VI Semester

SOFTWARE TESTING			
Course Code	**21IS63**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course Learning Objectives:

CLO 1. Explain different testing techniques.
CLO 2. Differentiate the various testing techniques.
CLO 3. Apply suitable technique for designing of flow graph.
CLO 4. Analyze the problem and derive suitable test cases.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) does not mean only traditional lecture method, but different type of teaching methods may be adopted to develop the outcomes.
2. Show Video/animation films to explain functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOTS (Higher order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyze information rather than simply recall it.
6. Topics will be introduced in a multiple representation.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module-1

Basics of Software Testing: Humans, Errors and Testing, Software Quality, Requirements Behavior and Correctness, Correctness versus Reliability, Testing and Debugging, Test Metrics, Testing and Verification, Test-generation Strategies, Static Testing.

A Perspective on Testing: Definitions, Test Cases, Insights from Venn Diagram, Identifying Test Cases, Error and fault taxonomies, Levels of testing.

Examples: Generalized pseudocode, the Triangle problem, the NextDate function, the Commission problem, the SATM system, the Currency converter, Saturn windshield wiper

Textbook 1:Ch1,Ch2 Textbook 2:Ch. 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.8, 1.11, 1.12

Teaching-Learning Process	Chalk and talk method/Project based Learning

Module-2

Functional Testing: Boundary Value Testing - Boundary value analysis, Robustness testing, Worst-case testing, Special Value Testing, Examples, Random Testing, Guidelines.

Equivalence Class Testing - Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations,

Decision Table Based Testing - Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations.

Textbook 1: Ch. 5, 6, 7	
Teaching-Learning Process	Chalk and talk method/Project based Learning

Module-3

Structural Testing: Overview, Statement testing, Program testing, Condition testing,

Path testing - DD paths, Test coverage metrics, Basis path testing, guidelines and observations,

Dataflow testing: Definition-Use testing, Slice-based testing, Guidelines and observations.

Textbook 1: Ch 9,10 Textbook 2:Ch. 6.2.1, 6.2.4

Teaching-Learning Process	Chalk and talk method/Project based Learning

Module-4

Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing.

Integration Testing: A closer look at the SATM system, Decomposition-based, call graph-based, Pathbased integrations.

Textbook 1: Ch. 12 & 13.1,13.2,13.3,13.4

Teaching-Learning Process	Chalk and talk method/Project based Learning

Module-5

System Testing: Threads, Requirement Specification, Finding Threads, Structural strategies for thread tesing, SATM test threads System testing guidelines, ASF testing example.

Interaction Testing: Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing

Textbook 1: Ch 14,15

Teaching-Learning Process	Chalk and talk method/Project based Learning

Course Outcomes:

At the end of the course students should be able to:

CO 1. Explain the significance of software testing and quality assurance in software development CO
2. Apply the concepts of software testing to assess the most appropriate testing method.
CO 3. Analyze the importance of testing in software development.
CO 4. Evaluate the suitable testing model to derive test cases for any given software CO
5. Develop appropriate document for the software artefact.

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

Three Unit Tests each of **20 Marks (duration 01 hour**)

1. First test at the end of 5th week of the semester
2. Second test at the end of the 10th week of the semester
3. Third test at the end of the 15th week of the semester

Two assignments each of **10 Marks**

4. First assignment at the end of 4th week of the semester
5. Second assignment at the end of 9th week of the semester

Group discussion/Seminar/quiz any one of three suitably planned to attain the COs and POs for **20 Marks** (**duration 01 hours**)

6. At the end of the 13th week of the semester

The sum of three tests, two assignments, and quiz/seminar/group discussion will be out of 100 marks and will be **scaled down to 50 marks**

(To have less stressed CIE, the portion of the syllabus should not be common /repeated for any of the methods of the CIE. Each method of CIE should have a different syllabus portion of the course).

CIE methods /question paper has to be designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester End Examination:

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the subject (**duration 03 hours)**

1. The question paper will have ten questions. Each question is set for 20 marks. Marks scored shall be proportionally reduced to 50 marks
2. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.

The students have to answer 5 full questions, selecting one full question from each module.

Suggested Learning Resources:

Textbooks:

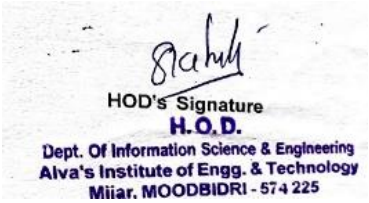
1. Paul C. Jorgensen: Software Testing, A Craftsman‟s Approach, 3rd Edition, Auerbach Publications, 2008.
2. Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008.

Reference Books:

1. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009.
2. Software testing Principles and Practices – Gopalaswamy Ramesh, Srinivasan Desikan, 2 nd Edition, Pearson, 2007.
3. Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004.
4. The Craft of Software Testing – Brian Marrick, Pearson Education, 1995.
5. Anirban Basu, Software Quality Assurance, Testing and Metrics, PHI, 2015.

Web links and Video Lectures (e-Resources):

1. https://nptel.ac.in/courses/106/105/106105150/ 2. https://onlinecourses.nptel.ac.in/noc19_cs71/preview 3. https://www.youtube.com/watch?v=OGImfxO2TEU&t=10s 4. https://www.youtube.com/watch?v=Q50ZyydS7pI 5. VTU e-Shikshana Program 6. VTU EDUSAT Program
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning
• Flip Class • Seminar/Poster Presentation • Role play/Team Demonstration/Collaborative Activity • Mini Project • Case study • Learn by Doing



VI Semester

AGILE TECHNOLOGIES			
Course Code	21CS641	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	3:0:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	03	Exam Hours	03

Course Learning Objectives:

CLO 1. To understand basics of agile technologies
CLO 2. To explain XP Lifecycle, XP Concepts and Adopting XP
CLO 3. To Evaluate on Pair Programming, Root-Cause Analysis, Retrospectives, Planning, Incremental Requirements and Customer Tests
CLO 4. To become Mastering in Agility
CLO 5. To provide well Deliver Value