolocu](#)
 2. [Sudoku 4x4](#)
 - Games and activities to introduce the following basic concepts and introduce logical thinking by [Barbara Ryder](#) are below:
 1. [Boolean Musical Chairs](#) is on introducing the idea of Boolean statements and operators
 2. [Collaborating on code](#): Each group of student write a piece of code. Pieces of code are combined and students analyse the program.
 3. [Illegible code](#): Each student write a code for the function assigned, and then the code is swapped with a neighbour, and discussed.
 - 4.
 - [Susan Rodger](#) has a group activity on [Concept maps](#) at <http://www.cs.duke.edu/csed/ptl/exercises/lessons/82/conceptMap.pdf> which can be used as a precursor to teaching UML later.
 - Students work through logic problems, involving weighing coins, writing logical code in Alice, and Boolean conditions game in an activity designed by [Susan Rodger](#) at <http://www.cs.duke.edu/csed/ptl/exercises/lessons/17/logicThinking.zip>
 - A blog with interesting post and comments on [teaching functional programming to kids](#), and especially using the familiar context of a Dr Seuss story to introduce the idea.
 - An activity suited for VBA(Visual Basic for Applications) in PowerPoint that is aimed at enhancing an interactive presentation by introducing VBA Code into its slides is by Tim Gamble at http://www.teach-ict.com/contributors/tim_gamble/vba_powerpoint/Year%2010%20VBA%20PowerPoint.doc

This excellent instructional booklet shows students how to set up interactive quizzes using PPT and some simple VBA.

- Activities developed by [Paul A. G. Sivilotti](#) to introduce Software Engineering concepts to high school girls are at the following locations:
 1. [Lab Manual on the activities](#)
 2. [Lecture slides supporting the activities with introduction to SE](#)
 3. [Lab 1: Save the Turtle](#)
 4. [Lab 2: Dragons and Butterflies](#)
 5. If you want to view samples of student work from a workshop for the above activities, you can download them [here](#).
- A multitude of lessons and activities in writing algorithms for problem solving in day to day activities by [Ritchie King](#) can be downloaded from http://www.teach-ict.com/contributors/Ritchie_King/Writing_Algorithms.pdf
- Lesson plans for learning to draw flow charts to learn about selection statements and control planning with regard to everyday activities at <http://www.abbotswood.hants.sch.uk/planning/y3control08.htm>.

These lesson plans allows the knowledge to be transferred to flow charts on the computer via a tool called [Flowol](#).

- [Mark Tippins](#) has some resources for Flowol2 below:
 1. Introduction to Flowol2 ([presentation](#))
 2. Introduction to Flowol2 ([handout](#))

The presentation goes through the main Flowol tools and gives an example of how they are used. The help sheet is a reminder for pupils and a simple exercise to complete once they have seen the presentation.

- [Barbara Ryder](#) has the following flow chart activities:
 1. Students play a game where they guess a number between 1 to 10 in 4 or less guesses, and [draw a flowchart of their guessing method](#).
 2. Students create a flowchart describing [what a person does in the morning](#) using the cards given.
 3. Each group of students [create their own flowchart](#) to solve the given problem.
 4. Students create a flowchart to perform [multiplication](#) without access to an explicit multiplication operation.
 5. Students turn the code, which determines how many stones to remove for [NIM 2 game, into a flowchart](#).
 6. Each student play a role in a [scenario of real-life situation](#), and identify the inputs and outputs needed for his/her role.

- [Rebecca Bates](#) has a KLA activity for students to help reinforce the concepts of selection and repetition using flow charts. At the end of the exercise, the teacher can assess student understanding of variable declaration and assignment, selection structures, and loop mechanics. View here at http://people.cs.ubc.ca/~kla/index.php?page=Flow_Chart_Hopscotch
- Activities that are developed for learning flowcharts using **Flowol** can be downloaded below:
 1. [Primary Control with Flowol](#)
 2. [Flowol How to Sheet](#)
- Teaching tips for some of the major control structures in programming from [Institute of Computing Education at Georgia Tech](#)
 1. [Arrays - Teaching Tips](#)
 2. [Conditionals - Teaching Tips](#)
 3. [Loops/Iteration - Teaching Tips](#)
 4. [Variables - Teaching Tips](#)

