# UNIVERSITY OF NORTHAMPTON

# MODULE SPECIFICATION

This document forms the definitive overview as to the nature and scope of this module and is used in the University’s quality assurance processes. The information in this document cannot be changed without approval (except for the Indicative Content).

[A glossary of key terms is available.](https://www.northampton.ac.uk/ilt/current-projects/defining-contact-time/types-of-student-contact-time/)

|  |  |
| --- | --- |
| **FACULTY** | Faculty of Art, Science & Technology |
| **SUBJECT AREA** | Technology |
| **SUBJECT FIELD** | Computing |
| **MODULE TITLE** | Problem Solving and Programming |

|  |  |
| --- | --- |
| **MODULE CODE** | CSY1020 |
| **LEVEL** | 4 |
| **CREDIT VALUE** | 20 |
| **MODULE LEADER** | Mohammed Bahja |

|  |  |
| --- | --- |
| **DELIVERY MODE(S)** | Standard |
| **DELIVERY LOCATION(S)** | UON |

**PRE-REQUISITES:**

None

**CO-REQUISITES:**

None

**RESTRICTIONS:**

None

**SUPPLEMENTARY REGULATIONS**\*:

This module has no supplementary regulations

**MODULE OVERVIEW:**

This module introduces students to the skills, principles and concepts necessary to solve problems in computing; develop essential skills to enable the solution of these problems with the construction of appropriate algorithms and a computer program; introduce principles underlying the design of a high level programming language (HLPL).

**INDICATIVE CONTENT:**

The following topics will be addressed:

Problem solving (PS)

Documenting the PS process - report writing

Presentation skills - give a presentation

Basic Constructs,

Psuedo-code

Flow Charts

Functions and testing

Graphical interfaces

Introduction to computers, and directories and programming languages.

Data, categories of data, data types, input.

Arithmetic and the Math calls.

Character type, expressions and statements.

Selection (if-else), selection (switch) and menu interfaces using switch statements.

For loop, more on for loop and while loops and the break statement.

Do-while loops, nested loops and random number generation.

Design and style review and compound data types and arrays.

Array processing, unconstrained arrays and program errors and testing review.

Strings, arrays of Instances and parameter Passing mechanisms.

Syntax/semantics and compilation.

Algorithm design (modelling).

Control flow (sequence selection & iteration).

Data structures (stacks/queues/linked lists/arrays/containers/bags etc.).

Data structures (Files/Databases[access/sorting/searching]).

**LEARNING OUTCOMES:**

|  |
| --- |
| **Module Learning Outcome** |
| **On successful completion of the module with detailed guidance students will be able to:** |
| **Subject-Specific Knowledge, Understanding & Application** |
| 1. Apply the principles and practice of analysis and design in the construction of robust, maintainable programs, which satisfy their specifications. |
| 1. Design, write, compile, test and execute straightforward programs using a high level language; |
| 1. Apply a professional approach to design supported by good documentation to the finished programs. |
| 1. Use an appropriate programming language to construct robust, maintainable programs, which satisfy the specified requirements. |
| 1. Develop a range of industrial standard expertise for designing and constructing software to solve real-life problems using programming |
| **Changemaker & Employability Skills** |
| 1. Recognise problems and develop a strategy for problem solving. |

**TYPICAL LEARNING, TEACHING AND ASSESSMENT HOURS (for the module as delivered on-site at the University of Northampton):**

[View this table on how learning, teaching and assessment hours map to the KIS Categories.](https://www.northampton.ac.uk/ilt/current-projects/defining-contact-time/kis-guidance/)

|  |
| --- |
| **Learning and teaching information for this module when delivered off-site by UN partners is available from the partner institution’s NILE site (or equivalent). Any variation in study hours must be approved by the University of Northampton before students are enrolled, ensuring that study hours provision is always appropriate to support student achievement of the module learning outcomes.** |

|  |  |
| --- | --- |
| **Learning, Teaching and Assessment activities** | **Study hours** |
| **Contact hours: (total)**  Comprising face-to-face and online contact hours as follows: | **48** |
| * **Face-to-face (total) -** this may include the following:   Specialist space (e.g. laboratories, studio space)   * Face to face interactive small group session (generic space in groups of approx. 30 e.g. seminars/workshops/tutorials) | 40 |
| * **Online contact hours** **(total)**  (comprising online activities with mediated tutor input) | 8 |
| **Guided independent study hours  (including hours for assessment preparation)** | **152** |
| **Module Total** | **200** |

**ALIGNMENT OF LEARNING OUTCOMES AND ASSESSMENTS:**

**University of Northampton:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessment Activity** | | | **Learning Outcomes** | **Weighting (%)** |
| **Code** | **Assessment Type** | **Assessment Deliverables** |  |  |
| AS1 | Assignment | Problem Solving  2,000 words | a,c,f | 50 |
| PJ1 | Project | Programming  (including 10 minute viva/demonstration)  1,500 words | a,b,c,d,e | 50 |

The assessment items listed above are graded and contribute to the overall module grade (assessment *of* learning). In addition, there are opportunities for formative assessment (assessment *for* learning), which are ungraded, to support students in achieving the module learning outcomes. These are NOT listed.

**APPROVAL/ REVIEW DATES:**

**Version:** 2

**Date of approval:** September 2018

**Date of revision:** October 2020. Restrictions removed, learning outcomes, study hours and assessments updated.