|  |  |
| --- | --- |
| **Minor Award Name** | Programming & Design Principles |
| **Minor Award Code** | 5N2927 |
| **Level** | 5 |

**Suggested resources to support delivery:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Theme/Topic** | **Type** | **Relevance** | **Author/Source** | **Web Link** |
| Programming History | Website | This brief overview the history of programming is great as it provides the key points in development with visuals. It can be used as a tool in the teaching of the history. | Visual.ly | <http://visual.ly/history-computer-programming> |
| Algorithms | Online (worksheet) | This website covers a lesson for teaching the very basic of the concept of an algorithm. A link on this page directs the user to a worksheet on ‘planting a seed’. Learners must complete this worksheet by labeling the steps 1 to 7. The worksheet enables learner see something as simple as planting as seed has a series of steps when broken down. | Code.org | <https://code.org/curriculum/course1/6/Teacher> |
| Algorithms, searches and sorts | online | This website covers a range of topics in algorithms, searching and sorting. From the very basic of what an algorithm is to the very complex of performing advanced operations through algorithms. There consists a series of lessons on algorithms, binary searches, linear and selection sorts. Each lesson consists of a task and some begin with an introductory video. | Khan Academy | <https://www.khanacademy.org/computing/computer-science/algorithms> |
| Powerpoint notes | online | These notes provide a very good introduction to the algorithm and flowcharts by introducing the idea of creating an algorithm to complete a basic function: make a cup of tea. | Ghanshyamji | <http://www.slideshare.net/ghanshyamji/algorithms-10437440> |
| Programming Languages | Website | This website contains alot of information (explanations / definitions) regarding various coding languages, their differences and generations. It contains, throughout, a number of quizzes also. | LOC | [www.landofcode.com](http://www.landofcode.com) |
| C Programming | Book | Essential reference material for teaching/learning. The book covers the very basic to the more difficult in the C programming language – from variables to linked lists – its all in here – but it doesn’t go into too much detail for the learner. The layout is excellent as it requires the learner to complete tasks step-by-step. The tasks are well laid out, are in a friendly font and the colour scheme is excellent. The instructions are clear and the material is extremely easy to follow. | Mike McGrath | <http://www.easons.com/p-862971-c.aspx> |
| C Programming (Basics / Essentials) | Online Website | Essential reference material – This provides a thorough overview of the basics of the C programming language. From the basics of variable creation to linked lists. The site is split into a number of structured lessons with easy examples. The lessons cover many basic topics including variable creation, conditional/looping statements, functions and arrays and also the more complicated, such as pointer, structures, recursion, linked lists and compiler/linker messages and how to deal with them. | Cprogramming.com | <http://www.cprogramming.com/tutorial/c-tutorial.html> |
| Programming Concepts | Website | This website contains many different worksheets for student activties to teach them programming ceoncepts through the scratch IDE | Colleen Lewis | <http://scratched.gse.harvard.edu/resources/22-scratch-worksheets> |
| Logical thinking (introduction to coding) | online | This website can be used as as introduction to coding and to get students to think at a basic/logical level. It involves a series of tasks (maze-games) they must accomplish using basic moves and loops. They do this interactively (through drag-and-drop) via a split-screen where they can see the maze and the coding simultaneously. Overall, it aims to introduce learners to basic concepts in programming. | Code.org | <https://studio.code.org/hoc/1> |
| Various (Computer Science) | Video | This collection of over 50 videos details everything from the fundamentals of how the computer work, through simple to sophisticated programming concepts, commenting, memory storage, conditional statements, looping, etc. It also contains assignments that can be done. Programming-wise, it mainly focuses on the Java language but can be used for others also. | [youtube] | <https://www.youtube.com/watch?v=_JG_QGDOOFM&list=PLrC-HcVNfULbGKkhJSgfqvqmaFAZvfHes> |
