1. **Course number and name:** CSCI 5530 Software Engineering

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

3. **Instructor’s or course coordinator’s name:** Andrew Allen, PhD / James Harris, PhD

4. **Text book, title, author and year:** Schaum Outline of Software Engineering, David Gustafson
   a. **Other supplemental materials:** None

5. **Specific course information**
   a. **Brief description of the content of the course (Catalog Description):** This course serves as a major integrative, capstone experience for students and requires teamwork. A study of the development and management of software; strategies and techniques of design, testing, documentation and maintenance. 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 5330 and CSCI 5335 and CSCI 5423 or CSCI 3432
   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>Understand the main phases of software development</td>
<td>1a, 1b, 1c, 1d, i, 2a, 2b</td>
</tr>
<tr>
<td>Ability to design, develop and maintain software according to software specifications and requirements</td>
<td>1a, 1b, 1c, 1d, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Ability to create all life-cycle technical and management documentation</td>
<td>1a, 1b, 1c, 1d, 1f, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Ability to convey technical material through persuasive and clear oral and written communication, oral presentations, multimedia, and interactions with an audience</td>
<td>1d, 1f</td>
</tr>
<tr>
<td>Ability to function effectively on multidisciplinary teams</td>
<td>1d, 1f</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
   - 2a: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
   - 2b: An ability to apply design and development principles in the construction of software systems of varying complexity

7. Brief list of topics to be covered
   - Capstone course, covers the phases of the software engineering life cycle.