CSC/CPE 305 Individual Software Design and Development

1. CSC/CPE 305 Individual Software Design and Development

2. credit units  4      contact hours  6

3. Course Coordinator: Clint Staley

4. Textbook:(and/or other required material)  A C++ Primer, Prata; Core Java V1, Horstmann; Design Patterns, Gamma et al

5. a. Course Description:  Practical software development skills needed for construction of mid-sized production-quality software modules, using the CSC upper division programming language. Topics include inheritance, exceptions, and memory and disk-based dynamic data structures. Students must complete an individual programming project of mid-level complexity. 3 lectures, 1 laboratory. Crosslisted as CPE/CSC 305.

b. Prerequisite:  CSC/CPE 357.

c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

<table>
<thead>
<tr>
<th></th>
<th>CSC</th>
<th>CPE</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Elective</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selective Elective</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

6. a. Course Goals/Outcomes

The student will be able to:

- Knowledge:
  - Thorough familiarity with the features of a standard OO language
  - Use of a standard debugger
  - Use of makefiles or a standard IDE
  - Development and proper organization of multi-sourcefile projects
  - Use of modularity in building a project
  - Use of a module-level testing
  - Use of a language-based assertion facility

- Comprehension:
  - Generic classes (e.g. templates in C++, or genericity through inheritance in Java)
  - Intermediate use (following introduction in 103) of a language-standard class library. (e.g. STL for C++ or Java Collection Classes)
  - Exception handling
  - Constructors and destructors, including copy constructors and assignment operators where applicable
  - Single inheritance, including abstract base classes
  - Class members and methods
o Polymorphic methods and interface types
o Familiarity with object-oriented design patterns
o Application Frameworks

• Application:
o Demonstrated ability to individually write and test mid-sized object-oriented software modules of a professional quality, using a standard OO programming language, and using the concepts listed under “Comprehension.”

b. **How Student Outcomes addressed**
(“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

<table>
<thead>
<tr>
<th></th>
<th>3a</th>
<th>3b</th>
<th>3c</th>
<th>3d</th>
<th>3e</th>
<th>3f</th>
<th>3g</th>
<th>3h</th>
<th>3i</th>
<th>3j</th>
<th>3k</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC</td>
<td>I</td>
<td>A</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>SE/</td>
<td>I</td>
<td>A</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPE</td>
<td>I</td>
<td>A</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. **Major Topics Covered: (number of lecture hours each)**

• Software Construction (15)
o Style rules
o Testing practices, assertions
o Automation of testing and regression testing.
o Basic OO design process
o Use of pointers or references (depending on language)
o Design by Contract

• Class Libraries (5)
o Generic classes
o Standard class library

• Object-Oriented Principles (6)
o Constructors/destructors
o Inheritance
o Virtual functions and polymorphism
o Exceptions and exception handling
o Design Patterns
o Dynamic extension and plug-in architecture