| Java Programming | Interactive website | This interactive website enables the learner to complete basic tasks and see the output simultaneously. The interactive tutorial covers many fundamental programming conepts such as input/output, variables, conditional statements, loops, arrays, etc | Code Academy | <https://www.codecademy.com/learn> |
| Problem Solving and Programming Concepts (9th edition) | Book | Explanations of various basic concepts in any programming language. In addition, techniques for problem solving, in general, are clearly outlined and can be adapted for teaching purposes. Various student tasks and activities throughout the book. Excellent resource for establishing basic knowledge amongst students | Maureen Sprankle / Jim Hubbard | <http://www.amazon.com/Problem-Solving-Programming-Concepts-Edition/dp/0132492644> |
| Simple Program Design – A step-by-step approach (5th Edition) | Book | Contains many areas of study including: how to design a program, how to write pseudocode and what is it, developing an algorithm , in addition to many programming structures such as arrays (and how to iterate through them), lists etc. It also covers the basic of variables and program modularisation. Book can serve to introduce Object-Oriented design and what it is. Information inside is easy to follow and there exist many tasks/activities. | Lesley Anne Robertson | <http://www.amazon.com/Simple-Program-Design-Approach-Edition/dp/1423901320> |
| Using UML – Software Engineering with Objects and Components (2nd edition) | Book | Contains a range of activites for students. The book includes detailed explanation of all the diagrams used in the UML. Some activities are complex but can be adapted to suit. Many UML examples are based on real-life events and can be easily understood. The book contains a series of self-test questions also to assist in student engagement, etc. | Perdity Stevens (& Rob Dooley) | <http://www.amazon.co.uk/Using-UML-Engineering-Components-Technology/dp/0321269675> |
| Programming Enthusiasts | Website | This website may be of interest to talented programming students. It requires them to register and they can then participate in coding to solve hundreds of mathematical (programming) problems. | Project Euler | <https://projecteuler.net/> |
| Boolean Logic | Website | This resource interacively shows learners how logic gates work and can assist in overall learning of boolean operators | Advanced-ict | <http://www.advanced-ict.info/interactive/circuits.html> |
| Programming Tasks | Website | This website contains multiple programming tasks split into different levels (Beginner, Intermediate and Advanced) | Go-Left-Software | <https://go-left.com/blog/programming/100-little-programming-exercises/> |
| Modularisation | Website | This website details information regarding why to split code and modularise it; the benefits, etc. It contains various tasks also. There exist links to other concepts in programming (algorithms, boolean logic, etc) | BBC | <http://www.bbc.co.uk/education/guides/z9hykqt/revision/1> |
| Modularisation | ppt notes | This collection of handy note describes what modularisation is an gives examples of such. | Nate Ullger | <https://www.hccfl.edu/media/52033/chapter03.ppt> |
| Software Testing | Video | This collection of 24 videos covers software testing comprehensively. From the basics of what software testing is and how it is used to verious Software Life Cycles. It also covers what test cases are and how to compile them. | [youtube] | <https://www.youtube.com/watch?v=TDynSmrzpXw&list=PLDC2A0C8D2EC934C7> |
| Testing | Ebook | This ebooks details alot of information about software testing – what it is and what it involves. | Tutorialspoint.com | <http://actoolkit.unprme.org/wp-content/resourcepdf/software_testing.pdf> |
| Principles of Programming | Ebook | This ebook contains alot of information on many topics in programming. Specifically unit testing, frameworks and coding standards. | Karl Seguin | <http://openmymind.net/FoundationsOfProgramming.pdf> |
| Best practices and standards | Website | This website lists, in detail, alot of the coding standards and practices such as indenting, commenting, variable declaration, string concatenation, etc – how they look and how to use them. | Drupal | <https://www.drupal.org/coding-standards> |
