National Institute of Technology, Silchar
(UG) End Semester Examination, December 2021

Subject Code: CS 307 | Subject: Software Engineering
Semester: 5th | Department: Computer Science and Engineering
Duration: 1 hour 15 minutes | Total Marks: 30

Answer five questions.
Q1 to Q4 are compulsory. Choose any one from Q5 and Q6

Q No.	Questions	Marks	CO
1 (a)	Defile Software quality? Discuss and Elaborate applicability of McCall's Model for Triangle-of-Quality for software development?	3	CO_1
1 (b)	Discuss the different approaches for software quality management?	3	CO_2
2 (a)	What do you understand by Unit Testing? Discuss the Unit test considerations & procedures. [3]	3	CO_1
2 (b)	With appropriate schematic diagrams discuss different variants of Integration testing strategies. [3]	3	CO_2
3 (a)	Define a program P as a collection of tokens, classified by operators or operands. The basic metrics for these tokens were, μ_1 = Number of unique operators μ_2 = Number of unique operands N_1 = Total Occurrences of operators N_2 = Number of unique operators Derive the effort required to generate P?	3	CO_3
3 (b)	Show the steps to calculate Final function count in Albrecht's Function Point Method.	3	CO_3
4 (a)	Discuss the principles for Designing Class-Based Components?	3	CO_2
4 (b)	What are the steps for Component-Level Design.	3	CO_1
5 (a)	Elaborate the three golden rules for UI design. How the UI design is evaluated?	3	CO_3
5 (b)	Discuss the different approaches for risk categorization? Show with an example how do you access the project risk impact?	3	CO_3
6 (a)	What are different classes of software product metrics? Discuss important software product metrics?	3	CO_3
6 (b)	Discuss with appropriate example how Effort estimate is calculated in COCOMO model?	3	CO_3

Course Outcomes (COs):

1. Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

2. Ability to design, implements, and evaluate a computer-based system, process, component, or program to meet desired needs.

3. Ability to apply design and development principles in the construction of software systems of varying complexity.