



College of Information Technology
Department of Information Technology
COURSE SYLLABUS/ SPECIFICATION

Course Code & Title: ITCS 201 – Object Oriented Programming I
Weight: (2-2-3)
Prerequisite: ITCS 122
NQF Level Allocated: 6

NQF Notional Hours / Credits: 120 notional hours/ 12 NQF credit

Description: This course emphasizes on object oriented programming techniques using Java. It covers the implementation of object oriented concepts, such as: classes, objects, inheritance and polymorphism.

Objectives:

1. To develop an understanding of the principles underpinning object oriented programming.
2. To designate the important features of an object oriented programming language.
3. To understand how to design and implement object oriented concepts and software.

Semester:

Instructor:

Office Telephone:

Email (s):

Intended Learning Outcomes (ILOs):

A. Knowledge and Understanding		NQF Descriptor/ Level
A1	Concepts and Theories: Demonstrate detailed knowledge and understanding of the concepts involved in Java programming for implementing object oriented software.	Knowledge: theoretical understanding [Level 6]
A2	Contemporary Trends, Problems and Research:	N/A
A3	Professional Responsibility:	N/A
B. Subject-specific Skills		NQF Descriptor/ Level
B1	Problem Solving: Solve defined and some undefined problems by writing Java programming code.	Knowledge: Practical Application [Level 6] Skills: Communication, ICT & Numeracy [Level 6]
B2	Modeling and Design: Design the prototype for solving different kinds of real world problems, by clearly stating the concepts involved; such as Constructors, Abstract class, Inheritance, Polymorphism, Interface and their respective set of Properties and Methods using algorithms and UML models.	Knowledge: Practical Application [Level 6]
B3	Application of Methods and Tools: Apply Java programming language constructs and tools to write, run, trace, and debug object oriented programs.	Knowledge: Practical Application [Level 6] Skills: Communication, ICT & Numeracy [Level 6]
C. Critical-Thinking Skills		NQF Descriptor/ Level
C1	Analytic skills: Evaluate and find the best Java object oriented concepts needed to develop efficient and effective programs.	Generic Problem Solving & Analytical skills [Level 6]
C2	Synthetic: Integrate different application objects and object oriented concepts into complete computer applications needed to solve real world problems.	Generic Problem Solving & Analytical skills [Level 6]

C3	Creative Thinking and innovation:	N/A
D. General and Transferable Skills (other skills relevant to employability and personal development)		NQF Descriptor/ Level
D1	Communication: Demonstrate the ability to present Java programs clearly in a well-structured manner.	Communication, ICT and Numeracy Skills [Level 6]
D2	Teamwork and Leadership:	N/A
D3	Organizational and Developmental Skills: Demonstrate ability to organize ideas and effectively allocate time in given assignments.	Competence: Autonomy, Responsibility and Context [Level 6]
D4	Ethics and Social Responsibility:	N/A

Course Structure (Outline)

Course Structure (Outline)						
Week	Hours		ILOs	Unit/Module or Topic Title	Teaching Method	Assessment Method
	Lab	Lect.				
1	2	2	A1	Introduction to object oriented programming language.	Lecture	
2	2	2	A1, B2, B3	Introduction to Classes, Objects, Methods and Instance Variables, Default, Public, and Private Access Modifiers, UML Notations.	Lectures/ lab demonstration	In-Lab Exercises
3	2	2	B3	Declaring a Class and Instantiating an Object, Setter, Getter, and Operational Methods.	Lecture/ lab demonstration	In-Lab Exercises
4	2	2	A1, B3, D1, D3	Constructor, Default, No-Argument and Arguments Constructors, Initializing Objects with Constructors.	Lecture/ Lab demonstration / In-Lab Supervised Work	Assignment 1
5	2	2	A1, B1, B3, C1	Overloading Methods, this Keyword, Encapsulation and Data Hiding.	Lecture/ Lab demonstration / In-Lab Supervised Work	In-lab Exercises
6	2	2	A1, B1, B3, C1	Static Variables, Static Methods, Static Class Members, Static Import.	Lecture/ Lab demonstration / In-Lab Supervised Work	In-lab Exercises
7	2	2	A1, B1, B2, B3, C1	Enumerations, Garbage Collection and Method finalize, Final Instance Variables, Creating Packages, Package Access, and UML Package Notations.	Lecture/ Lab demonstration / In-Lab Supervised Work	In-lab Exercises
8	2	2	A1, B1, B2, B3, C1	Association, Aggregation Relationships and UML Notation.	Lecture/ In-Lab Supervised Work	Lab Test 1