| Coding Standards | Video | This short video explain some of the standards in coding. Can be used as teaching tool or a backup to teaching the standards | Akash Sarswat | <https://www.youtube.com/watch?v=2nPHqIJDEmI> |
| Teamwork | Video | This short video simply shows the benefit of good and bad teamwork. It can be used for discussion session, etc in the class | [youtube] | <https://www.youtube.com/watch?v=fUXdrl9ch_Q&list=PLJtw61qZ4J7vWkGjhJOER4Iyr00tddrmU&index=2> |

**Useful Organisations:**

|  |  |
| --- | --- |
| **Name** | **Contact Information** |
| Irish Computer Society (ICS) | <https://www.ics.ie/ics> |
| Microsoft | <https://www.microsoft.com> |
| Computer World | <http://www.computerworld.com/> |
| School of Computer Technology | <http://www.sct-ireland.com/> |
| Quality and Qualifications Ireland (QQI) | <http://www.qqi.ie> |
| Further Education Support Service (FESS) | <http://www.fess.ie> |

|  |  |
| --- | --- |
| **MOOCs (Massive Online Open Courses)** | |
| Free access to online courses  Search regularly for new courses and new start dates | <https://www.mooc-list.com/>  <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x>  <https://alison.com/courses/Diploma-in-Programming-in-C>  <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/> |

|  |  |
| --- | --- |
| **Minor Award Name** | 5N2927 |
| **Minor Award Code** | Programming and Design Principles |
| **Level** | 5 |

**Suggested resources to support delivery:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Theme/Topic** | **Type** | **Relevance** | **Author/ Source** | **Web Link** |
| Introduction | Website/ e-book | Introduction to programming, concepts, definitions (computer science, data, programming, algorithms, abstraction, problem solving) using a very limited control set for a program that is similar to the Roomba- an automated vacuum cleaner. | Harvey Mudd College | <https://www.cs.hmc.edu/csforall/Introduction/Introduction.html> |
| Beginning to code | Website | Great as an introduction to programming, Scratch is a 4GL language that users can create code by snapping “blocks” together.  They can ‘see’ loops, input, output, processing, math and logical operators, Booleans, numbers and strings. It is easy to use and the concepts can be used in many languages.  The offline version (1.4) can be installed on PCs running Windows or Mac.  The online version (2.0) can be run on any OS with a browser such as Chrome (recommended.  May also be useful for showing sprites as a type of ‘Class’ where new ‘instances’ may be created (cloned- only in Scratch 2.0) based on the ‘blueprint’ for a Sprite. | MIT | <https://scratch.mit.edu/about/> |
| e-book | Tutorials on using Scratch for programming. | Jessica Chiang | <http://www.cs.sun.ac.za/rw146/doc/ScratchAnimation.pdf> |
| Website | May be considered for use as a follow-on from Scratch. Based on concepts similar to Scratch, this online environment assists new users create an Android App. There are three introductory tutorials that can be completed in an hour- provides an “early win” for newcomers to programming. There are additional tutorials online as well as regular MOOCs (see list) that run to show how to develop Android Apps using this tool. | MIT | <http://ai2.appinventor.mit.edu/> |
| Historical Development of programming | Website  (text) | A brief (4-page) summary of the evolution of computer programming, with links to more detailed sources. | Andrew Ferguson | <http://cs.brown.edu/~adf/programming_languages.html> |
| Website (Video) | An in-depth (1.2 hrs) review of the evolution of computer programming. | BBC | <https://youtu.be/3g86LL2fnYQ> |
| Website  (infographic) | Shows the highlights of programming concepts from 1801-2008. | Anon | <http://visual.ly/history-computer-programming> |
| Website  (infographic) | Shows the highlights of programming languages from 1957-2011. | Sebastian Anthony | <http://www.extremetech.com/computing/91572-the-evolution-of-computer-languages-infographic> |
| Website  (text) | Generations of Programming (1GL - 5GL) with links. | Rick Minerich | <http://content.atalasoft.com/h/i/56641168-a-short-history-of-programming-languages> |
| Website | Assemblers, compilers, interpreters and translators. Also links to information on Code Editors, Debugging and more. | BBC Bytesize | <http://www.bbc.co.uk/education/guides/zgmpr82/revision/1> |
