

**COURSE NAME : COMPUTER ENGINEERING GROUP**  
**COURSE CODE : CO/CM/CD**  
**SEMESTER : SIXTH FOR CO/CM AND SEVENTH FOR CD**  
**SUBJECT TITLE : SOFTWARE TESTING**  
**SUBJECT CODE : 9164**

**Teaching and Examination Scheme:**

Teaching Scheme			Examination Scheme						
TH	TU	PR	PAPER HRS.	TH	TEST	PR	OR	TW	TOTAL
04	--	02	03	80	20	--	25@	--	150

**Rationale:**

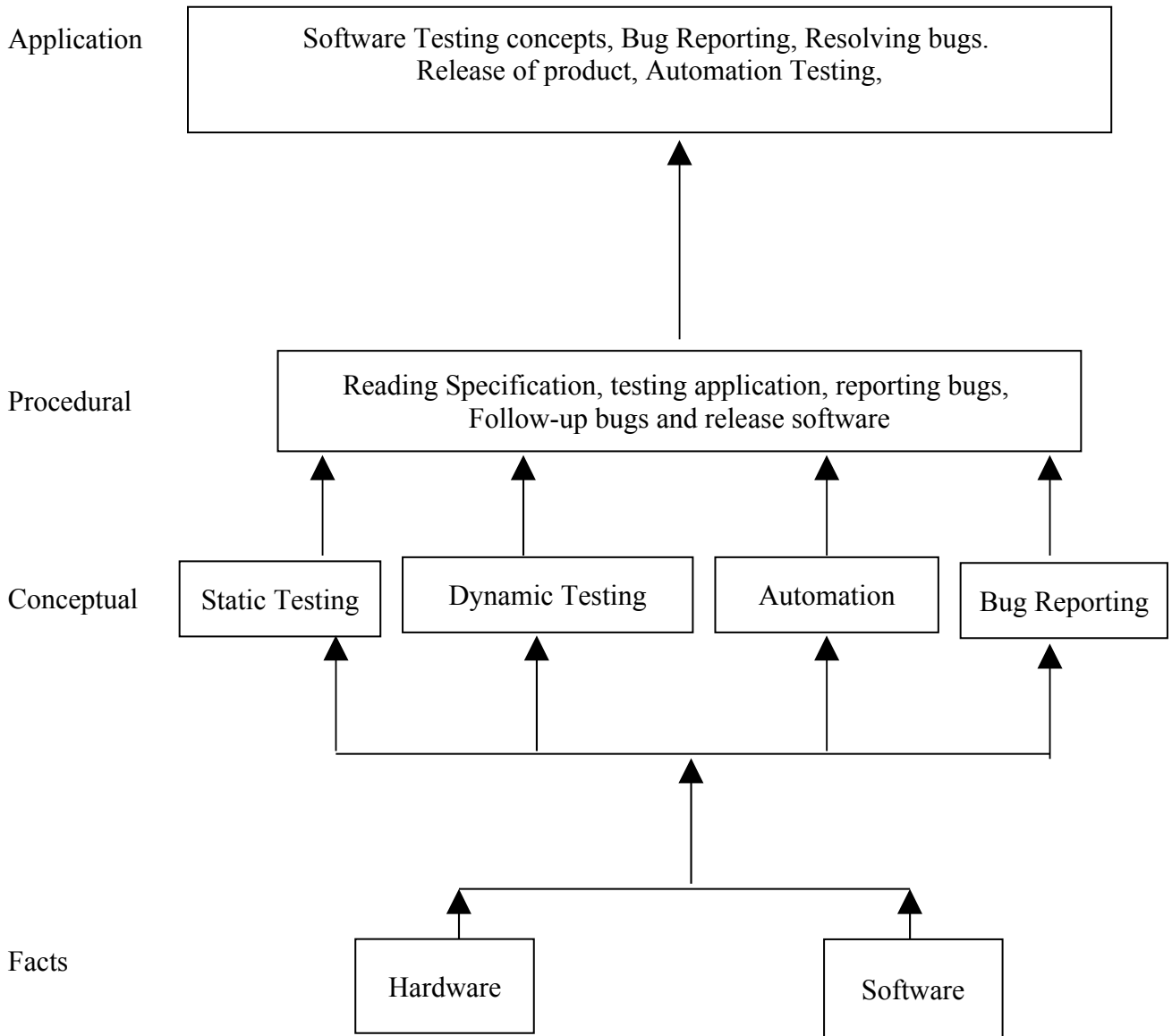
This subject will introduce you to basics of software testing, teaching you not just the fundamental technical skills but also the supporting skills necessary to become a successful software tester. You will learn how to immediately find problems in any computer program, how to plan an effective test approach, how to clearly report your finding and how to tell when your software is ready for release.

