

Module Details	
Module Title	Fundamentals of Programming
Module Code	COS4016-B
Academic Year	2024/5
Credits	20
School	School of Computer Science, AI and Electronics
FHEQ Level	FHEQ Level 4

Contact Hours	
Type	Hours
Online Lecture (Asynchronous)	16
Laboratories	48
Directed Study	136

Availability	
Occurrence	Location / Period
BDA	University of Bradford / Semester 1

Module Aims
<p>Programming underpins a huge proportion of applications and research in Computer Science - quite simply it is the way that we make computer hardware do what we want. Computer programming can change the world, and understanding fundamental concepts forms a first step towards making a positive difference. The creation of computer code begins by first understanding how to think about problems in a logical and methodical manner. Problem solving techniques, and ways of representing solutions, are demonstrated and applied at the beginning of the module and are then built on by carefully introducing several threshold concepts that help crystallise understanding and broaden the horizons for future coders.</p> <p>This module will introduce key concepts, both theoretical and practical, in the design, implementation and testing of computer software. A single primary programming language is used, Java, but the ideas and knowledge students will gain will be presented as transferrable skills that will underpin the learning and use of new languages in the future.</p>

Outline Syllabus
<p>Introduction to problem solving techniques and their part in the software development process. Basic software tools for software development. Data types and basic data structures. The concepts of object, class and interface. Program constructs: sequence, selection and iteration. Testing and debugging programs.</p>

Learning Outcomes	
Outcome Number	Description
01	Apply basic of software construction techniques and demonstrate knowledge of the tools required develop computer program code
02	Apply basic principles of computer programming to common problems; analyse and run computer program code; test and evaluate programs against basic requirements
03	Apply algorithmic problem-solving approaches

### Learning, Teaching and Assessment Strategy

This module aims to introduce fundamental concepts for problem solving, algorithmic thinking and representation, practical programming and wider context through theoretical lecture sessions followed by closely coupled practical lab sessions. Labs will include demonstrations and comprehensive learning materials illustrating how concepts from lectures can be applied using industry-standard software tools, as well as exercises that explore and expand upon these ideas.

Learning materials will be principally delivered and made available to all students through our Canvas virtual learning environment. Other tools, such as Microsoft Teams, can be used to facilitate feedback, answer queries and support independent learning outside of formally timetabled sessions. Formative feedback on both lab exercises and the coursework are primarily provided through timetabled lab sessions where staff are available to review and advise students about their approach and outputs from working on practical tasks.

Students will be assessed through a single large-scale piece of coursework. This coursework has been designed first and foremost to form a core part of the learning experience of the module by providing a highly structured, interactive and engaging framework for students to apply skills developed during the taught part of the module in a step-by-step, task-by-task manner. The assignment provides students with a partially functional interactive software system, including graphical user interface, with key functional elements removed. Students are given a set of tasks, growing in depth and complexity as each task is completed. Every task can be self-evaluated both formally through code analysis and empirically by running the software and testing potential solutions at every stage. Students will be required to submit both their most complete functional version of the software and a short written report in a descriptive technical style describing their solutions to each task.

### Mode of Assessment

Type	Method	Description	Weighting
Summative	Coursework - Written	An exercise in problem solving involving the development of computer software and an accompanying written report of ap	100%

### Reading List

To access the reading list for this module, please visit <https://bradford.rl.talis.com/index.html>

*Please note:  
This module descriptor has been published in advance of the academic year to which it applies. Every effort has been made to ensure that the information is accurate at the time of publication, but minor changes may occur given the interval between publishing and commencement of teaching. Upon commencement of the module, students will receive a handbook with further detail about the module and any changes will be discussed and/or communicated at this point.*