| Website | Assembly and Machine Code: Demonstrates the relationship between high level code, assembly and machine code. Sample code provided. | Yassin Hassan | <https://assembly.ynh.io/> |
| Website | Comparison of interpreted and compiled languages | Manikandan10 | <http://www.codeproject.com/Articles/696764/Differences-between-compiled-and-Interpreted-Langu> |
| Software Development Life Cycle | Website | Tutorial which includes an overview of S/W development models and discusses the following models: Waterfall; Iterative; Spiral; V; Big Bang; Agile; RAD; S/W Prototype.  The root site has many other resources. | tutorialspoint | <http://www.tutorialspoint.com/sdlc/index.htm> |
| Differentiate between programming languages by identifying their distinguishing characteristics. | Website | Programming Paradigms: Discussion on the various philosophies used in approaching programming. | Gotesborgs University | <http://www.cse.chalmers.se/~bernardy/pp/Lectures.html> |
| Website | A table outlines several major languages which are commonly used for educational purposes. Each question posed and evaluated is asked specifically within the scope of education and use in the classroom.  The root site has additional resources for a range of programming languages. | Programming Dojo | <http://programming.dojo.net.nz/resources/programming-language-comparison/index> |
| Website | Unique Features of Various Programming Languages | David Foster | <http://dafoster.net/articles/2013/01/29/unique-features-of-various-programming-languages/> |
| Website | Compare and contrast “Hello World” in many languages | Cunningham & Cunningham, Inc. | <http://c2.com/cgi/wiki?HelloWorldInManyProgrammingLanguages> |
| Structured programming and design concepts | Website | PseudoCode: programming language neutral explanation of pseudocode and how it relates to algorithms and problem solving. Included exercises/examples. See also BBC Bytesize. | Wikihow | <http://www.wikihow.com/Write-Pseudocode> |
| Website | Sequencing, selection and iteration. Tutorial. | BBC | <http://www.bbc.co.uk/education/guides/zg46tfr/revision/1> |
| Website | How to create a flowchart. Instructions are good but the LucidChart app is a 30 day trial. Free tool below. | lucidchart | <https://www.lucidchart.com/pages/how-to-make-a-flowchart> |
| Online Tool | Create and save flowcharts in the cloud to your Google Drive, Dropbox, or to your computer. These may also be exported for use offline or for printing. Free! | draw.io | <https://www.draw.io/> |
| Download | Free Cross-platform desktop application to create diagrams such as flowcharts | yWorks | <http://www.yworks.com/en/products_yed_download.html> |
| Understanding algorithms and their applications in solving real‐world problems | Website/ pdf | Lesson plan with resources (50 minutes total)  Introduces the concept of algorithms and uses everyday examples to demonstrate the step-by-step nature of documenting a process to achieve a goal. | Code.org with Thinkersmith | <https://code.org/curriculum/course1/6/Teacher#Review> |
| Website | A variety of useful articles and tutorials which aim to help you with learning the basic concepts of Computer Science through explaining different algorithms for sorting: Bubble, Insertion, Merge, Quick and Selection sort. Example code is based on Java but the concept ideas outlined are readily formatted to notes and can be demonstrated (language neutral) in class as shown in the CS50 examples (see link under General) | University of Waterloo | <http://www.mycstutorials.com/articles/> |
| Website | A higher level Analysis of Algorithms including an assessment of their relevant efficiency or run time using Big-Oh notation | Cal Poly Pomona University | <http://www.cpp.edu/~ftang/courses/CS240/lectures/analysis.htm> |
| Website  (Problem sets) | Challenges in programming. It is possible to use a range of different programming languages (Java, C, C++, Python, Haskell, Ruby, BASIC and many more) to solve the ‘puzzle’. Upon completion, the user can review other solutions to the problem in a variety of different languages. This is also useful to show the structures and similarities/differences between different languages, as well as different approaches to solving the same problem. | projecteuler | <https://projecteuler.net/> |
|  | Website | Algorithmic Thinking/ Computational Thinking | University of London | <http://www.cs4fn.org/computationalthinking/index.php> |