**Objectives:**

The students will be able to:

1. Understand the impact of software bugs and importance of software testing
2. Develop the skills necessary to find bugs in any types of software.
3. Learn how to effectively plan your tests, communicate the bugs you find, and measure your success as a software tester.
4. Use your new testing skills to test not just the software , but also the product specification the raw code, and even the user’s manual
5. Learn how to test software for compatibility, usability and cultural issues.
6. Discover how to improve your testing efficiency by automating your tests.

**Learning Structure:**



## Contents: Theory

Chapter	Name of the Topic	Hours	Marks
01	<p><b>Purpose of Testing</b>            Software Testing Background            Software Error Case Studies:- Disney Lion King, Intel Pentium Floating Point Division Bug, NASA Mars Polar Lander, Patriot Missile Defense System, Y2K Bug.            What is Bug? Terms for software Failures, Software Bug: A Formal Definition , Why do Bug occurs? , cost of bugs, What Exactly does a software tester do? What makes a good software tester?            Software Development Process            Product Components:- What Effort Goes into a software product?, What parts make up a software product? , Software Project Staff , Software Development Lifecycle Models :- Big-Bang Model , Code and fix Model, Waterfall model, Spiral Model            The Realities of Software Testing            Testing Axioms: - It's impossible to test a program completely, software testing is a risk-based exercises, testing can't show that bug don't exist, the more bug you find, the more bugs there are, the pesticide paradox.</p>	06	08
02	<p><b>Testing Fundamentals</b>  <b>Examining the Specification</b>            Getting Started :- Black-Box and white-box Testing,            Static and Dynamic Testing , Static Black Box Testing :- Testing the specification            Performing a High Level Review of the Specification:- Pretend to be a customer, Research Existing Standards and guidelines , Review and test similar software            Low Level Specification Test Techniques:- Specification Attributes Checklist , Specification Terminology Checklist.  <b>Testing the software with Blinders On</b>            Dynamic Black-Box Testing : testing the software While, Blindfolded, Test-to-pass and Test-to-fail, Equivalences Partitioning , Data Testing :- Boundary Condition, Sub-Boundary Conditions, default, empty, blank, Null, Zero and None, Invalid, Wrong, Incorrect and garbage data. State Testing:- Testing Software 's Logic Flow, Testing States to Fail. Other Black Box Test Techniques :- Behave like a Dumb User, Look for bugs where you have already found them, follow experience, intuition and hunches</p>	14	16
03	<p><b>Examining the Code</b>  <b>Static White Box Testing:</b> Examining the design and code, Formal Review: - Peer Review, Walkthroughs, Inspections. Coding Standards and Guidelines:- Examples of Programming Standards and Guidelines, Obtaining Standards. Generic Code Review Checklist:- Data Reference Errors, Data Declaration Errors, Computation Errors, Comparison Error, Control Flow Errors, Subroutine Parameter Errors, Input/Output Errors, Other checks.  <b>Testing the software with X-Ray Glasses</b>            Dynamic White Box Testing, Dynamic white box testing versus debugging, Testing the Pieces:- Unit and Integration Testing, An Example of Module Testing.            Data Coverage: - Data Flow, Sub-Boundaries, Formula and Equations,</p>	14	16