More tips on [How to Make Better Programming Assignments](#)

- Here is a tried and tested structure for an exercise developed by [Sumant Murugesu](#) that can be administered in class. Students can start with writing or drawing the program logic first and then write the code in a programming language of choice
 1. Introduce students to [Pseudocode](#) if you haven't already done so. A relaxed format could be acceptable for activity purposes as long as it is some form of Structured English. Discuss the need for formalising the spoken language to express ideas better in algorithms. Please be aware that there would be syntactical differences in Pseudocode, but at the end of the day it's only a medium to write algorithms in a structured way.
 2. [Pseudocode](#) task - Students write the logic behind Linear Search in Structured English or Pseudocode

See a sample at Wikipedia http://en.wikipedia.org/wiki/Linear_search
 3. Students can complete a desk check to test their own logic
 4. The Pseudocode can be translated into a [flow chart](#)

5. The Pseudocode can be used to write a program in a programming language
 6. Play the number guessing game to illustrate a Binary search. Get students to write Pseudocode, draw a logic diagram and may be program the code as they did for Linear search.
- Some activities in **Scratch** covering flowcharts, event handling, simple sequential execution, loops, variables, conditionals, parallel execution, and message passing. These were developed by [Barb Ericson](#)
 1. A lesson plan for Scratch at <http://coweb.cc.gatech.edu/ice-gt/uploads/629/LessonPlanForScratch-6-5-09.doc>
 2. A programming activity in Scratch at <http://coweb.cc.gatech.edu/ice-gt/uploads/629/SimpleWitchGame-6-4-2009.ppt>
 3. A programming activity in Scratch at <http://coweb.cc.gatech.edu/ice-gt/uploads/629/SimpleGameBabyCatch-6-26-09.ppt>
 - [Learn Scratch](#) is a website dedicated to teachers and students wanting to learn **Scratch**. It has 3 lesson plans as shown below:
 1. [Getting Started: An Introductory Course](#)
 2. [Step by Step: A Course in Scratch Programming](#)
 3. [Scratch Projects: A Comprehensive Course](#)
 - [Lero](#) has a complete set of teaching materials for **Scratch** (requires registration) which can be downloaded at <http://www.lero.ie/EducationOutreach/Secondlevel/ScratchLessonPlans/CompleteSet.html>
 - [Karen Brennan](#) has a guide for setting up a workshop in **Scratch** that discuss setting goals, meeting one another, introducing Scratch, creating projects, sharing experiences, preparing for next steps at <http://scratched.media.mit.edu/sites/default/files/WorkshopDesignGuide.pdf>
 - [Ralf Romeike](#) explains how **Scratch** creatively introduces computer programming, and provides lesson outlines for initial programming lessons at <http://www.cs.uni-potsdam.de/~romeike/UEWettbewerb/index-english.htm>
 - [Nigel Ward](#) has 10 Scratch lessons which he developed for 15 year olds at <http://morpheus.cc/ict/scratch/default.htm>

- [Redware Research Limited](http://scratch.redware.com/) has a website dedicated to resources in **Scratch** for parents, teachers and computer club facilitators that includes video tutorials too at <http://scratch.redware.com/>
- [Bill Kerr](http://billkerr2.blogspot.com/2008/03/scratch-challenges-introductory.html) has a [blogspot](http://billkerr2.blogspot.com/2008/03/scratch-challenges-introductory.html) with introductory challenges in **Scratch** at <http://billkerr2.blogspot.com/2008/03/scratch-challenges-introductory.html>
- Programming projects in **Scratch** developed by [Nebo Elementary School](http://nebomusic.net/scratch.html) are at <http://nebomusic.net/scratch.html>

See also 'Line Follow Robot Step-by-Step in Scratch' activity at [http://nebomusic.net/Scratch Line Follow.html](http://nebomusic.net/Scratch%20Line%20Follow.html)

- Classroom20 has a Wiki on **Scratch** at <http://wiki.classroom20.com/Scratch>
- [David J. Malan](http://www.davidjmalan.com/) offers a lecture in introductory programming in **Scratch** along with introducing some basic concepts in programming: Pseudocode. Constructs: instructions, variables, conditions, branches, and loops. Languages: interpreted and compiled. Scratch. Available in [Flash](#), [MP3](#) and [QuickTime](#) formats, along with [jargon](#), [slides](#), and [transcript](#) in PDF.

