

Module Code:	COM537
---------------------	--------

Module Title:	Applied Programming
----------------------	---------------------

Level:	5	Credit Value:	20
---------------	---	----------------------	----

Cost Centre(s):	GACP	<u>JACS3</u> code:	I322
		<u>HECoS</u> code:	100960

Faculty :	Arts, Science and Technology	Module Leader:	John Worden
------------------	------------------------------	-----------------------	-------------

Scheduled learning and teaching hours	30 hrs
Guided independent study	170 hrs
Placement	0 hrs
Module duration (total hours)	200 hrs

Programme(s) in which to be offered (not including exit awards)	Core	Option
BSc (Hons) Computer Science	✓	<input type="checkbox"/>
BSc (Hons) Computer Science (with Industrial Placement)	✓	<input type="checkbox"/>
BSc (Hons) Computing	✓	<input type="checkbox"/>
BSc (Hons) Computing (with Industrial Placement)	✓	<input type="checkbox"/>
BSc (Hons) Computer Networks and Security	✓	<input type="checkbox"/>
BSc (Hons) Computer Networks and Security (with Industrial Placement)	✓	<input type="checkbox"/>

Pre-requisites
None.

Office use only

Initial approval: 30/08/2018

Version no:1

With effect from: 01/09/2018

Date and details of revision: Modification approved by APSC 03/04/19

Version no: 2

Module Aims

The module builds on the foundation of object-oriented design and implementation to provide a deeper understanding by introducing more advanced features of object-orientation, such as inheritance, abstract classes, nested classes, graphical-user interfaces (GUIs), IO (input/output) and exceptions. These allow an application-level view of design and implementation to be explored. Throughout the module, the quality of application design and the need for a professional approach to software development is emphasized. A student who has successfully completed this module will be able to design code and test object-oriented programs from initial specifications. The student will be able to highlight the key steps in a design process where security should be considered and explain how security controls can be implemented to protect systems and information.

Intended Learning Outcomes

Key skills for employability

KS1	Written, oral and media communication skills
KS2	Leadership, team working and networking skills
KS3	Opportunity, creativity and problem solving skills
KS4	Information technology skills and digital literacy
KS5	Information management skills
KS6	Research skills
KS7	Intercultural and sustainability skills
KS8	Career management skills
KS9	Learning to learn (managing personal and professional development, self-management)
KS10	Numeracy

At the end of this module, students will be able to

Key Skills

At the end of this module, students will be able to		Key Skills	
1	Demonstrate an understanding of current theoretical and methodological approaches to constructing a computer program using a modern development environment	KS3	KS4
		KS6	KS9
		KS10	
2	Implement program designs in an object-oriented programming language	KS3	KS4
		KS9	KS10
3	Select a suitable ADT in order to implement a solution for a specified problem	KS3	
4	Apply standard algorithmic problem solving techniques	KS3	KS4
		KS9	KS10
5	Demonstrate an understanding of the principles of defensible programming approaches to ensure that software being developed securely.	KS4	KS10
		KS5	
		KS6	

Transferable skills and other attributes

Derogations

None.

Assessment:

Indicative Assessment Tasks:

This module has one piece of assessed coursework involving report & implementation.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)	Duration (if exam)	Word count (or equivalent if appropriate)
1	1, 2, 3, 4, 5	Coursework	100%	2 hours	4000

Learning and Teaching Strategies:

This module has an emphasis in the practical issues related to Applied Programming and it will be delivered using a combination of formal lecturers, tutorials, practical demonstrations and lab sessions. Links to freely available software and further reading material (scientific papers, tutorials, exercises, etc.) will be uploaded to the university VLE.

Syllabus outline:

- Concepts and fundamentals of programming styles: procedural, object-oriented, functional. Benefits of the object-oriented development approach.
- OOP Fundamentals; Structure of an object-oriented program. Classes: attributes, methods. Objects.
- Object-Oriented Design Concepts including modularity, data abstraction, reusability, encapsulation, inheritance and polymorphism, simple patterns.
- Abstract Data Types; Stacks; Queues; Heaps and hash tables
- Problem solving techniques and algorithm design for sorting and searching
- Program testing and debugging techniques

Indicative Bibliography:**Essential reading**

None.

Other indicative reading

Deitel, H. (2016), *C# 6 for Programmers*. 6th ed. Pearson Professional.