**MODULE SPECIFICATION**

|  |  |
| --- | --- |
| 1. **Programme title** | n/a |
| 1. **Module title** | Software Development 1 |
| 1. **Module code** | CMP020L001 |
| **4. Compulsory for** | MSc Computing  MSc Web Development |
| 1. **Optional for** |  |

|  |
| --- |
| 1. **Module description and context**   Describe the content of, and rationale for, the module. Explain how it fits within the context of the programme as a whole and state the academic and professional benefits to students of studying this module. |
| Software Development 1 introduces students to the fundamental concepts, methodologies, and techniques of software development. Programming is a key component of computer science and is an in-demand skill for the workplace inside and outside of the IT industry. Software Development 1 introduces the fundamental principles of software development, including syntax and semantics, variables and primitive data, expressions and assignment, input-output, conditions, iteration, functions, recursion, and an introduction to algorithms. The module details how to build programs using these techniques and how to apply problem-solving strategies in the design and implementation of simple programs. Students will practise the skills of programming. They will work in a high-level language, using the tools to design, implement, build, execute, and test software applications.  Software Development 1 provides students with core programming competencies. The aim of Software Development 1 is to develop students’ fluency in programming languages and software development. The module will require students to both implement their own programs and trace the behaviour of existing programs. |

|  |
| --- |
| 1. **Module learning outcomes**   List the learning outcomes of the module. Briefly explain how each outcome furthers the programme’s design and aims, as set out in the programme specification and context paper.  **LO1: Design, implement, test, and debug a program that uses each of the following fundamental programming constructs – basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing.**  **LO2: Analyse and explain the behaviour of simple programs involving the fundamental programming constructs – variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion.**  **LO3: Identify the relative strengths and weaknesses among multiple designs or implementations for a problem.**  **LO4: Use a programming language to implement, test, and debug algorithms for solving simple problems.**  These learning outcomes are focused on students producing application code that demonstrates an understanding of software development principles. Computing programmes use these fundamental programming principles as a basis for many further modules within software development. |

|  |
| --- |
| 1. **Planned list of topics, arranged sequentially** |
| Introduction to programming. |
| Basic syntax and semantics of a higher-level language. |
| Variable and primitive data types (numbers, characters, Booleans). |
| Expressions and assignments. |
| Simple I/O including file I/O. |
| Conditional and iterative control structures. |
| Functions and parameter passing. |
| Recursion. |
| Introduction to algorithms. |
| Problem solving with algorithms. |
| Introduction to algorithmic problem-solving strategies. |
| Fundamental design concepts and principles (abstraction, problem decomposition, encapsulation). |

|  |
| --- |
| 1. **Teaching and learning**   Describe how students will be taught and how they will study, providing specific examples of activities. |
| Software Development 1 will be delivered in a blended manner. Lecture content will be made available via videos on Moodle. Several videos will support a unit-week: one for each topic covered. Tutorials will allow students to apply the principles of problem solving and program tracing in a group-working environment using problem-based learning. Practical labs will provide students with hands-on practice of developing software using the principles described in the lectures and tutorials.  In the labs, students will work with a partner (pair programming) to enhance their learning. Students will deliver one part of the assessment in a pair. This is to ensure students undertake the benefits of pair programming. The aim is for students to develop their abilities in collaboration and support.  The material will be delivered in a manner so students can build their confidence in developing increasingly complex software applications. Initial use of simple examples which students follow in order to complete a program step-by-step will provide definite practice and feedback. Later use of complex examples with no starting code will allow students to experience building full applications from a simple specification. |

|  |
| --- |
| 1. **Assessment**   Describe how students will be assessed, providing specific examples of assessment types. Explain how those types are designed to demonstrate achievement of the module learning outcomes listed in section 2. Explain too how they will ensure that all students will graduate with strong prospects for employment or further study. |
| Coursework assessment is by submission of source code and by demonstration of working programs (LO1 and LO4). Students will have the opportunity to undertake a range of problems as part of the coursework, thus recognising the wide range of abilities that a student can have, allowing students that want to push their capabilities to demonstrate their skills, and weaker students to pass by demonstrating the requirements of the learning outcomes. The coursework embeds formative methods to provide feedforward as students build larger solutions to problems, using an iterative approach with smaller submissions leading to the final submission. This emulates how software is delivered in industry – using iterative development cycles.  Tests are used to analyse students’ knowledge and understanding of how programs work by evaluating students’ ability to read code and understand the output (LO2 and LO3). This is a summative assessment. The ability to read code is a fundamental skill necessary for students to succeed as software developers. |

|  |
| --- |
| 1. **Assessment table**   The terminology in this section is taken from Online Marks Entry (OME) and must be retained for consistency. It is very important that these tables are completed accurately, as failure to do so leads to errors and delays when grades are ratified. Any questions should be addressed to the Academic Office.  Please note the following:   * Table 1 should be completed when there is only one component (assessment), weighted at 100%. Table 2 should be completed when there are two or more components (assessments). * The *component type/titles* must be identical to the assessment types listed in section 28 (“Assessment weighting”) of the programme specification. * The *component type/title* must be one of the following: coursework, examination, presentation, in-class test or practical. If the component type is the same as the title, it only need be stated once. (The title has a maximum of 30 characters.) * For each *coursework component*, specify the assessment type: essay, portfolio, report, review etc. * The *percentage* (%**)** indicates the weighting of the component or sub-component. In the example provided in the green cells of table 2, the two sub-components are each weighted at 50%, which combined comprise 50% of the final grade:   Report 50%  + Essay 50%  = Portfolio 100%  Portfolio 50%  + Exam 50%  =Final grade 100%   * For *result type,* if the outcome is expressed as a percentage, insert M for mark. For pass/fail, insert G for grade. * For *final component*, answer “yes” if this is the last component to be submitted for the module, “no” if not. * For *must attempt*, answer “yes” if the student must attempt the component in order to pass the module, “no” if not. * For *must pass*, answer “yes” if the student must pass the component in order to pass the module, “no” if not. * The *sub-component type/title* must be one of the following: coursework, examination, presentation, in-class test or practical. For coursework, specify the assessment type. * Note: if for this module the mark is calculated on the basis of only selected components (eg. the best three of five components), please place a tick mark here: [ ] |

