جامعة الزيتونـــة الأردنيـة
Al-Zaytoonah University of Jordan
كلية العلوم و تكنولوجيا المعلومات
Faculty of Science and Information Technology

" عراقة وجودة"
"Tradition and Quality"

QF01/0408-4.0E	**Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Computer Science Department**

Study plan No.	**2021-2020**		**University Specialization**		**Computer science**	
Course No.	**0112220**		**Course name**		**Object Oriented Programming**	
Credit Hours	**3 Hours**		**Prerequisite Co-requisite**		**Programming Principles**	
Course type	□ **MANDATORY UNIVERSITY REQUIREMENT**	□ **UNIVERSITY ELECTIVE REQUIREMENTS**	□ **FACULTY MANDATORY REQUIREMENT**	□ **Support course family requirements**	□ **Mandatory requirements**	□ **Elective requirements**
Teaching style	□ **Full online learning**		□ **Blended learning**		□ **Traditional learning**	
Teaching model	□ **2Synchronous: 1asynchronous**		□ **2 face to face : 1synchronous**		□ **3 Traditional**	

Faculty member and study divisions information (to be filled in each semester by the subject instructor)

Name	**Academic rank**	**Office No.**	**Phone No.**	**E-mail**	
To be filled by the instructor					
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**
To be filled by the instructor					

Brief description

In this course, You will be to complete the journey of learning Java by learning the concepts of object-oriented programming starting with learning classes and objects and use UML graphical notation to describe class and object (Chapter 9) , discovering the relationship between classes (Association , aggregation and Composition) (Chapter 10) , Inheritance and Polymorphism (chapter 11) , abstract classes and interfaces (Chapter 13) and Recusrsion with examples (chapter 20)

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	1. Y. Daniel Liang: Introduction to Java Programming, Eleventh Edition, Global Version. 2017 2. D. S Malik: Java Programming from problem analysis to program design, 5th Edition 2011.			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	1. Java™ How to Program, 9th Edition, By H. M. Deitel. & P. J. Deitel , Prentice Hall, 2012.			
Supporting websites	https://www.w3schools.com/java/java_intro.asp			
The physical environment for teaching	□ Class room	□ labs	□ Virtual educational platform	□ Others

Necessary equipment and software	**Net Beans 8.2** (https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html)
Supporting people with special needs	------------------
For technical support	------------------

Course learning outcomes (S*= Skills, C= Competences K= Knowledge,)*

No.	**Course learning outcomes**	**The associated program learning output code**
	Knowledge	
K1	Describe objects and classes.	**MK2**
K2	Discovering the relationship between classes.	**MK2**
K3	Discovering the relationship between classes (Inheritance and Polymorphism)	**MK2**
K4	Designing and use abstract classes and interfaces	**MK2**
K5	Explain Recursion	**MK2**
	Skills	
S1	Using UML notation to describe classes	**MS2**
S2	Define, design and develop classes and create objects	**MS2**
S3	Determine the relationship between classes (Association , Aggregation and Composition)	**MS2**
S4	Define a subclass from a superclass using inheritance	**MS2**
S5	Distinguish differences between overriding and overloading	**MS2**
S6	Discovering polymorphism and dynamic binding	**MS2**
S7	Design and use abstract class and Interface and distinguish differences between them	**MS2**
S8	explaining the difference between method and recursion method and Application of recursions	**MS2**
	Competences	
C1	Ability to describe classes and design them	**MC3**
C2	Ability to create relationships between classes	**MC3**
C3	Ability to apply the inheritance and polymorphism between classes	**MC3**
C4	Ability to create an abstract classes and interfaces	**MC3**
C5	Ability to write recursion methods	**MC3**

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	**Fully electronic learning**	**Blended learning**	**Traditional Learning (Theory Learning)**	**Traditional Learning (Practical Learning)**
First exam	0	**0**	%20	**0**
Second / midterm exam	%30	**%30**	%20	**30%**
Participation / practical	0	**0**	10	**30%**

applications				
Asynchronous interactive activities	%30	**%30**	0	**0**
final exam	%40	**%40**	%50	**40%**

Note: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Schedule of simultaneous / face-to-face encounters and their topics

Week	Subject	learning style*	Reference **
1	Review Methods with Example Two-Dimensional Array with Examples Two-Dimensional with examples by passing an Array to the methods	**Lectures**	-
2	Objects and Classes 9.1 Introduction 9.2 Defining Classes for Objects 9.3 Example: Defining Classes and Creating Objects 9.4 Constructing Objects Using Constructors 9.5 Accessing Objects via Reference Variables 9.6 Using Classes from the Java Library	**Lectures**	Textbook1 Pages: 346 - 360
3	9.7 Static Variables, Constants, and Methods 9.8 Visibility Modifiers 9.9 Data Field Encapsulation 9.10 Passing Objects to Methods 9.11 Array of Objects 9.12 Immutable Objects and Classes 9.13 The Scope of Variables 9.14 The this Reference	**Lectures**	Textbook1 Pages: 361 - 388
4	Object-Oriented Thinking 10.1 Introduction 10.2 Class Abstraction and Encapsulation 10.3 Thinking in Objects 10.4 Class Relationships	**Lectures**	Textbook1 Pages: 389 - 397
5	10.7 Processing Primitive Data Type Values as Objects 10.8 Automatic Conversion between Primitive Types and Wrapper Class Types	**Lectures**	Textbook1 Pages:

	10.9 The BigInteger and BigDecimal Classes 10.10 The String Class 10.11 The StringBuilder and StringBuffer Classes		404 – 432
6	Inheritance and Polymorphism 11.1 Introduction 11.2 Superclasses and Subclasses 11.3 Using the super Keyword 11.4 Overriding Methods	**Lectures**	Textbook1 Pages: 420 - 451
7	11.5 Overriding vs. Overloading 11.6 The Object Class and Its toString() Method 11.7 Polymorphism 11.8 Dynamic Binding 11.9 Casting Objects and the instanceof Operator	**Lectures**	Textbook1 Pages: 420 - 451
8	11.10 The Object's equals Method 11.11 The ArrayList Class 11.12 Useful Methods for Lists **MIDTERM EXAM**	**Lectures**	Textbook1 Pages: 455 - 467
9	11.13 Case Study: A Custom Stack Class 11.14 The protected Data and Methods 11.15 Preventing Extending and Overriding	**Lectures**	Textbook1 Pages: 463 - 473
10	Revision, Examples and Assignments Homework discussion	**Lectures**	
11	Abstract Classes and Interfaces 13.1 Introduction 13.2 Abstract Classes	**Lectures**	Textbook1 Pages: 521 - 526
12	13.3 Case Study: the Abstract Number Class 13.4 Case Study: Calendar and Gregorian Calendar 13.5 Interfaces	**Lectures**	Textbook1 Pages: 527 - 534
13	13.6 The Comparable Interface 13.7 The Cloneable Interface 13.8 Interfaces vs. Abstract Classes 13.9 Case Study: The Rational Class	**Lectures**	Textbook1 Pages: 535 - 564
14	Recursion 18.1 Introduction 18.2 Case Study: Computing Factorials 18.3 Case Study: Computing Fibonacci Numbers 18.4 Problem Solving Using Recursion	**Lectures**	Textbook1 Pages: 742- 748
15	• Programming Examples. • Project. • Revision.	**Lectures**	

16	Final Exam		

* Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.

** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.

Schedule of asynchronous interactive activities (in the case of e-learning and blended learning)

Week	Task / activity	Reference	Expected results
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			