

Course specification

(2103 Object Oriented Programming)

Faculty: HICIT

Programme(s) on which the course is given: Under graduate program in Computer Science

Major or minor element of programme: Compulsory

Department offering the programme: Department of Computer Science

Department offering the course: Department of Computer Science

Year / Class: 2nd Year – 1st semester

Date of specification approval: 22/9/2015

A- Basic Information

Title: Object Oriented Programming

Code: 2103

Weekly Hours:

Lecture : 3

Exercise: -

Practical: 4

Total: 7

B- Professional Information

1- Course Objectives:

This module builds on students' previous knowledge of basic programming to provide an introductory approach to object-oriented software development. Fundamentals of classes and objects as key features of programming in terms of design and implementation will be emphasized. Collection objects are also covered and the availability of library classes as building blocks. In addition, students will learn how polymorphism can reduce software production time and implement software re-usability.

After completing this course, the student should be able to:

- a. Build and manipulate with classes.
- b. using objects.
- c. implement the inheritance and polymorphism concepts.

2- Program ILOs Covered by Course

Program Intended Learning Outcomes			
Knowledge and understanding	Intellectual Skills	Professional and practical skills	General and Transferable skills
a2, a11, a13, a20	b3, b4	c1, c5, c10, c16	d5

3 - Intended learning outcomes of course (ILOs)

After Completing this course the student must demonstrate the Knowledge and ability to:

a: Knowledge and Understanding

Upon successful completion of this course, graduates should be able to:

- a1- Understand and apply a wide range of principles and tools available to the object oriented Programming.
- a2- Defining classes and Using objects
- a3- Know and understand the fundamental concepts, principles and theories of a class, object, polymorphism, and inheritance.
- a4- Implement the concepts of inheritance and polymorphism.
- a5- Implement the Abstract classes, Interfaces and OO Model.

b: Intellectual skills

Upon successful completion of this course, graduates should be able to:

- b1-Think in object-oriented programs.
- b2-Build independent modules (classes and objects) that can be used in different programs.

c: Professional and practical skills

- c1- Implement and develop an object-oriented program efficiently.
- c2- reuse the classes that were built during his practical work in computer lab to develop a bigger project.
- c3- use a range of software development tools (e.g. text editor and compiler);
- c4- Use features of an object-oriented programming language (e.g. inheritance, polymorphism to write programs);
- c5- Design appropriate interfaces between modular components;

d: General and transferable skills

- d1- Learn some Internet/Library searching strategies.
- d2- write a short report using appropriate scientific language.
- d3. Use IT skills and display mature computer literacy.

4- Contents

Topic	Hours	Lec.	Exc/Lab
The conceptual basis of Object Orientated Programming	7	3	4
Introduction to the Unified Modeling Language.	7	3	4
Primitive data types and data types as objects. Data Abstraction and encapsulation	14	6	8
Classes and object as abstract data types. Message passing.	7	3	4
An object-oriented programming language syntax, creating objects from class definitions (e.g. C#).	7	3	4
Operators, Operator overloading, delegates, and events	14	6	8
Object oriented programming : Inheritance Access control, Method hiding. Virtual methods and dynamic binding, method overriding	14	6	8
Object oriented programming : Polymorphism, Abstract class, Interface	7	3	4

Exception handling	7	3	4
Course Project	7	3	4

5- Teaching and learning methods

- 5.1 Lectures
- 5.2 Tutorial Exercises
- 5.3 Practical Lab
- 5.4 Discussions.

6 -Student assessment methods

- 6.1 Midterm Exam: To assess the knowledge and understanding achieved by the student during the previous weeks.
- 6.2 Final Exam: To evaluate what the student gain at the end of the course, and to assess: the knowledge and understanding, general skills, and intellectual skills.
- 6.3 Course Project: To allow students work in team, and to evaluate knowledge, understanding, intellectual, and transferable skills.
- 6.4 Course Work & Quizzes: To keep the student always in the course, and to evaluate knowledge, understanding, intellectual, and transferable skills.
- 6.5 Practical Exam: to measure the ability of students to design and implement a software program.

Assessment Schedule

Assessment	Week #
Mid Term Exam	8
Final Exam	16
Course Project	3-14
Course Work & Quizzes	2-14
Practical Exam	15

Assessment Weight

Assessment	Weight %
Mid Term Exam	5%
Final Exam	70%
Course Project	10%
Course Work & Quizzes	5%
Practical Exam	10%
Total	100

Course Work & Quizzes: (Short Exams, Assignments, Researches, Reports, Presentations, Class/Project discussion)

7 -List of references

- 7.1 Text Books
 - Visual C# 2010 How to Program, 4/e, Deitel & Associates,2010
 - Fowler, M., “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, 3rd Edition, Pearson, ISBN10: 0321193687 (2003).

7.2 Internet Location :

Practical Exam	√	√	√	√	√	√	√	√	√	√	√			
----------------	---	---	---	---	---	---	---	---	---	---	---	--	--	--

Course coordinator: Dr Mohamed EL-Zeweidy ()

Head of Department: Dr. Farouk Shabaan ()

Date: 22/9/2015