Prifysgol **Wrecsam**
Wrexham University

Module specification

When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory

Module Code	COM469
Module Title	Introduction to Programming
Level	4
Credit value	20
Faculty	FACE
HECoS Code	100366
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
Stand-alone module aligned to BSc (Hons) Computer Science for QA and assessment	Option
Software Engineering Summer School	Core

Pre-requisites

Breakdown of module hours

Learning and teaching hours	36 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	0 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**36** hrs
Placement / work based learning	0 hrs
Guided independent study	164 hrs
Module duration (total hours)	200 hrs

For office use only	
Initial approval date	19/05/2022
With effect from date	19/05/2022

For office use only	
Date and details of revision	011/03/2025 Addition of module to the Software Engineering Summer School 8/11/2023 Computing revalidation - module update
Version number	3

Module aims

This module aims to introduce the key foundations of programming with a current, object-oriented programming language (indicatively Python). This module will build from the fundamentals to explore key areas of programming logic and problem solving. Throughout this module, the concepts of programming will be tightly linked to the context of developing within an Integrated Development Environment (IDE). Students will demonstrate their understanding with design solutions based upon contextualised problems.

Module Learning Outcomes - at the end of this module, students will be able to:

1	Identify syntax and structure of an industry-standard programming language.
2	Apply programming techniques to solve contextualised problems.
3	Demonstrate design solutions within an Integrated Development Environment.

Assessment

Indicative Assessment Tasks:

This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

This module will indicatively be made of several coursework pieces that build on/focus on individual areas of expertise within programming and IDE activities. This may include smaller sequential activities for students to build up skills and self-efficacy towards the start of the module and finalise with a larger piece that demonstrates their cumulative skill learned throughout.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3	Coursework	100

Derogations

N/A

Learning and Teaching Strategies

Due to the nature of the subject, many of the sessions will represent didactic segments that include demonstration of key concepts that student will be able to watch and/or follow. This content will also be fully represented online through VLE content that will be available to students as they work. Indicatively these could be sections of code/pre-recorded videos depending on what it most appropriate.

Assessment will occur throughout the module to solidify key threshold concepts of programming. Individual briefs will be given to clearly identify areas of focus through each stage, so students are fully aware of their progress throughout.

Indicative Syllabus Outline

The following may change dependant on the relevant programming language:

- Introduction to Python & Anaconda
- Variables & Data Types
- Working with Strings & Numbers
- Getting Input from the user
- Lists & Tuples
- Functions & Returns
- If Statements
- While and For Loops
- Nested Loops
- Reading and Writing to Files
- Introduction to Objects & Classes

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads

Lutz, M. (2013), *Learning Python: Powerful Object-Oriented Programming*. 5th ed. California: O'Reilly Media.

Other indicative reading

Shaw, Z. (2017), *Learn Python 3: The Hard Way.* Boston: Addison-Wesley.
Ferrone, H. (2020), *Learning C# by Developing Games with Unity 2020*. 5th ed. Birmingham: Packt
Publishing.
Kelly, S. (2019), *Python, PyGame, and Raspberry Pi Game Development*. 2nd ed. Niagara Falls: Apress.