Course Title: Software Design(1)  
Course code: 721284

Course Level: 2
Course prerequisite (s) and/or corequisite (s): 721283

Lecture Time: 13:10-14:00  
Credit hours: 3

Academic Staff

<table>
<thead>
<tr>
<th>Name</th>
<th>Rank</th>
<th>Office Number and Location</th>
<th>Office Hours</th>
<th>E-mail Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Moayad A. Fahdil</td>
<td>Associated Prof.</td>
<td>IT Faculty-313</td>
<td>14:00-16:00 Saturday/Tuesday/Thursday</td>
<td><a href="mailto:mathami@philadelphia.edu.jo">mathami@philadelphia.edu.jo</a></td>
</tr>
</tbody>
</table>

**Course module description:**
During design phase of software development process, developers define the architecture of the system in terms of design goals and subsystem decomposition. They address global issues, such as mapping of the system onto hardware, the storage of persistent data, and global control flow. This course focuses on how developers can use architectural styles, and UML to deal with solution domain complexity.

**Course module objectives:**
This course is designed to develop in students the knowledge, understanding, skills and values to solve problems through the creation of software solutions. It is a course in concepts and methods for the architectural design of software systems of sufficient size and complexity to require the effort of several people for many months. Fundamental design concepts and design notations are introduced. Several design methods are presented and compared, with examples of their use. Students will undertake a term project working in small groups addressing the design of a relatively complex software system.
Course/module components

- Books (title, author(s), publisher, year of publication)
  
  Title: The Unified Software development process  
  Author(s)/Editor(s): Jackson, Booch and Rumbaugh  
  Publisher: Addison Wesley, 2003

  2. Title: Software engineering 7/e  
  Author(s)/Editor(s): I. Sommerville  
  Publisher: Addison Wesley, 2004

  3. Title: The Unified Modeling Language User Guide  
  Author(s)/Editor(s): Grady Booch, James Rumbaugh, Ivar Jacobson  
  Publisher: Addison Wesley, 1999

  4. Title: Object-Oriented Analysis and Design  
  Author(s)/Editor(s): Grady Booch  
  Publisher: Addison Wesley, 2002

- Support material(s) (vcs, acs, etc).
- Study guide(s) (if applicable)
- Homework and laboratory guide(s) if (applicable).

Teaching methods:
Duration: 15 weeks, 45 hours in total  
Lectures: 35 hours, Tutorial: 8 hours  
Exams (first and second): 2 hours

Learning outcomes:
- Knowledge and understanding
  1) The system development lifecycle;
  2) A wide range of principles and tools available to the software developer, such as software process methodologies, choice of algorithm, language, software libraries and user interface technique;
  3) The principles of object-oriented software construction;

- Cognitive skills (thinking and analysis).
  1) Design and plan software solutions to problems using an object-oriented strategy.
  2) Identify a range of solutions and critically evaluate and justify proposed design solutions.

- Communication skills (personal and academic).
  1) Develop software applications in a development environment that makes use of commonly supported tools.
  2) Identify some of the main risks of software development and use.

- Practical and subject specific skills (Transferable Skills).
  1) Effectively participate in team-based activities.
  2) Structure and communicate ideas effectively, both orally, in writing, and in cases involving a quantitative dimension.
  3) Use IT skills and display mature computer literacy.
  4) Work independently and with others.
**Assessment instruments**

- Short reports and/or presentations, and/or Short research projects
- Quizzes.
- Home works
- Final examination: 50 marks

<table>
<thead>
<tr>
<th>Allocation of Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment Instruments</td>
</tr>
<tr>
<td>First examination</td>
</tr>
<tr>
<td>Second examination</td>
</tr>
<tr>
<td>Final examination:</td>
</tr>
<tr>
<td>Reports, research projects, Quizzes, Home works, Projects</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

**Documentation and academic honesty**

- Documentation style (with illustrative examples)
- Protection by copyright
- Avoiding plagiarism.

**Course/module academic calendar**

<table>
<thead>
<tr>
<th>week</th>
<th>Basic and support material to be covered</th>
<th>Homework/reports and their due dates</th>
</tr>
</thead>
</table>
| (1)  | - Definition of analysis, definition of design  
- Analysis modeling approaches  
(Structured analysis, Object-oriented analysis) | |
| (2)  | **Structured analysis**  
- Data modeling concepts (data objects, data attributes, relationships, cardinality) | |
| (3)  | - Structured analysis model elements  
(DFD, ERD, STD, data dictionary) | |
| (4)  | - Object-oriented analysis (Unified Process)  
Building Analysis Model:  
- Identifying the candidate classes and objects  
Define the interaction between objects | **First Homework** |
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| **(5)** | - **Object-oriented analysis (Unified Process)**  
  - Describe Responsibilities  
  - Describe Attributes and Associations  
  - Qualify Analysis Mechanisms  
  Tutorials |
| **(6)** | **First examination**  
  Software design concept  
  (Abstraction, Architecture, Patterns, Modularity, Stepwise refinements) |
| **(7)** | - Design principles (information hiding, cohesion and coupling)  
  - Levels of cohesion (Coincidental, Logical, Temporal, Procedural, Communicational, Informational, Functional) |
| **(8)** | Levels of coupling (Content, Common, Control, Stamp, Data)  
  **Second Homework** |
| **(9)** |  
  Structured design |
| **(10)** |  
  **Object-oriented design (Unified process)** |
| **(11)** | **Second examination**  
  -Architecture design  
  (Unified process) |
| **(12)** |  
  - System Design (Unified process)  
  **Third Homework** |
| **(13)** |  
  -Class Design  
  (Unified process) |
| **(14)** |  
  - Use case design (Unified process) |
| **(15)** | **Specimen examination (Optional)**  
  Tutorials |
| **(16)** | **Final Examination**  
  Tutorials |

**Expected workload:**  
On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.
Attendance policy:

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

Books

Title: Object-Oriented Software Engineering: Using UML, Patterns, and Java
Author(s)/Editor(s): Bernd Bruegge, Allen Dutoit
Publisher: Prentice Hall, 2003

Title: Modern Systems Analysis and Design
Author(s)/Editor(s): J. Hoffer, and J. Valacich,
Publisher: Prentice Hall

Journals

Websites