Object-Oriented programming
Course Specifications

Program(s) on which the course is given : Bachelor in Computer & Information Sciences

Major or Minor element of programs : all majors

Department offering the program : **Computer Science**

Department offering the course : Computer Science

Academic year / Level : 2nd Year /B.Sc.

Date of specification approval : 7/3/2010

A. Basic Information

Title: Object-Oriented Programming   **Code:** CSW 251

Lecture: 3 hrs/week   **Practical:** 3 hrs/week   **Tutorial:** ---

Credit Hours: ---   **Total:** 3 hrs/Week

B. Professional Information

1. **Overall Aims of Course:**

The aims of this module are: give students a good understanding of basic concepts and terminology related to the Object-Oriented Paradigm; introduce them to the fundamental principles of abstraction, modularity and reusability; and to develop the programming skills and

experience needed to design and write object oriented programs within the Java language.

CSW 251 aimed at students who wish to learn more about basic Object Oriented Programming (OOP) concepts, including objects, classes, methods, parameter passing, information hiding, inheritance and polymorphism. Those concepts are introduced along with their implementations using Java. The course also covers developing graphical user interface (GUI). A number of more advanced Java topics will also be discussed. These may include: Java core classes, exception handling, Threads, and Streams.

2. **Intended Learning Outcomes of Course (ILOs):**

a. **Knowledge and Understanding:**

a1- Understand basic principles of object-oriented program design.

a2- Understand the basic and some advanced issues related to writing classes and methods - such as data, visibility, scope, parameters passing, object references, and nested classes.

a3- Understand the basic ideas behind class hierarchies, polymorphism, and programming to interfaces.

a4- Understand basic principles, main features and operations of abstract data types, such as linked lists, stacks, and queues.

b. **Intellectual Skills:**

b1- Use structured and object-oriented techniques in designing programs.

b2- develop solid programming skills using Java as an object oriented language.

b3- Be able to put in practice the acquired knowledge and understanding of the Java language and object-oriented design in relatively simple case studies.

b4- Be able to develop Java implementations of abstract data types using different approaches, and evaluate their differences.

b5- Be able to use abstract data types and related implementations in designing and implementing efficient solutions to real application problems.

c. Professional and Practical Skills:

c1- Be able to solve a given application problem by going through the basic steps of program specifications, analysis, design, implementation and testing --- within the context of the object-oriented paradigm.

c2- Be able to competently read 'foreign' Java source code and analyze object and class hierarchies.

d. General and Transferable Skills:

d1- The ability to work effectively as an individual, in teams and in multidisciplinary settings together with the capacity to undertake lifelong learning;

d2- The ability to design to design and conduct experiments and to analyze and interpret data.

E. Attitude:

e1. Learn how to make relation with other, and the limit of this relation.

e2. Discuss the legal background of copyright in national and international law.

e3. Demonstrate an ethical behavior toward software copyrights.

3. Contents:

Topic	No. of hours	Lecture	Tutorial/ Practical
OOP, The Big Picture: 1) Introduction to Object Oriented Paradigm 2) *Key concepts:* encapsulation, inheritance, and polymorphism	3	3	0
Introduction to Objects & Classes: 1) Creating Classes	6	3	3

2) Creating, instantiating and using objects 3) Private and public Access members. 4) Constructors, and methods.			
Methods & Arrays: 1) Static member fields and functions. 2) Parameter passing 3) Scope of declaration 4) Method overloading 5) Arrays as Objects 6) Enhanced For	6	3	3
Arrays & Array list 1) Arrays of Objects 2) Multi dimensional Arrays 3) Variable length Argument lists 4) Collections & ArrayList	6	3	3
Classes & objects: A deeper Look 1) Using this reference 2) Overloading constructors 3) Composition 4) Method finalize 5) Static & final members	6	3	3
Understanding Inheritance: 1) Superclass & subclass relationships 2) Protected members 3) Constructors in Parent Class 4) Overloading & Overriding 5) Is-A and Has-a relationships	6	3	3
Polymorphism in action: 1) Introduction to Polymorphism 2) Overriding & Dynamic binding. 3) Abstract Base classes 4) Final methods & classes	6	3	3

Using Interfaces : 1) Creating Interfaces 2) Using Interfaces 3) Case Studies 4) Common Interfaces of Java API	6	3	3
Exception Handling : 1) Overview on Error-handling 2) Using exceptions 3) Try..catch..finally blocks 4) Declaring new exception types	6	3	3
Introducing GUI : 1) Using JOptionPane 2) Swing components 3) Text fields 4) Event handling and nested classes	6	3	3
GUI Components : 1) Working with Buttons 2) Working with Checkboxes & radio Buttons 3) Working with combo boxes & Lists 4) Mouse & keyboard event handling	6	3	3
Files & Steams : 1) Data Hierarchy 2) Class File 3) Sequential Access Text files 4) Object serialization	6	3	3
Multi-threading (*optional*): 1) Introduction to concurrency 2) Creating & executing threads 3) Thread synchronization 4) Multithreading with GUI	6	3	3