Problem Set 8: Programming. Reinforce your understanding of programming! Available in [PDF](#).

- [PC Pro UK](http://www.pcpro.co.uk/tutorials/350926/how-to-write-your-own-computer-game) has a fun activity in **Scratch** 'How to write your own computer game' at <http://www.pcpro.co.uk/tutorials/350926/how-to-write-your-own-computer-game>
- [PC Pro UK](http://www.pcpro.co.uk/tutorials/350965/create-a-quiz-using-alice) has a fun activity in **Alice** 'Create a Quiz using Alice' at <http://www.pcpro.co.uk/tutorials/350965/create-a-quiz-using-alice>
- [Steven Huss-Lederman](#) has developed activities in **Alice** on the following concepts in programming:
 1. [Creating Storyboards](#)
 2. [Creating a clock in Alice](#)
 3. [Ice Breaker, Listener Talker Style, Algorithms via Making Cookies](#)
- Lesson plans that uses MSWLogo as a programming learning tool can be downloaded at <http://www.abbotswood.hants.sch.uk/planning/y5logoplan05.htm>
- A help sheet for using Logo is at <http://www.kented.org.uk/ngfl/ict/teaching/documents/logo-handout.doc>

- [SwissEduc](#) has a programming environment called **Kara** (requires Java installed), which is a programmable ladybird robot that has been developed to help teach Computer Science concepts such as [finite state machines](#), [turing machines](#), [concurrent programming](#), **Java** programming and also [Lego Mindstorms](#). The variations of this program are given below:
 1. [Kara, programming with state machines](#)
 2. [TuringKara, turing machines illustrated](#)
 3. [JavaKara, programming in Java](#)
 4. [MultiKara, concurrent programming](#)
 5. [LegoKara, Kara for Lego Mindstorms](#)
 - [Slippery Rock University](#) has resources from their high school programming competitions from the past years below. The problems and solutions are available in **BASIC**, C, C++, Pascal, and **Python** at [2001](#), [2002 contest](#), [2002 intramural](#), [2003](#) (includes [solutions](#)), [2004](#), [2005](#) and [2006](#).
 - [American Computer Science League](#) has free sample problems in programming you can download and customise for your needs below.
 1. [Finding the biggest number](#)
 2. [Pentominoes](#)
 3. [Paragraph Analyser](#)
 4. [BinArrays](#) using Binary Trees
 - [American Computer Science League](#) has a complete set of logic questions with solutions you could use as test questions when teaching programming in **BASIC** at [contest 1](#), [contest 2](#), [contest 3](#) and [contest 4](#).
- Note: Even though the above questions were based on BASIC language, you can customise these questions to suit the language of your choice**
- To make learning **Python** easier for beginners, [André Roberge](#) has developed an environment for younger children called [RUR-PLE](#). It is based on a programming a simple robot called Reeborg to move around a grid using simple commands in Python. It also includes the Python's own IDLE and a simplified version of an IDE. Follow steps below to install RUR-PLE:
 1. [Download](#) and install Python versions 2.4 or 2.5 or 2.6
 2. Now [download](#) and install wxPython
 3. Finally [download](#) and install RUR-PLE

Read Learning Python: Child's Play with RUR-PLE: [Documentation](#)

