Semester: 7th			
Paper code: OAE405T	**L**	**T/P**	**Credits**
Subject: Software Verification, Validation and Testing	3	0	3

Marking Scheme:
1. Teachers Continuous Evaluation: As per university examination norms from time to time
2. End Term Theory Examination: As per university examination norms from time to time

INSTRUCTIONS TO PAPER SETTERS: Maximum Marks: As per university norms

1. There should be 9 questions in the end term examination question paper.
2. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions.
3. Apart from Question No. 1, the rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, students may be asked to attempt only 1 question from each unit.
4. The questions are to be framed keeping in view the learning outcomes of course/paper. The standard/ level of the questions to be asked should be at the level of the prescribed textbooks.
5. The requirement of (scientific) calculators/ log-tables/ data-tables may be specified if required.

Course Objectives:

1.	Explain the importance of software verification and validation in the context of AI, ML, IoT, and Data Science.
2.	Apply different testing techniques and methodologies to identify and resolve software defects effectively.
3.	Implement automated testing and utilize test automation tools for efficient and continuous testing.
4.	Evaluate and validate AI/ML models and perform data validation in Data Science projects.

Course Outcomes:

CO1	Understand the concepts of software verification, validation, and testing and their significance in AI, ML, IoT, and Data Science applications.
CO2	Develop expertise in applying various testing methodologies, automated testing, and test automation tools to ensure software quality and reliability.
CO3	Demonstrate the ability to use test management and bug tracking tools effectively to plan, monitor, and manage the testing process.
CO4	Assess the trade-offs between different testing approaches and make informed decisions to ensure comprehensive software testing.

Course Outcomes (CO) to Programme Outcomes (PO)

Mapping **(**Scale 1: Low, 2: Medium, 3: High

CO/ PO	PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	PO12
CO1	2	1	1	-	-	-	-	-	-	-	-	-
CO2	2	1	-	-	2	1	-	-	2	-	-	-
CO3	1	2	3	2	3	1	-	-	-	-	-	-
CO4	2	2	3	2	3	2	-	-	-	3	-	-

Course Overview:
This course introduces students to the principles and techniques of software verification, validation, and testing. It covers the various testing methodologies, tools, and best practices used to ensure the quality and reliability of software in the context of Artificial Intelligence, Machine Learning, Industrial Internet of Things, and Data Science applications.

UNIT I **[8]**
Introduction: Terminology, evolving nature of area, Errors, Faults and Failures, Correctness and reliability, Testing and debugging, Static and dynamic testing, Exhaustive testing: Theoretical foundations: impracticality of testing all data, impracticality of testing all paths, no absolute proof of correctness.

UNIT II **[8]**
Software Verification and Validation Approaches and their Applicability: Software technical reviews; Software testing: levels of testing - module, integration, system, regression; Testing techniques and their applicability-functional testing and analysis, structural testing and analysis, error-oriented testing and analysis, hybrid approaches, integration strategies, transaction flow analysis, stress analysis, failure analysis, concurrency analysis, performance analysis; Proof of correctness; simulation and prototyping; Requirement tracing.

UNIT III **[8]**
Test Generation: Test generations from requirements, Test generation pats, Data flow analysis, Finite State Machines models for flow analysis, Regular expressions based testing, Test Selection, Minimizations and Prioritization, Regression Testing.

UNIT IV **[8]**
Mutation and mutants: Introduction, Mutation and mutants, Mutation operators, Equivalent mutants, Fault detection using mutants, Types of mutants, Mutation operators for C and Java.

Text Books:
1. Software Verification and Validation: An Engineering and Scientific Approach, Marcus S. Fisher, Springer, 2007
2. Foundations of Software Testing, Aditya P. Mathur,Pearson Education, 2008
3. Software Testing: Principles and Practices,SrinivasanDesikan, Gopalaswamy Ramesh, Pearson Education India,2006

Reference Books:
1. "Software Testing: Principles, Techniques, and Tools" by K. K. Aggarwal and Yogesh Singh
2. "Software Testing" by Ron Patton
3. "Testing Computer Software" by Cem Kaner, Jack Falk, and Hung Q. Nguyen

4. "The Art of Software Testing" by Glenford J. Myers, Corey Sandler, and Tom Badgett