1. **Course number and name:** CSCI 5335 Object-Oriented Design

2. **Credits and contact hours:** 3 credit, 3 contact

3. **Instructor’s or course coordinator’s name:** Wen-Ran Zhang, PhD

4. **Text book, title, author and year:** Head First Design Patterns, Freeman and Freeman Teach Yourself UML in 24 Hours, Schuller
   - **Other supplemental materials:** None

5. **Specific course information**
   a. **Brief description of the content of the course (Catalog Description)**
      Introduction to concepts, methods, and current practice of object oriented design and analysis. Topics include the study of the Unified Modeling Language (UML), which has become an industry standard notation. UML topics will include use cases, diagramming notation (class, object, sequence) and object state diagrams. Students will use UML to design and implement individual and small group projects. Additional topics include understanding design patterns in building applications. Graduate students will be given an extra assignment determined by the instructor that undergraduates will not be required to do.
   b. **Prerequisites:** A minimum grade of “C” in CSCI 3230
   c. **Indicate whether a required, elective, or selected elective course in the program**
      Required course for BS-CS.

6. **Specific goals for the course**
   a. **Specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.**

<table>
<thead>
<tr>
<th>Course Learning Outcomes</th>
<th>Student Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master various software engineering paradigms; become proficient in using OO analysis,</td>
<td>1a, 1b, 1c, 1i</td>
</tr>
<tr>
<td>design, coding, and testing techniques to develop medium scale software in teams</td>
<td></td>
</tr>
<tr>
<td>Become proficient in using design patterns (focus)</td>
<td>1a, 1b, 1c, 1i</td>
</tr>
<tr>
<td>Become proficient in using UML diagrams for OO analysis, design, and testing</td>
<td>1a, 1b, 1c, 1d, 1f, 1i</td>
</tr>
<tr>
<td>Learn to collaborate in project teams, producing project documentation, and performing</td>
<td>1a, 1b, 1c, 1i</td>
</tr>
<tr>
<td>presentations</td>
<td></td>
</tr>
<tr>
<td>Apply OO software product metrics</td>
<td>1a, 1b, 1c, 1i</td>
</tr>
<tr>
<td>Understand programming Language Paradigms</td>
<td>1a, 1b, 1c, 1i</td>
</tr>
</tbody>
</table>
b. Student Outcomes:
   - 1a: An ability to apply knowledge of computing and mathematics appropriate to the discipline
   - 1b: An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   - 1c: An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
   - 1d: An ability to function effectively on teams to accomplish a common goal
   - 1f: An ability to communicate effectively with a range of audiences
   - 1i: An ability to use current techniques, skills, and tools necessary for computing practice

7. Brief list of topics to be covered
   (1) Introduction to UML and OO Design Patterns
      a. Intro to UML for self-learning
      b. Advanced concepts in programming
      c. Intro. to object-oriented metrics.
      d. Intro to design patterns
   (2) Design Patterns with Examples
      - Strategy Pattern
      - Observer Pattern
      - Decorator Pattern
      - Singleton Pattern
      - Factory Pattern
      - Command Pattern
      - Adaptor Pattern
      - Facade Pattern
      - Iterator Pattern
      - Composite Pattern
      - Template Pattern
   (3) OO Analysis and Design with Design Patterns
      a. Use-Case model
      b. Domain model
      c. From requirements to design
      d. Iteration diagram notation
      e. Designing with responsibilities and patterns
   (4) Intro to paradigms of programming languages
      a. Imperative languages
      b. Object-oriented languages
      c. Functional languages
      d. Logical programming languages
   (5) OOA/D Example/Project: Design and implement an OO system with design patterns.