

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	GME412
Module Title	Programming Fundamentals
Level	4
Credit value	20
Faculty	FACE
HECoS Code	100956
Cost Code	GAGM

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Game Development	Core
BSc (Hons) Computer Game Development (with Industrial Placement)	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	48 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	48 hrs
Placement / work-based learning	0 hrs
Guided independent study	152 hrs
Module duration (total hours)	200 hrs

For office use only	
Initial approval date	10 th May 2023
With effect from date	September 2023
Date and details of revision	March 24, the new module code GME472 has been assigned to replace COM474 for games programmes
Version number	2



Module aims

This module aims to introduce the fundamentals of programming through a relevant programming language that aligns to contemporary practice. This module will deliver content through the context of student subject areas and explore programming logic and contextual problem solving. Throughout this module, the concepts of programming will be tightly linked to the context of developing within an Integrated Development Environment (IDE).

On completion of this module students should have a clear concept of the role of programming syntax and logic in the context of their chosen subject area. They should be able to utilise fundamental programming techniques and strategies to solve contextual problems or development criteria.

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	Relate programming concepts to the context of subject area.
3	Apply programming techniques within an Integrated Development Environment.
4	Demonstrate design solutions to contextualised problems.

Assessment

Indicative Assessment Tasks:

This module will indicatively be made of several practice-based 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 or pieces that demonstrates their cumulative skill learned throughout.

Indicatively these coursework pieces may assess initial threshold concepts for programming syntax near the start of the module to ensure progress. Later these coursework pieces could represent more complex and time-consuming contextualised problems or development criteria that require critical thought to apply a combination of techniques and strategies.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1, 2, 3, 4	Coursework	100%



Derogations

None

Learning and Teaching Strategies

Due to the nature of the subject, many of the sessions will be didactic and include demonstrations of key concepts for students to be able to watch and/or follow. This content will also be fully represented online through VLE content and will be available to students as they progress with the module. Indicatively these could be sections of code/diagrams/PDF tutorials/pre-recorded videos depending on what is most appropriate.

Assessment will occur throughout the module to solidify key threshold concepts of programming fundamentals. 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

Depending on the relevance to current industry trends a programming language will be chosen that aligns with contemporary practice. The following indicative syllabus outline will apply to many languages and contexts though terms, styles or order of delivery may vary:

- Conditionals & Loops
- Random Numbers
- Arrays & Sorting
- Saving & Loading
- Introduction to Data Structures
- Enumerated Data Types
- Methods & Functions
- Return Types, Parameters & Keywords
- Objects & Instances
- Console Interfaces & Simple Graphics
- Introduction to Source Control

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update. Please *ensure correct referencing format is being followed as per University Harvard Referencing Guidance*.

Essential Reads

McGrath, M. (2020), *C# Programming in Easy Steps*, Second Edition, Warwickshire: In Easy Steps.

Other indicative reading

Dawson, M. (2014), *Beginning C++ Through Game Programming*, Fourth Edition, Boston: Cengage Learning.

Ferrone, H. (2020), *Learning C# by Developing Games with Unity 2020*, Fifth Edition, Birmingham: Packt Publishing.

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

Shaw, Z. (2017), *Learn Python 3: The Hard Way*, Boston: Addison-Wesley.

