MODULE SPECIFICATION

Part 1: Information			
Module Title	Introduction to OO Systems Development		
Module Code	UFCFC3-30-1	Level	Level 4
For implementation from	2020-21		
UWE Credit Rating	30	ECTS Credit Rating	15
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Module type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description

Educational Aims: See Learning Outcomes.

In addition to the educational experience set out in Learning Outcomes, this module will explore, develop, and practise:

Working in small groups and presenting work as a team.
The ability to complete problem solving tasks

Outline Syllabus: The syllabus will include the following topics:

Software development lifecycle
Software development methods (e.g. prototyping)
Problem solving & design with pseudo code (thinking algorithmically)
Problem solving & Intro to OOA&D with the UML class diagram
Introduction to a Java IDE(e.g. Netbeans)

A basic introduction to Object Oriented Paradigm including:
computer architecture overview
source code, byte code, machine code, compilers, interpreters

the role of the JVM
coding style guidelines
primitives, classes & objects
scope of variables
Iteration & Selection statements
arrays & collection classes
file I/O
interfaces
inheritance (& overriding)
GUIs (Netbeans GUI designer)
Deploying java applications (.jar files)
Testing & Use of IDE (Netbeans) debugger

Teaching and Learning Methods: The module is delivered through a combination of formally scheduled sessions and independent learning. The scheduled learning includes lectures, tutorials, demonstrations and practical classes/workshops.

The lecture session will be exploring OO software development theory and demonstrating good practice. These sessions will be responsive to feedback from tutorial sessions.

Practical/Tutorial sessions will concentrate on problem solving and developing/supporting learning of and practice of required skills – use of IDE, development tools (UML, pseudocode), testing and debugging. The tutors will also help to create an environment where students can develop their interpersonal skills, team working skills, and prepare themselves to work with and motivate other people in a professional manner.

The lecture and practical sessions will be closely integrated with each delivery mode informing the other.

In addition students will pursue directed independent learning. This will include time spent reading and absorbing the set text, completing practical exercises, case study preparation, assignment preparation and exam revision. The students will also work through a series of software problems which they will be able to self-assess using software tools. The formative feedback from the tool will help the students monitor their own progress.

Three hours of weekly contact time will be divided between lecture and practical/tutorial sessions as appropriate.

Activity (hrs)
Contact time (72)
Assimilation and development of knowledge including completing formative assessment exercises (153)
Exam preparation (55)
Coursework preparation (20)
Total study time (300)

Part 3: Assessment

The assessment will consist of:

1. A series of in-class tests resulting in a portfolio of programming exercises.

2. An individual coursework assignment of problem solving and implementation. Students will be required to go through the full development cycle - given a problem specification they should demonstrate skills in solution formulation using appropriate techniques (pseudocode/UML) and implementation (computer based). Assessment is by an e-portfolio submission.

Students will have the opportunity for formative feedback during practical lab/tutorial sessions.

The objective here is to encourage and enable students to confidently solve OO problems in a supportive atmosphere.			
First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component A		50 %	A portfolio of unseen, in-class programming exercises.
Project - Component B	✓	50 %	An individual coursework software development assignment
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component A		50 %	Submission of individual portfolio of programming exercises.
Project - Component B	✓	50 %	Design and implementation a software system. Submitted as a report with supporting software. Assessment is by an e-portfolio submission

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will achieve the following learning outcomes:

Module Learning Outcomes	**Reference**
Demonstrate knowledge of the object oriented (OO) paradigm by producing software solutions to simple problems.	MO1
Solve simple problems using OO techniques and express the solutions algorithmically	MO2
Design an OO system using a design notation that has been explored during the module.	MO3
Implement and test an simple OO software system using a suitable Integrated Development Environment (IDE).	MO4
Locate and utilise on-line resources (e.g. as JAVA API) to support self-learning.	MO5

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	228
Total Independent Study Hours:	228
Scheduled Learning and Teaching Hours:	
Face-to-face learning	72
Total Scheduled Learning and Teaching Hours:	72

	Hours to be allocated	300
	Allocated Hours	300
Reading List	*The reading list for this module can be accessed via the following link:* https://uwe.rl.talis.com/modules/ufcfc3-30-1.html	

Part 5: Contributes Towards
This module contributes towards the following programmes of study: Software Engineering for Business [Sep][FT][Frenchay][3yrs] BSc (Hons) 2020-21 Software Engineering for Business [Sep][SW][Frenchay][4yrs] BSc (Hons) 2020-21 Computer Security and Forensics [Feb][FT][GCET][4yrs] BSc (Hons) 2019-20 Computer Security and Forensics [Oct][FT][GCET][4yrs] BSc (Hons) 2019-20 Software Engineering for Business {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2019-20 Software Engineering for Business {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2019-20 Business Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2019-20 Business Computing {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2019-20 Business Computing {Foundation} [Feb][FT][GCET][4yrs] BSc (Hons) 2019-20 Business Computing {Foundation} [Oct][FT][GCET][4yrs] BSc (Hons) 2019-20