

| INTRODUCTION TO SOFTWARE TESTING [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – V | | | |
|--|---------|------------|-----------------------|
| Subject Code | 15CS552 | IA Marks | 20 |
| Number of Lecture Hours/Week | 3 | Exam Marks | 80 |
| Total Number of Lecture Hours | 40 | Exam Hours | 03 |
| CREDITS – 03 | | | |
| Course objectives: This course will enable students to <ul style="list-style-type: none"> • Differentiate the various testing techniques. • Analyze the problem and derive suitable test cases. • Apply suitable technique for designing of flow graph. • Explain the need for planning and monitoring a process. | | | |
| Module – 1 | | | Teaching Hours |
| Basics of Software Testing: Basic definitions, Software Quality , Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies , Levels of testing, Testing and Verification, Static Testing. Textbook 3: Ch 1:1.2 - 1.5, 3; Textbook 1: Ch 1 | | | 8 Hours |
| Module – 2 | | | |
| Problem Statements: Generalized pseudo code, the triangle problem, the NextDate function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, NextDate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Textbook 1: Ch 2, 5, 6 & 7, Textbook 2: Ch 3 | | | 8 Hours |
| Module – 3 | | | |
| Fault Based Testing: Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis. Structural Testing: Overview, Statement testing, Branch testing, Condition testing, Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice-based testing, Guidelines and observations. T2:Chapter 16, 12 T1:Chapter 9 & 10 | | | 8 Hours |
| Module – 4 | | | |
| Test Execution: Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay Process Framework :Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties ,Analysis Testing, Improving the process, Organizational factors. Planning and Monitoring the Process: Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the | | | 8 Hours |

| | |
|---|----------------|
| process, the quality team. T2: Chapter 17, 20. | |
| Module – 5 | |
| Integration and Component-Based Software Testing: Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. Levels of Testing, Integration Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing, A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations. T2: Chapter 21 & 22, T1 : Chapter 12 & 13 | 8 Hours |
| Course outcomes: The students should be able to: | |
| <ul style="list-style-type: none"> • Derive test cases for any given problem • Compare the different testing techniques • Classify the problem into suitable testing model • Apply the appropriate technique for the design of flow graph. • Create appropriate document for the software artefact. | |
| Question paper pattern: The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module. | |
| Text Books: | |
| 1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3 rd Edition, Auerbach Publications, 2008. 2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009. 3. Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008. | |
| Reference Books: | |
| 1. Software testing Principles and Practices – Gopalaswamy Ramesh, Srinivasan Desikan, 2 nd Edition, Pearson, 2007. 2. Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004. 3. The Craft of Software Testing – Brian Marrick, Pearson Education, 1995. 4. Anirban Basu, Software Quality Assurance, Testing and Metrics, PHI, 2015 5. Naresh Chauhan, Software Testing, Oxford University press. | |


H.O.D.

Dept. Of Computer Science & Engineering
Alva's Institute of Engg. & Technology
Mijar, MOODBIDRI - 574 225