| Testing | Website | A summary of a range of types of software testing | softwaretestinghelp.com | <http://www.softwaretestinghelp.com/types-of-software-testing/> |
| Website | A guide to testing code- the process of defining suitable tests for a program. | tutorialspoint | <http://www.tutorialspoint.com/software_testing/> |
| Website | Coding and Testing: Testers and Programmers Working Together. Examples of how testing is a process rather than a single ‘stage’ in the software development cycle. | Lisa Crispin | <http://www.methodsandtools.com/archive/archive.php?id=88> |
| Scripting and Debug | Downloads | Many IDEs (Integrated Development Environments) will detect and highlight syntax errors, and also allow the user to step through the code, set breakpoints and view the stack- this is very useful when trying to debug code and see what is happening. Here is a limited set of IDEs that have such features (not sorted). There are many many more available.: | Various |  |
| JAVA.  BlueJ is a free Java Development Environment designed for beginners and teaching Java. | University of Kent | <http://www.bluej.org/> |
| JAVA, C, C++, JS. NetBeans IDE lets you quickly and easily develop Java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. The IDE also provides a great set of tools for PHP and C/C++ developers. It is free and open source. | Sun/Oracle | <https://netbeans.org/downloads/index.html> |
| PYTHON.  PyCharm Community Edition: Free. includes support for checking conformance with the PEP-8 standard. Others include WingIDE, Spyder, Eric, DrPython and more. | JetBrains | <https://www.jetbrains.com/pycharm/download/> |
| MANY.  Eclipse: primarily use is for developing Java applications, but it may also be used to develop applications in other programming languages through the use of plugins, including: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. Development environments include the Eclipse Java development tools for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others. | Eclipse Foundation | <https://eclipse.org/downloads/> |
|  | Website: Online Tool | With this tool you can visualise the execution line by line of code, including viewing the data type and creation of functions, global and local variables. Supports Python, Java, JavaScript, TypeScript, and Ruby code execution. Some example code is also available. | Philip Guo | <http://pythontutor.com/>(Links to other visualisers are available from this single link) |
| Data Types | Website | One tab is geared towards C/Java data types, and the other (Actionscript) is similar to Python/ Ruby but the text is adaptable for others. The root site also contains a range of other useful resources. | University of Utah | <http://www.cs.utah.edu/~germain/PPS/Topics/data_types.html> |
| Utilise a selection of modularisation concepts such as functions, procedures, variable scope and parameter passing. | Website | An introduction to the use of procedures/ functions/ subroutines, arguments/ parameters and the scope of variables.  Example code uses Python but may be adapted for other languages. | BBC | <http://www.bbc.co.uk/education/guides/z9hykqt/revision> |
| Team-work and Collaboration | Website | Teamwork and involvement of stakeholders throughout the Software Development Life Cycle is a central part of the Agile Methodology. |  | <http://www.allaboutagile.com/what-is-agile-10-key-principles/> |
| Module Descriptor | Consider cross modular assessment with Communications 5N0690 skills demonstration or portfolio. | QQI | <https://www.earlychildhoodireland.ie/wp-content/uploads/2015/06/5N0690_AwardSpecifications_English.pdf> |
| Comply with an accepted set of coding standards in their use of comments, indentation and variable naming. | Website | You can download the coding standards, coding style guides, code conventions, code guidelines, manuals and references for several general programming languages from here for free:  C/C++ Java C# Delphi/Pascal  PHP ASP Visual Basic/VBS Python  Perl JavaScript Assembly SQL | Various authors | <http://www.sourceformat.com/coding-standard.htm> |
| Website | C Coding Standard adapted from http://www.possibility.com/Cpp/CppCodingStandard.html and NetBSD's style guidelines | Carnegie Mellon University | <https://users.ece.cmu.edu/~eno/coding/CCodingStandard.html> |
| Website | C++ Coding Standard adapted from http://www.possibility.com/Cpp/CppCodingStandard.html | Carnegie Mellon University | <https://users.ece.cmu.edu/~eno/coding/CppCodingStandard.html> |
