



COURSE SYLLABUS

I- Course Information :

Title of the course	Object Oriented Programming
Department	Data Science
Specialization	Data Science
Academic Year	2020-2021
Semester Duration	4 months
Credit Number	
Number of Hours	45 hours
Course Instructor	Mira Abboud
Email	mira.abboud@ul.edu.lb
Office Hours	2 hours / week

II- Course Description

This course emphasizes current techniques in object oriented design and programming. Such topics as encapsulation, information hiding, inheritance and polymorphism are studied in class with assignments and programming exercises. The course focuses on OOP (Object-Oriented programming) concepts to help students design and implement significant projects. Java programming language is utilized to apply (implement) the object-Oriented concepts.

III- Course Outline and Calendar

Learning Objectives	Session(s)	Learning Content	Pedagogical Strategy& Tools
Introduction to OOP concept	Session 1	Course syllabus + Introduction to OOP concept	Interactive course (chapter 1)
OOP basics	Session 2	Classes and Objects (attributes, methods), instantiation	Interactive course (chapter 2) + JAVA Implementation
	Sessions 3-4	Constructors, Encapsulation, modifiers, setters/ getters / toString	Interactive course (chapter 3) + JAVA Implementation
	Session 5	Static fields, methods, delegation (aggregation / composition)	Interactive course (chapter 4) + JAVA Implementation



		Assign graded homework 1 . Deadline: before session 9	
Inheritance	Session 6	Inheritance – part 1: inheritance concept, extends, overriding	Interactive course (chapter 5 : part 1, 2, 3) + JAVA implementation
	Session 7	Inheritance – part 2: reuse of constructors, constructors chaining, inheritance default constructor, inheritance shadowed variables	
	Session 8	Inheritance – part 3: inheritance visibility, inheritance classes visibility, final, object class	
	Session 9	Assignment 1 – corrected	Interactive correction
		Assign projects on different topics. Deadline: to be delivered before session 29	
Polymorphism / abstraction	Session 10	Polymorphism (upcasting/downcasting, dynamic binding, equals function)	Interactive course (chapter 6) + JAVA implementation
	Sessions 11-12	Abstraction and Interfaces (abstract, abstract methods, interface, implements)	Interactive course (chapter 7) + JAVA implementation
	Session 13	Supplementary exercise (overall revision)	JAVA implementation (class work, interactive with students)
Collections	Sessions 14-15	Collections – part 1 : collection interface, iterator, Set interface, HashSet, LinkedHashSet, List interface, ArrayLists, Vector, LinkedList, sort, comparator, comparable, compareTo	Interactive course (chapter 8 : part 1) + JAVA implementation
		Assign graded homework 2 (research) . Deadline: before session 22	
	Session 16	Supplementary exercise on collections	JAVA implementation
	Session 17-18	Collections – part 2 : Maps, Map Interface, Entry Interface, HashMap, LinkedHashMap	Interactive course (chapter 8 : part 2) + JAVA implementation
Exceptions	Sessions 19-21	Exceptions (try-catch block, finally, built-in Exceptions, throw, throws, exception handling, user-defined exceptions)	Interactive course (chapter 9) + JAVA implementations (built-in and user-defined exceptions)
	Session 22	Assignment 2 – corrected	Interactive correction
Input/Output	Sessions 23-25	Input/Output (Streams, binary I/O classes, InputStream/OutputStream, FileInputStream/FileOutputStream, DataInputStream/DataOutputStream,	Interactive course (chapter 10) + JAVA implementation



		endOfFile, Buffer, Object I/O, Serializable Interface, Text I/O)	
Advanced topics:	Sessions 26-27	Generic types	Interactive course (chapter 11) + JAVA implementation
Generic / Linked List	Session 28	Introduction to Linked Lists	Interactive course (chapter 12) + JAVA implementation
	Sessions 29-30	Projects presentation + revision	Revision

IV- Teaching approach & methods

- Most of the sessions will follow the Interactive approach
- Practice (implemented examples) will be a key factor
- Homeworks + research are included

V- Assessment Methods

The assessment of the student is carried out according to the following grid:

Assessment Type	Final Grade Weighting	%	Dates/Deadlines
Class participation	/15	40% of the final grade	Continuous + session 13
Assignment 1	/25		Session 6 – session 9
Assignment 2 (Research)	/20		Session 15 – session 22
Project	/40		Session 9 – session 29
Final exam	/100	60% of the final grade	Date: End of the course
Total		100%	

VI- Specific Course Requirements

1- Regulations concerning deadlines for written and applied work submissions

The teacher has the right to:

- Refuse to receive written applied work outside the deadlines.
- Refuse homework/project copies with high percentage of similarities.
- Refuse homework/project not submitted on MS teams.
- Give the student additional time after omitting up to 25% of the mark awarded to the concerned activity.

2- Academic measures in case of plagiarism

The student shall be liable to the following penalties in case of any attempt of illicit scientific citation:

- Cancellation of the student's contribution, activity or scientific article and reduction of the final mark.



- If the same offense is reported in other courses, cancellation of the final exam and failure in the relevant course(s).

VII- Reference Books

Liang, Y. Daniel. "Introduction to Java Programming 8-th edition." (2011).

VIII- Bibliography & Additional Sources

Lewis, John, and William Loftus. "Java software solutions." *Boston: Addison Wesley Longman. MASIE Center E-learning CONSORTIUM (2003). Making sense of learning standards and specifications, retrieved November 8 (2003): 2004.*