Software Testing and Quality Assurance

B. Sc. (Information Technology)		**Semester – VI**	
Course Name: Software Testing and Quality Assurance		**Course Code: USIT601**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- To understand the effective testing techniques for ensuring high quality software.
- To become familiar with quality assurance metrics and comprehend test tool capabilities.
- To become proficient in test project planning, test case and data design, testing operations.
- To understand the challenges and solutions related to software test automation.
- To understand the taxonomy of testing tools.

Unit	Details	Lectures
I	**Introduction to Quality:** Historical Perspective of Quality, What is Quality? (Is it a fact or perception?), Definitions of Quality, Core Components of Quality, Quality View, Financial Aspect of Quality, Customers, Suppliers and Processes, Total Quality Management (TQM), Quality Principles of Total Quality Management, Quality Management Through Statistical Process Control, Quality Management Through Cultural Changes, Continual (Continuous) Improvement Cycle, Quality in Different Areas, Benchmarking and Metrics, Problem Solving Techniques, Problem Solving Software Tools. **Software Quality:** Introduction, Constraints of Software Product Quality Assessment, Customer is a King, Quality and Productivity Relationship, Requirements of a Product, Organisation Culture, Characteristics of Software, Software Development Process, Types of Products, Schemes of Criticality Definitions, Problematic Areas of Software Development Life Cycle, Software Quality Management, Why Software Has Defects? Processes Related to Software Quality, Quality Management System Structure, Pillars of Quality Management System, Important Aspects of Quality Management.	**12**
II	**Fundamentals of testing:** Introduction, Necessity of testing, What is testing? Fundamental test process, The psychology of testing, Historical Perspective of Testing, Definitions of Testing, Approaches to Testing, Testing During Development Life Cycle, Requirement Traceability Matrix, Essentials of Software Testing, Workbench, Important Features of Testing Process, Misconceptions About Testing, Principles of Software Testing, Salient Features of Good Testing, Test Policy, Test Strategy or Test Approach, Test Planning, Testing Process and Number of Defects Found in Testing, Test Team Efficiency, Mutation Testing, Challenges in Testing, Test Team Approach, Process Problems Faced by Testing, Cost Aspect of Testing, Establishing Testing Policy, Methods, Structured Approach to Testing, Categories of Defect, Defect, Error, or Mistake in Software, Developing Test Strategy, Developing Testing Methodologies (Test Plan), Testing Process, Attitude Towards Testing (Common People Issues),	**12**

	Test Methodologies/Approaches, People Challenges in Software Testing, Raising Management Awareness for Testing, Skills Required by Tester, Testing throughout the software life cycle, Software development models, Test levels, Test types, the targets of testing, Maintenance testing.	
III	**Unit Testing: Boundary Value Testing:** Normal Boundary Value Testing, Robust Boundary Value Testing, Worst-Case Boundary Value Testing, Special Value Testing, Examples, Random Testing, Guidelines for Boundary Value Testing, **Equivalence Class Testing:** Equivalence Classes, Traditional Equivalence Class Testing, Improved Equivalence Class Testing, Edge Testing, Guidelines and Observations. **Decision Table–Based Testing:** Decision Tables, Decision Table Techniques, Cause-and-Effect Graphing, Guidelines and Observations, **Path Testing:** Program Graphs, DD-Paths, Test Coverage Metrics, Basis Path Testing, Guidelines and Observations, **Data Flow Testing:** Define/Use Testing, Slice-Based Testing, Program Slicing Tools.	**12**
IV	**Software Verification and Validation:** Introduction, Verification, Verification Workbench, Methods of Verification, Types of reviews on the basis of Stage Phase, Entities involved in verification, Reviews in testing lifecycle, Coverage in Verification, Concerns of Verification, Validation, Validation Workbench, Levels of Validation, Coverage in Validation, Acceptance Testing, Management of Verification and Validation, Software development verification and validation activities. **V-test Model:** Introduction, V-model for software, testing during Proposal stage, Testing during requirement stage, Testing during test planning phase, Testing during design phase, Testing during coding, VV Model, Critical Roles and Responsibilities.	**12**
V	**Levels of Testing:** Introduction, Proposal Testing, Requirement Testing, Design Testing, Code Review, Unit Testing, Module Testing, Integration Testing, Big-Bang Testing, Sandwich Testing, Critical Path First, Sub System Testing, System Testing, Testing Stages. **Testing Tools:** Introduction, Features of Test tools, Guidelines for selecting a tool, Tool and skills of a tester, Static Testing tools, Dynamic Testing tools, Advantages of using Tools, Disadvantages of Using Tools, When to use Automated Test tools, Testing Using Automated Tools, Difficulties while introducing new tools. **Taxonomy of testing tools**: Functional/Regression testing tools, Source code testing tools, Performance testing tools, Java testing tools, Embedded software testing tools, Network protocol testing tools, Configuration management /Bug tracking tools, Testing management tools. How to select a testing tools?	**12**

Books and References:					
Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Software Testing and Continuous Quality Improvement	William E. Lewis	CRC Press	Third	2016

2.	Software Testing: Principles, Techniques and Tools	M. G. Limaye	TMH		2017
3.	Foundations of Software Testing	Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black	Cengage Learning	3rd	
4.	Software Testing: A Craftsman’s Approach	Paul C. Jorgenson	CRC Press	4th	2017
5.	Software Testing Tools	Dr.K. V. K. K. Prasad.	Dreamtech Press		

Course Outcome:

After completing the course, the learner will be able to:
CO1: Learners understand various software testing methods.

CO2: Learners can identify defects and manage those defects for improvement in quality.

CO3: Learners analyze and comprehend the use of modern software testing tools and procedures for their projects testing.

CO4: Understand and apply methods for verifying and validating software to ensure it meets requirements and functions correctly.

CO5: Gain comprehensive knowledge of various testing levels and methodologies to ensure thorough software quality assurance from requirements to system testing stages.