**Course Information**

*Software Engineering: COMP220: 3 credit hour: Pre-requisite COMP 200*

**Office Hours:**  Monday, Wednesday and Friday from 1:00 pm to 2:00 pm

**Online Resources:**

* Lecture slides and notes
* Recorded lectures
* Recommended book

**Recommended Book:** *Software Engineering* By Ian Sommerville.10th edition, Published in 2016

**Course Contents:**

Introduction to software engineering, Software development lifecycle: Plan- driven and Agile, Phases of software engineering, umbrella activities, Requirement engineering, planning the development of software, Tool and techniques for analysis and designing of the software. Use case Model, Domain Model, Sequence diagrams and Class diagrams, quality assurance, validation and verification, test cases,.

**Attendance Policy:**  If *the attendance is less than 75% marks will be deducted.*

**Course Objectives:**

The aim of this course is to introduce students with the discipline of software engineering, give overview of plan-driven and agile software development process models and software process activities, introduce the field of software requirements engineering, develop software system analysis & modelling skills, introduce configuration and version management and software testing disciplines. During this course, the students will get hands on experience of developing use case and structural models of the system using UML. Overall, this course will help students to develop the basic concepts for pursuing further studies in software engineering discipline and / or to start a career as a software developer in industry

**Learning Outcomes**

* Introducing the field of Software Engineering to students.
* Helping students to understand the basic principles of software engineering.
* To develop an understanding of software project planning and the ability to select the suitable model to use in software development.
* Knowledge of all phases of software life cycle including the artifacts that are produced.
* Ability to analyze, design and develop the system models using object oriented methodology for software development.

**Course requirements**

* Students are expected to watch every video and read the notes uploaded on Moodle.
* Assignments shall be uploaded on LMS and instructions to complete these assignments will be provided.
* Students are required to upload completed assignments with in the due dates
* Students are required to form groups of no more than three student for their class project
* Student can select their own project topic, and get it approved by the instructor, or take the topic assigned by the instructor
* Students are required to submit their project with in the due date to get grades.
* Instructor will be available to help you with your problems during office hours, as well as through email.
* Academic dishonesty or cheating will result in zero points and will be referred to AIC (Academic Integrity Committee) at FCC for necessary action.

**Assessment Scheme:**

Students shall be assessed through:

* Quizzes
* Assignments
* Project
* Midterm Examination
* Final term Examination

Announcements will be made on Modle before midterm/final examination, quizzes, assignments, and project due date. They will also be intimated through email by the TA of the course. Before coming for quizzes and submission of assignment, students should read the lecture notes/slides, given reading material and watch the uploaded videos for the given week.

**Course Evaluation:**

Quizzes: 10%

Assignment: 10%

Project: 20%

Midterm Examination: 25%

Final Ter Examination 35%

**Course Plan:**

|  |  |  |
| --- | --- | --- |
| **Week** | **Topics** | **Assessment** |
| **1** | Software, Software engineering, difference between software and Hardware, types pf software, myths about software development |  |
| **2-3** | Software development process, Software development phases, umbrella activities, software development lifecycle models, classical SDLC Models, Waterfall, JAD, RAD, Prototype models | Quiz |
| **4-5** | Incremental, Spiral Models, Agile software development models, Agile manifesto, Agile Principles |  |
| **6** | Agile software development paradigm (continued) Extreme Programming, Scrum , lean and Kanban models | Assignment |
| **7** | Software requirement engineering;  Functional and non-functional requirements,  Software requirement document,  Requirement specification,  Requirement engineering processes,  Requirement elicitation and analysis, | Quiz |
| **8** | Software project Planning: analysis of software scope, analysis of resources, task decomposition and allocation and software project estimation | Assignment |
| **9 & 10** | Software Modeling using UML notation: Why software modeling, Use case model. Use case diagrams, Use case documentations. | Quiz |
| **11** | About Domain Model, Classes and Objects, association aggregation, attributes, Analysis of requirements using domain model | Assignment |
| **12 & 13** | Designing of the software using dynamic model: interaction diagrams such as sequence diagrams, Designing the software using Static Object Model i.e. class diagrams | Quiz |
| **14** | Quality assurance: validation and verification, development of test cases, concepts of reviews and inspection. Configuration Management | Class Project |