

H06OOP: Object Oriented Programming

Module Code:	H06OOP
Long Title	Object Oriented Programming APPROVED
Title	Object Oriented Programming
Module Level:	LEVEL 6
EQF Level:	5
EHEA Level:	Short Cycle
Credits:	5
Module Coordinator:	FRANCES SHERIDAN
Module Author:	FRANCES SHERIDAN
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	MSc Degree in Computing or cognate discipline, or the equivalent experience in industry as programmer.
Learning Outcomes	
<i>On successful completion of this module the learner will be able to:</i>	
#	Learning Outcome Description
LO1	Apply theoretical concepts to a range of contexts and problem domains
LO2	Formulate computer program solutions to well defined abstract problems
LO3	Design, develop, test and debug moderately complex object-oriented programs
LO4	Identify and discuss object-oriented design principles and design patterns in a given object-oriented design.
Dependencies	
Module Recommendations	
No recommendations listed	
Co-requisite Modules	
No Co-requisite modules listed	
Entry requirements	Learners should have attained the knowledge, skills and competence gained from stage 1 of the BSc (Hons) in Computing.

H06OOP: Object Oriented Programming

Module Content & Assessment			
Indicative Content			
Inheritance • Introduction to Inheritance • The role of reuse and inheritance			
Polymorphism • How to utilize polymorphic constructs in programming			
Polymorphism and Inheritance • Use of support libraries from external sources • Revision of Polymorphism and Inheritance			
Array Lists • Arrays of Objects • Introduction to Java Collection Framework Using ArrayLists in Java • Two-Dimensional Arrays			
File Input and Output • File and FileDialog Object. • Low-Level File I/O. • High-Level File I/O. • Creating and Handling Exceptions. • Object I/O.			
Principles of good Object Oriented Design • Object Programming Principles • Object Oriented Programming Principles i.e. SOLID, GRASP			
Design Patterns • What are Design Patterns? An Introduction to Design Patterns – object-oriented software design patterns • Creational Design Patterns (e.g. Singleton Pattern) • Structural Design Patterns (e.g. Decorator Pattern) • Behavioural Design Patterns (e.g. Observer Pattern)			
GUI Objects and Event Driven Programming • GUI Objects • Positioning GUI Objects • Event Handling • Handling Multiple Events			
Regular Expressions • Introduction to Regular Expressions • Developing programs for data processing activities (e.g., data extraction, cleaning, merging, aggregation, analysis, reporting) using regular expressions			
Software Testing • The importance of testing • Methods of testing • Writing a Unit Test • Preconditions and post conditions • Black Box and White Box			
OOP Programming other than Java Java vs Programming Language			
File I/O & Exception Handling • File and FileDialog Object • Low-Level File I/O • High-Level File I/O • Creating and Handling Exceptions • Object I/O			
Data Connectivity • Database Programming - CRUD			
OOP Programming in another programming language • Java vs Another Programming language			
Assessment Breakdown			%
Coursework			100.00%
Assessments			
Full Time			
Coursework			
Assessment Type:	Continuous Assessment	% of total:	40
Assessment Date:	n/a	Outcome addressed:	1,2,3,4
Non-Marked:	No		
Assessment Description: Each week student will submit program code to the Moodle server for grading. Student will be supplied with an interface specification for the program(s) and the grading will be conducted via automated unit testing based on unknown inputs. Students will be examined on their ability to convey understanding of the programs which they have developed.			
Assessment Type:	Project	% of total:	10
Assessment Date:	n/a	Outcome addressed:	1,2,3,4
Non-Marked:	No		
Assessment Description: Students will work in groups to create an OOP programming in another language other than Java.			
Assessment Type:	Practical	% of total:	50
Assessment Date:	n/a	Outcome addressed:	1,2,3,4
Non-Marked:	No		
Assessment Description: The students will have to develop solutions to programming problems relevant to all material covered in the module using a proctored computer in an examination environment. There will be a written component to assess the student ability to determine errors in a program.			
No End of Module Assessment			
No Workplace Assessment			
Reassessment Requirement			
Coursework Only <i>This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.</i>			
Reassessment Description This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination. Project that meets all learning outcomes is provided to the student.			

H06OOP: Object Oriented Programming

Module Workload				
Module Target Workload Hours 0 Hours				
Workload: Full Time				
Workload Type	Workload Description	Hours	Frequency	Average Weekly Learner Workload
Lecture	No Description	24	Every Week	24.00
Lab	No Description	24	Every Week	24.00
Independent Learning	No Description	77	Every Week	77.00
Total Weekly Contact Hours				48.00
Workload: Part Time				
Workload Type	Workload Description	Hours	Frequency	Average Weekly Learner Workload
Lecture	No Description	24	Every Week	24.00
Lab	No Description	12	Every Week	12.00
Independent Learning	No Description	89	Every Week	89.00
Total Weekly Contact Hours				36.00

Module Resources	
<i>Recommended Book Resources</i>	
Vaskaran Sarcar. (2016), Interactive Object Oriented Programming in Java, Apress, p.211, [ISBN: 978-1-4842-2543-1].	
<i>Supplementary Book Resources</i>	
Herbert Schildt. (2017), Java: The Complete Reference, Tenth Edition, McGraw-Hill Education, p.1344, [ISBN: 978-1259589331].	
<i>This module does not have any article/paper resources</i>	
<i>This module does not have any other resources</i>	
Discussion Note:	