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	COM439
Module Title	Problem Solving with Programming
Level	4
Credit value	20
Faculty	FAST
HECoS Code	100956
Cost Code	GACP

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 Science	Core
BSc (Hons) Computing	Core
BSc (Hons) Computer Networks and Security	Core
BSc (Hons) Cyber Security	Core
BEng Electrical and Electronic Engineering	Core
BEng Electrical and Electronic Engineering with Industrial Placement	Core
MEng Electrical and Electronic Engineering	Core

Pre-requisites

None

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

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	30/08/2018
With effect from date	01/09/2018
Date and details of revision	July 2022 Template update, programme list update in Engineering revalidation
Version number	3

Module aims

This module will introduce students to the key concepts of software design and development.

It will take a systematic approach to problem solving, and will use design methods to enable students to construct programmed solutions. A modern, object-oriented computer programming language will be used in a hands-on laboratory setting, where students will work through a number of exercises to develop the fundamental skills to prepare them for more complex software engineering practice at a higher level.

This module aims to:

- Use logical thinking and algorithmic techniques to enable students to solve procedural problems.
- Provide students with knowledge and skills to use notations and tools to articulate problem solutions in the form of program designs.
- Give students a clear understanding of the software development process, including analysis, design, implementation and testing.
- Introduce a modern object-oriented programming language, giving students a clear understanding of the syntax and structure of that language.
- Give students a clear understanding of the object-oriented programming paradigm.
- Introduce students to an Integrated Development Environment to support the production of object-oriented applications.

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

1	Interpret problem specifications, and translate them into logical, designed solutions.
2	Use program designs to develop working computer programs.
3	Demonstrate an understanding of object-oriented programming.

4	Use an Integrated Development Environment (IDE) to build graphical user interfaces.
5	Understand the key stages of software development and their relationship to the discipline of Software Engineering.

In addition to the module learning outcomes, engineering students will also cover the following accreditation of higher education programme (AHEP) fourth edition learning outcomes: C1.

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.

The assessment will comprise of two pieces of coursework, comprising of exercises and/or larger programs, program design, program listings and evidence of testing will be the main components of the assessments.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3,4,5,	Coursework	50
2	1,2,3,4,5,	Coursework	50

Derogations

None for Computing programmes

For BEng/MEng Electrical and Electronic Engineering, derogation from regulations has been approved for this module which means that whilst the pass mark is 40% overall, each element of assessment (where there is more than one assessment) requires a minimum mark of 30%.

Learning and Teaching Strategies

The module will be delivered through a combination of formal lectures, tutorials and labs. Students will have access to lecture materials and ancillary resources, via the University's VLE platform.

Indicative Syllabus Outline

1. Problem solving techniques and logical thinking.

2. Program design tools.

3. Programming rules.

4. Program constructs (sequence, selection, iteration).

5. Subprograms.

6. Data structures.

7. Object-oriented programming techniques.

8. Graphical user interface programming with an IDE.

9. The practice of software engineering.

Indicative Bibliography:

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

Essential Reads

None

Other indicative reading

Stroustrup, B. (2013) The C++ Programming Language. 4th ed. Upper Saddle River, NJ: Pearson Addison Wesley

Stroustrup, B. (2014), Programming: Principles and Practice Using C++. 2nd ed. Addison Wesley.

Picking, R. (2007), Get on up with Java. Colchester: Le xden Publishing.

C/C++ Language and Standard Libraries

https://msdn.microsoft.com/en-us/library/hh875057.aspx

The Java Language Specification

https://docs.oracle.com/javase/specs/jls/se7/html/index.html

Employability skills – the Glyndŵr Graduate

Each module and programme is designed to cover core Glyndŵr Graduate Attributes with the aim that each Graduate will leave Glyndŵr having achieved key employability skills as part of their study. The following attributes will be covered within this module either through the content or as part of the assessment. The programme is designed to cover all attributes and each module may cover different areas.

Core Attributes
Engaged
Creative

Key Attitudes
Commitment
Curiosity
Confidence

Adaptability

Practical Skillsets
Digital Fluency
Organisation
Critical Thinking
Communication