**Table 1 – one component**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component (assessment) Type/Title** | **Coursework Volume or Length/Type of Exam** | **%** | **Result type**  **Mark/**  **Grade**  **M/G** |
|  |  |  |  |

**Table 2 – multiple components**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Component (assessment) Type/Title** | **Coursework Volume or Length/Type of Exam** | **%** | **Result type**  **Mark/**  **Grade**  **M/G** | **Final**  **ComponentY/N** | **Must Attempt Y/N** | **Must Pass**  **Y/N** | **Sub-component type/title** | **Coursework Volume/** | **%** | **Result type**  **Mark/**  **Grade**  **M/G** | **Final Component Y/N** | **Must Attempt Y/N** | **Must Pass Y/N** |
| Coursework (program) |  | 60 | M | Y | Y | Y |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Class Test | Unseen  2 hours | 40 | M | Y | Y | Y |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Reading and resource list**

Please refer to the appropriate section below, depending on whether the programme is a Roehampton one, or delivered by a collaborative partner as a validation arrangement, a “typical” franchise, or through QAHE or Croydon College.

1. **Roehampton programme**

The ‘Resource List’ refers to the University’s online reading list software, which offers real-time information about library holdings at the University and easy access to online resources. A variety of resources can be recommended, e.g. books, journals, audio-visual, and online resources. As confirmed by LTQC in spring 2021, all resources deemed to be ‘Essential’ should be provided in a digital format (e.g. an e-journal article, an e-book or a Copyright-cleared digitisation), to maximise ease of access for all students. These lists are linked to the module’s Moodle site.

The Panel will be asked to look at the Resource List online in advance of the Panel meeting. Please ensure that your list is ‘published’ and that you are logged out of the online resource lists system, then copy the URL and paste it into the box below. If you need help please contact [LibraryEngagement@roehampton.ac.uk](mailto:LibraryEngagement@roehampton.ac.uk)’

|  |
| --- |
| <https://roehampton.rl.talis.com/lists/6CC30CC8-DD02-0B89-9B27-F804AFEF4426.html> |

If you need a brand new module resource list, please contact [LibraryEngagement@roehampton.ac.uk](mailto:LibraryEngagement@roehampton.ac.uk). If you are making substantial changes to an existing module a new “Module Validation” resource list can be created for you.

All lists are created using the online reading list tool and will follow the guidelines set out in the University [Reading List Framework](https://library.roehampton.ac.uk/ld.php?content_id=33292110). Lists should provide a clear week-to-week or topic guide for students about what they should read, when and why. This reading should be directly related to the work they are doing on a module during any one week or over a period of time studying a specific topic, whatever is appropriate to the discipline. Resources should be representative of our student body and as diverse as possible.

Directed reading should be provided in advance of the class and clearly indicate the importance of the items listed by using the following headings (if appropriate for the programme):

* Essential Reading (all to be provided in a digital format).
* Further Reading
* Further Independent Study (for independent study/assignments)

For new modules, it is not necessary at this early stage to fully meet the requirements of the Framework, e.g. by specifying weekly readings. However, in order for the Library to be able to establish any cost implications, it is very important to bookmark the following:

* All ‘Essential’ books
* All ‘Essential’ journals or databases

Resources already held by the Library should be bookmarked, wherever possible. The library will base any purchasing decisions, on the information provided in the Resource lists and in line with the [Collection Development Policy](https://www.roehampton.ac.uk/globalassets/documents/corporate-information/policies/library-collection-development-policy-may-2018.pdf).

1. **Validated programme**

All library services and support are provided by the partner institution.

1. **“Typical” franchise**

The partner may have access to some e-resources through Roehampton Library. However, they do not use UR’s online resource lists software.

Reading list

A variety of resources can be recommended, e.g. books, journals, audio-visual, and online resources. The university aims to provide all resources deemed to be ‘Essential’ in a digital format (e.g. an e-journal article, an e-book or a Copyright-cleared digitisation), to maximise ease of access for all students. Please list your readings in the box below:

|  |
| --- |
|  |

Reading lists should aim to follow the principles set out in the University’s [Reading List Framework,](https://library.roehampton.ac.uk/ld.php?content_id=33292110) (with the exception of the use of the Talis Aspire resource lists software which is not required). Reading lists should provide a clear week-to-week or topic guide for students about what they should read, when and why. This reading should be directly related to the work they are doing on a module during any one week or over a period of time studying a specific topic, whatever is appropriate to the discipline. Resources should be representative of our student body and as diverse as possible.

Directed reading should be provided in advance of the class and clearly indicate the importance of the items listed by using the following headings (if appropriate for the programme):

* Essential Reading (all to be provided in a digital format).
* Further Reading
* Further Independent Study (for independent study/assignments)

Resources that are already available to students should be selected, wherever possible. If you have access to any library resources through the university, you will be made aware of those.

1. **QAHE and Croydon College (full UR students)**

These institutions have access to almost all resources, including online resource lists. Thus, the standard UR template as it appears in section A applies in these cases.