COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>Dept., Number</th>
<th>Course Title</th>
<th>Advanced Course Coordinator</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 3331</td>
<td>Advanced Object-Oriented Programming</td>
<td>Nigel Ward</td>
</tr>
</tbody>
</table>

Semester hours 45 hours

Current Catalog Description

The organization and structure of the major hardware components of computers; the mechanics of information transfer and control within digital computer systems.

Textbook:


Other References:

*Head First Object-Oriented Analysis and Design.* Brett D. McLaughlin, Gary Pollice, and Dave West. O'Reilly 2006.


Course Outcomes:

**Level 1: Knowledge and Comprehension**

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level.

Upon successful completion of this course, students will be able to:

1a. Explain the differences between an object-oriented approach and a procedural approach.

**Level 2: Application and Analysis**

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.) Upon successful completion of this course, students will be able to:

2a. Formulate and specify user requirements of a software system using use case diagrams and scenarios.

2b. Use object-oriented design tools such as CRC and UML class diagrams to model problem solutions.

2c. Use basic object-oriented design patterns to structure solutions to software design problems.

2d. Implement association relationships and multiplicities.

2e. Use frameworks, classes, and methods from standard libraries in problem solutions.

**Level 3: Synthesis and Evaluation**
Level 3 outcomes are those in which the students can apply the material in new situations. This is the highest level of mastery.

Upon successful completion of this course, students will be able to:

3a. Design and implement software employing the principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.
3b. Design, implement, and use classes and methods in an object-oriented programming language, employing standard naming conventions and making appropriate use of advanced features such as inheritance, exception handling, I/O, references, and simple GUIs.
3c. Evaluate existing classes and software for the purposes of extension through inheritance.
3d. Use and create standard API documents to understand and document the use of classes and methods.
3e. Design and implement test suites for unit testing.
3f. Re-factor existing software to improve its design or efficiency.

Student Outcomes

Student Outcomes: 1, 2, 3, 6, 8, 9

Prerequisites by Topic

CS 2302 with a grade of “C” or better