<table>
<thead>
<tr>
<th>Course Number</th>
<th>CS 435</th>
<th>Course Title</th>
<th>Software Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semester Hours</td>
<td>3</td>
<td>Course Coordinator</td>
<td>Koushik Sinha SP20</td>
</tr>
</tbody>
</table>

**Catalog Description**
Principles, practices and methodology for development of large software systems. Object-oriented principles, design notations, design patterns and coping with changing requirements in the software process. Experiences with modern development tools and methodologies. A team project is an integral part of this course.

**Textbooks**

**References**
Various references to tool and language documentation, resources on patterns, principles, etc.

**Course Learning Outcomes**
- To understand and develop experience working within a collaborative team environment.
- To become familiar with concepts of software development methodologies and notations.
- To be able to apply modern development tools and practices to create software both individually and collaboratively.
- To understand basic principles of Object Oriented design and the value of software patterns.

**Assessment of the Contribution to Student Outcomes**

<table>
<thead>
<tr>
<th>Outcome</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessed</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Prerequisites by Topic**
CS 330 with a grade of C or better or graduate standing; CS 306 with a grade of C or better recommended.
## Major Topics Covered in the Course

1. Introduction to software development {2 classes}
2. Perspectives on software process {3 classes}
3. Introduction to software best practices {3 classes}
4. Communication, collaboration and teamwork {6 classes}
5. Software development tools & environment IDE, testing framework, build scripts {3 classes}
6. Coding style and conventions {2 classes}
7. Object oriented principles {5 classes}
8. Practices and process in depth {6 classes}
9. Design notations {3 classes}
10. Software design patterns {5 classes}
11. Anti-patterns {2 classes}