| Website | C# Coding Standards | ETH Zürich (Swiss Federal Institute of Technology in Zurich) | <http://se.inf.ethz.ch/old/teaching/ss2007/251-0290-00/project/CSharpCodingStandards.pdf> |
| Website | PEP 0008 -- Style Guide for Python Code  Built into the PyCharm IDE | Guido van Rossum | <https://www.python.org/dev/peps/pep-0008/> |
|  | Website tool | With Code Beautifier, you can deobfuscate and reformat your source code written in JavaScript, HTML, CSS, PHP or other programming languages. | codebeautify.org | <http://codebeautify.org/> |
| Tutorial Sites: useful as a source of notes, exercises, links, ideas and other resources. | Website | Learn the following:   * Java * Git * JavaScript * PHP * Python * Ruby   And more. includes notes, examples, exercises and in-browser execution of code. | Codecademy.org | <https://www.codecademy.com/> |
| Website | A site with tutorials (notes), exercises and an interactive shell in the browser for the following languages:   * Python * Java * C * JavaScript * PHP * Shell * C# | learnpython.org | <http://www.learnpython.org/>  (other languages can be linked to from here as well) |
|  | Primarily C, but includes PHP, HTML and JS. Many of the resources can be used with other languages. Innovative, detailed and challenging, this is one of the most highly rated programming MOOCs and runs as a self-paced course each calendar year. | Harvard University | <https://study.cs50.net/>  <http://cs50.tv/2015/fall/>  Also available as a MOOC on EdX. |
| Website | Java. Includes Cross-platform IDE. Greenfoot lets students create ‘actors' that live in ‘worlds' to build simulations, games, and and other visual programs. The interface is a full IDE that allows you to edit source code, compile, and debug. Greenfoot provides a variety of tutorials, community support, and teacher resources. If you want a beginner to learn using a visual interface you should consider giving Greenfoot a test drive. | University of Kent | <http://www.greenfoot.org/doc> |
| Website | Java. Resource CD, teaching strategy and useful links included. BreezyGUI is also available here- it uses a graphical user interface package that allows beginning programmers to  easily create simple graphical user interfaces for their programs (and more). | Mathbits.com | <http://mathbits.com/MathBits/Java/JavaOpenPage.htm> |
| Website | Visual Basic tutorials and resource center. | Dr.Liew Voon Kiong | <http://www.vbtutor.net/> |
| Problem sets | Website | Like Project Euler. Although aimed at Python users, the ideas can be adapted for any language. | n/a | <http://rosalind.info/> |
| Free and open-source books | Website | A collection of books that you may download and share with learners. Covers a range of programming languages and topics | Many | <https://github.com/vhf/free-programming-books/blob/master/free-programming-books.md> |

**Useful Organisations:**

|  |  |
| --- | --- |
| **Name** | **Contact Information** |
| Lero (the Irish software research centre) | <http://www.lero.ie/aboutlero/contactinformation> |
| CoderDojo | <https://coderdojo.com/> |
| Computing at school (see Teaching Resource section on the tabs at the top right hand side of the page) | <http://www.computingatschool.org.uk> |
| BBC Bitesize (Computing Area) | <http://www.bbc.co.uk/education/subjects/zft3d2p> |

|  |  |
| --- | --- |
| **MOOCs (Massive Online Open Courses)** | |
| Free access to online courses  Search regularly for new courses and new start dates  MOOCs are fantastic sources of ideas and demonstrate a range of teaching methods, approaches and assessment ideas, both formative and summative. Some courses also supply books and other resources which they often make freely available to other providers of training and education. | [What is a MOOC?](https://youtu.be/eW3gMGqcZQc%20%20)  MOOC Providers  <http://www.alison.com/>  <http://www.coursera.org/>  <http://www.edx.org>  <http://www.extension.harvard.edu/open-learning-initiative>  <http://www.futurelearn.com>  <http://www.microsoftvirtualacademy.com/>  <http://www.mooc-list.com/>  <http://ocw.mit.edu/>  <http://online.stanford.edu/>  <http://oyc.yale.edu/>  <http://www.oeconsortium.org/>  <http://www.open2study.com>  [http://www.uclaextension.edu/](http://www.uclaextension.edu/pages/search.aspx?c=free+courses)  <http://www.udacity.com>  <http://www.udemy.com/> |