Downloads

Videos

- [CS Unplugged Activity 10](#) – Routing and Deadlock [YouTube](#) video
- [Teaching Kids to Code](#) on YouTube by [Michael Kölling](#)
- View the video lecture on 'Introduction to Programming and Scratch' by [David Malan](#) at <http://academicearth.org/lectures/introduction-to-programming-and-scratch>
- View the video lecture by [Larry Snyder](#) and [Mel Oyler](#) on programming and algorithmic thinking at the following locations:
 1. [Algorithmic Thinking](#)
 2. [Programming Basics](#)
- [Dave Briccetti](#) has a [blogspot](#) at <http://young-programmers.blogspot.com/> demonstrates software he uses to teach programming to kids: Scratch, Alice, Python and Pygame at <http://davebsoft.com/young-programmers/teaching-programming-overview.mov>
- **Scratch:** Several videos on the use of Scratch can be downloaded [here](#)
- [Learn Scratch](#) is a website dedicated to teachers and students wanting to learn **Scratch**. It has 3 video courses and videos from Scratch Day as below:
 1. [Scratch: Getting Started](#)
 2. [Programming in Scratch](#)
 3. [Scratch Projects](#)
 4. [Scratch Day](#)
- **Alice:** A set of 3 video tutorials on starting off with Alice by user [chris101b](#) can be viewed at the following YouTube links:
 1. [Alice Tutorial 1: The Basics](#)
 2. [Alice Tutorial 2: Basics Animation](#)
 3. [Alice Tutorial 3: Complex Coding](#)
- **Java:** A complete set of step-by-step video tutorials into programming in Java by user [thenewboston](#) can be viewed on [YouTube](#)
- **Python:** A complete set of step-by-step video tutorials into programming in Python by user [thenewboston](#) can be viewed on [YouTube](#)

- **Python:** This 24 minute video contains interviews with luminaries from the Python community interspersed with A Python Love Story. It was created for use as an introductory activity in a computer science course using Python. You can view this at <http://video.google.com/videoplay?docid=9023849479319414382>
- **VBA:** YouTube user [ExcelVBAHelp](#) has a good set of introductory tutorials for programming using VBA in Excel. See the complete playlist at <http://www.youtube.com/user/ExcelVBAHelp?view=videos>

DRAFT

Be able to read, understand, write, and debug software programs using an appropriate programming language, tools, and software development process.

- [L6] Be able to read, understand, write, and debug simple software programs
 - Be able to develop programs using meaningful variable names, appropriate layout and comments.
-

Online Guides

- [Computer Programming/Standards and Best Practices](#) by Wikipedia offers some of the standards in computer programming that have been agreed upon as being acceptable.

The following Wikibooks offer design patterns/coding standards in the popular languages **Java** and **Visual Basic**:

1. http://en.wikibooks.org/wiki/Java_Programming/Design_Patterns
 2. http://en.wikibooks.org/wiki/Visual_Basic/Coding_Standards
- Wikipedia offers some good explanation for using Camel Case in writing program code in an Object Oriented programming environment at <http://en.wikipedia.org/wiki/CamelCase>
 - Coding Standards in **Visual Basic** is covered in 'Visual Basic.Net Programming Guidelines' by Marc Truitt for [Johnson County ITS, Kansas](#) at <http://its.jocogov.org/documents/VB%20Dot%20Net%20Coding%20Standards.pdf>
 - 'Java Programming Style Guidelines' at <http://geosoft.no/development/javastyle.html> is a comprehensive guide to laying out code appropriately in Java developed by [Geosoft Norway](#)
 - 'Suggestions for writing Computer Programs' by [Nelson Padua-Perez](#) at <http://www.cs.umd.edu/~nelson/documents/SuggestionsForWritingComputerPrograms.htm>
 - Writing comments in **Java** is covered in Java Notes by [Fred Swartz](#) at <http://www.leepoint.net/notes-java/language/10basics/10comments.html>
 - A brief explanation on variables in **Java** by [Fred Swartz](#) at <http://www.leepoint.net/notes-java/data/variables/40variables.html>

- Wikibooks section on control flow in **Python** has the basic essentials needed to understand documentation and commenting at http://en.wikibooks.org/wiki/Python_Programming/Source_Documentation_and_Comments
- View the 'Style Guide to **Python** Code' by Guido van Rossum & Barry Warsaw at <http://www.python.org/dev/peps/pep-0008/>

DRAFT