9	2	2	A1, B1, B2, B3, C1, C2, D1, D3	Inheritance and UML Notation.	Lecture/ Lab demonstration / In-Lab Supervised Work	Assignment 2
10	2	2	A1, B1, B3	Protected Access Modifier, Software Engineering with Inheritance, Object Class and super Keyword.	Lecture/ Lab demonstration / In-Lab Supervised Work	In-Lab Exercises
11	2	2	A1, B1, B3, C1	Polymorphism and Overriding Methods.	Lecture/ Lab demonstration / In-Lab Supervised Work	In-Lab Exercises
12	2	2	A1, B1, B2, B3, C1, C2, D1, D3	Static and Dynamic Binding.	Lecture/ Lab demonstration / In-Lab Supervised Work	Assignment 3
13	2	2	A1, B1, B2, B3, C1, C2	Abstract Class, Abstract Method and UML Abstract Notation.	Lecture/ In-Lab Supervised Work	Lab Test 2
14	2	2	B1, B3	Final Methods and Classes.	Lectures/ lab demonstration	In-Lab Exercises
15	2	2	A1, B1, B2, B3, C1, C2, D1, D3	Interfaces, Common Interfaces of the Java API and UML Interfaces Notation.	lab demonstration/ In-Lab Supervised Work	Assignment 4
16	2	-	A1, B1, B2, C1	All Topics		Final Exam

Teaching Materials:

Textbook(s):	Deitel T. R. Nieto. (2017) <i>Java How to Program</i> , 11 th Edition, Prentice Hall.
Handout(s):	- Internal handouts (Hardcopies) prepared by course instructors. - Available on Moodle i.e. http://www.ahlia.edu.bh/moodle
Reference(s):	<ol style="list-style-type: none"> 1. White B. (2018) <i>Mastering Java: An Effective Project Based Approach including Web Development, Data Structures, GUI Programming and Object Oriented Programming</i>, independently published. 2. Gaddis T. and Muganda G. (2018) <i>Starting Out with Java, From Control Structures through Data Structures</i>, 4th Edition, Pearson Education. 3. Liang Y. D. (2017) <i>Introduction to Java Programming Comprehensive Version</i>, 11th Edition Prentice Hall. 4. Barnes D. J. and Kölling M. (2016) <i>Objects First with Java: A Practical Introduction Using BlueJ</i>, 6th Edition, Prentice Hall.

Assessments:

Type of Assessment	Description	ILOs	Weighting
Assignments	Each group of students is required to prepare four assignments; each assignment worth 20%, the average will be taken. Assignments include both theoretical and practical questions to develop an object oriented program for solving a problem.	A1, B1, B2, B3, C1, C2, D1, D3	20%
Lab Tests	Two practical tests will be for two hours and worth 20% each, The tests used to assess students in the implementation of object-oriented programs using Java.	B1, B3, C1, C2	40%
Final Exam	Final exam will be for two hours and including all types of question: problem solving, MCQs and T/F, short answers, programming.	A1, B1, B2, C1	40%
In-Lab Exercises	The students will practice through a number of exercises to design and implement the object oriented programming concepts.	B1, B2, B3	Formative
Overall			100%

Admissions	
Minimum number of students	5
Maximum number of students	20

Ahlia University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.ahlia.edu.bh/integrity for more information).