Computer Science CSC 310
Software Engineering I

Credits and Contact Hours: 3 Credits, Contact Hours: 42 (1 contact hour = 50 minutes)

Instructor: Dr. Vinitha Hannah Subburaj

Textbook:

Course description:

a. Catalog description: This course is an introductory software engineering course. This course will cover fundamental topics of the software engineering life cycle process from requirements specifications to testing of completed software system. Software development will allow students to implement the software system designed and the programming languages to be used will be according to student and instructor interest. The course will emphasis on writing skills and oral presentation skills which forms the basic platform for students entering into software industries. Software engineering is a course that involves practical implications along with team work. The major part of the course will involve students to work in teams to design and develop software systems. Students will be supervised, but are expected to be organized well while working with team members and developing their presentation and management skills.

b. Prerequisites: CSC-235 & CSC-241/242

c. Required/Elective:
   a. Computer Science – Required
   b. Software Engineering – Required

Specific Goals of Course:

a. Specific outcomes of instruction

After completion of the course, students will be able to

i. Define and use the concepts of software life-cycle process models
ii. Describe and apply fundamentals of software engineering methodologies and techniques
iii. Recognize the importance of requirements gathering phase and elicit requirements using elicitation techniques
iv. Translate software requirements to design, design to code, and then test the software system based on appropriate software engineering methodologies
v. Choose appropriate models, design patterns, architecture, and programming language for real-world software projects based on the particular domain
vi. Develop reliable software following the software life-cycle process that meets customers’ requirements
vii. Employ team work – that includes project management skills, interpersonal, and communication skills.
viii. Define and use basic concepts of software testing and thereafter the maintenance phase of the software life cycle
b. CAC Criterion 3 outcomes addressed by the course:
   a. An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
   b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   c. An ability to design, implements, and evaluate a computer-based system, process, component, or program to meet desired needs
   d. An ability to function effectively on teams to accomplish a common goal
   e. An understanding of professional, ethical, legal, security and social issues and responsibilities
   f. An ability to communicate effectively with a range of audiences
   i. An ability to use current techniques, skills, and tools necessary for computing practice.
   k. An ability to apply design and development principles in the construction of software systems of varying complexity.

c. EAC Criterion 3 outcomes addressed by the course:
   a. An ability to apply knowledge of mathematics, science, and engineering
   b. An ability to design and conduct experiments, as well as to analyze and interpret data
   c. An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
   d. An ability to function on multidisciplinary teams
   e. An ability to identify, formulate, and solve engineering problems
   g. An ability to communicate effectively
   h. The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
   i. A recognition of the need for, and an ability to engage in life-long learning
   k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Brief list of topics to be covered:
   a. Software life cycle models
   b. Requirements engineering
   c. Software design, architecture, and modeling techniques
   d. Use cases (UML diagrams)
   e. Software testing techniques
   f. Software management topics