Course Specification

Program on which the course is given: IT Diploma
Department offering the course: Computer Science
Department offering the course: Computer Science
Academic year /level: 2008/2009 – First year
Date of specification approval: 

A- Basic Information

Title | Object-Oriented Programming
Lecture | Three Hours /Week
Code: | CS502

B- Professional Information

1- Overall Aims of Course:
The student should understand what object-oriented programming is. Through this process, the complexity is reduced and larger programs can be managed. All object-oriented programming languages have three things in common: encapsulation, polymorphism, and inheritance. Also, the student should know that the object-oriented programming has taken the best ideas of structured programming and has combined them with several powerful concepts to allow the programmer to organize his programs more effectively.

2- Intended Learning Outcomes of Course:
a) Knowledge and Understandings:
   At the end of the course, the student will know:
   a1- Different programming approaches.
   a2- The difference between structured programming and object oriented programming.
   a3- Concepts of objects and the method of modeling them in the Object-Oriented Programming System
   a4- The three main features of object-oriented programming: encapsulation, polymorphism, and inheritance.

b) Intellectual Skills:
At the end of the course, the student 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:
At the end of the course, the student will be able to:
c1-Implement and develop an object-oriented program efficiently.
c2-Benefit from the classes that were built during his practical work in the computer lab to develop a bigger project.

d) General and Transferable Skills:
At the end of the course, the student should be able to:
d1-Deal with classes and objects.
d2-Deal with different approaches of programming (object-oriented programming – structured programming).

e) Attitude:
At the end of the course, the student will be able to:
e1-Enhance self-study abilities.
e2-Enhance team-work skills.

3- Course Content:

<table>
<thead>
<tr>
<th>Lecture Topic</th>
<th>Lecture</th>
<th>ILOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision 1:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pointers.</td>
<td>3</td>
<td>A1,A3,A2</td>
</tr>
<tr>
<td>- Pointers with (arrays, functions, structures).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Structures.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Debugging.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revision 2:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Functions (call by value and call by reference).</td>
<td>3</td>
<td>B1,B2,A3</td>
</tr>
<tr>
<td>- Overloading, default arguments.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Storage class specifier.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Class member access.</td>
<td>3</td>
<td>D1,D2,B2</td>
</tr>
<tr>
<td>- Constructor / Destructor.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Inline functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Array of objects.</td>
<td>3</td>
<td>D1,D2,B2</td>
</tr>
<tr>
<td>- Pointers to objects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Object references.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Classes:
- Friend function.
- Overloading constructor functions.
- Assigning objects.
- Passing objects to functions.
- Copy constructor.
- This keyword.

### Operator overloading:
- Member functions.
- Friend functions.
- Binary operators such as (+, -, *, /)
- Unary operators such as (++-, --, (), [])

### Inheritance:
- Overview about inheritance.
- Protection for base class.
- Discuss an example for inheritance.

### Inheritance:
- Inheriting multiple base class.
- Constructors and destructors.
- Discuss an example to illustrate multiple classes.

### Polymorphism:
- Pointers to derived types.
- Virtual function.
- Virtual base class.
- Abstract classes.

### Files Input / Output:
- Streams.
- Stream classes.
- Overloading input / output operators.
- Formatted input / output.

### Templates and exception handling:
- Generic functions.
- Generic classes.

---

**4- Teaching and Learning Methods:**

- Lectures
- Tutorials
- Class discussions
- Internet searches
5- Assessment:
   a) Student Assessment Methods:
      - Assignments
      - Midterm written exam
      - Oral exam
      - Practical exam
      - Final written exam
   b) Assessment Schedule and Weighting:
      - Four assignments with a rate one assignment every 2 weeks (7.5%)
      - One written mid-term exam at the sixth week of the semester (7.5%)
      - One oral and practical exam at the end of the semester (15%)
      - Final written exam (70%)

6- List of Recommended Textbooks:

7- Facilities Required for Teaching and Learning:
   a) Vital Facilities:
      - Computer lab with an object-oriented programming language compiler installed.
      - Data show device.
   b) Lecturing Facilities:
      - Overhead Projector, Data show device.
8- **Attitudes:**

At the end of the course, the students are expected to:

1. Have a positive attitude towards the aim of the course.
2. Like analyzing with software tools and packages in object oriented programming.
3. Be satisfied with the important points of the course contents.

**Course lecturer /Coordinator:**

**Head of the Department:** Prof. Dr. Hamed Nassar.