	Error Forcing. Code Coverage: - Program Statements and Line Coverage, Branch Coverage, Condition Coverage.		
04	<p><b>Applying Your Testing Skills</b></p> <p><b>Configuration Testing</b>  An Overview of Configuration Testing: - Isolating Configuration Bugs, Sizing up the job. Approaching the Task: - Decide the Types of Hardware You'll Need, Decide What Hardware Brands, Model, and Device Drivers are available. Decide which Hardware features, modes and options are possible. Pare Down the identified Hardware Configuration to a Manageable Set.  Identify your Software's Unique Features that work with the Hardware Configurations. Design the test Cases to Run on each configuration. Execute the tests on each configuration. Rerun the tests until the results satisfy your team. Obtaining the hardware, Identify hardware standards, configuration testing other hardware.</p> <p><b>Compatibility Testing</b>  Compatibility Testing Overview, Platform and Application Versions, Backward and forward compatibility, the impact of testing multiple versions. Standards and Guidelines: - High-Level standards and Guidelines, Low- level standards and Guidelines, Data Sharing Compatibility.</p>	06	08
05	<p><b>Foreign Language Testing</b>  Making the words and Pictures Make Sense , Translation Issues :- Text Expansion , ASCII , DBCS and Unicode , Hot Keys and shortcuts , Extended Characters , Computation on characters , Reading Left to Right and Right to Left , Text on Graphics, Keep the Text out of the code . Localization Issues: - Content, Data Formats.  Configuration and Compatibility Issues: - Foreign platform configurations, Data Compatibility. How much should you Test?</p>	02	04
06	<p><b>Usability Testing</b></p> <p><b>User Interface Testing:</b> What makes a Good UI? , Follows standards or Guidelines, Intuitive, Consistent, Flexible, Comfortable, Correct, Useful. Testing for the Disabled: Accessibility Testing: - It's the Law, accessibility features in software.</p> <p><b>Testing the Documents</b>  Types of Software Documentation, The importance of documentation testing, what to look for when reviewing documentation, the realities of documentation testing.</p> <p><b>Web site Testing</b>  Web Page Fundamentals, Black-Box Testing: - Text, Hyperlinks, graphics, forms, object and other simple miscellaneous Functionality. Gray Box Testing, White Box Testing, Configuration and compatibility testing, Usability Testing, Introducing Automation.</p>	08	10
07	<p><b>Supplementing Your Testing</b></p> <p><b>Automation Testing and test tools</b>  The benefits of automation and tools, Test tools: - Viewers and Monitors, Drivers, Stubs, Stress and load tools, Interference injectors and noise generators, analysis tools. Software Test Automation: - Macro Recording and playback, programmed macros, Fully Programmable Automated Testing Tools. Random Testing: monkeys and gorillas, Dumb monkeys, Semi-smart monkeys, Smart Monkeys, Realities of using test tools and automation.</p> <p><b>Bug Bashes and Beta Testing</b></p>	04	06

	Only as far as the eye can see, Test sharing, beta testing, outsourcing your testing.		
08	<p><b>Working With Test documentation</b></p> <p><b>Planning your test effort :</b> the goal of the test planning , test planning topics :- high level expectations , people , places , and things , definitions , Inter group Responsibilities , what will and won't be tested , test phases , test strategy , resource requirements , tester assignments , test schedule , test cases , bug reporting , Metrics and statistics , Risk and Issues.</p> <p><b>Writing and Tracking Test Cases</b></p> <p>The goal of test case Planning, Test case planning overview, test design, test cases, test procedures, test case organization &amp; tracking.</p> <p><b>Reporting What you Find</b></p> <p>Getting your bugs fixed, isolating &amp; reproducing bugs , Not all bugs are created equal , a bug's life cycle , bug tracking system :- The standard : The test incident Report , Manual Bug Reporting and Tracking , Automated bug reporting and tracking.</p> <p>-Measuring Your Success</p> <p>Using the information in the bug tracking database, Metrics that you'll use in your daily testing, Common Project level Metrics.</p>	06	08
09	<p><b>The Future</b></p> <p><b>Software Quality Assurance :</b> Quality is free, testing and quality assurance in the workplace , software testing , Quality Assurance, other names for software testing groups, Test management and organizational structures, Capability Maturity Model (CMM), ISO 9000</p> <p><b>Your Careers As a Software Tester:</b> Your job as a software tester, finding software testing position, gaining hands-on experience, Internet links, Professional Organizations.</p>	04	04
<b>Total</b>		<b>64</b>	<b>80</b>

**Practical:**

Skills to be developed:

Intellectual Skills:

1. Use installation procedure
2. Creation of GUI objects and their applications
3. Know various tools
4. Know Test procedures

**List of Practical: (Any 10)**

Sr. No.	Practical Name
1	Introduction To Software Testing Concepts
2	Case Study:- Study any system specification and report bugs
3	Write Test Cases For any Application (e.g. Railway Reservation Form)
4	Display “Hello World”
5	Write a program to demonstrate use of 1) For ...Loop 2) Switch ... Case 3) Do...While 4) If....else
6	Automate Notepad Application.
7	Automate any installation procedure (e.g. WinZip)
8	Automate Microsoft Word Application 1) Open Microsoft Word 2) Type text (automatically) 3) Generate random file name. 4) Save file and close Microsoft Word.
9	Create GUI Objects.
10	Create any GUI Application e.g. Calculator
11	Assignment for Web Testing (use any Web testing tools e.g. Selenium)
12	Assignment for any Bug Tracking Tool (e.g. Bugzilla, Bugit)
13	Assignment for any test management tool (e.g. Test Director)

All above Practical may be performed on **Windows or Linux** Platform, using the tools mentioned below:

Sr. No	Testing Tools	Type of Tool
1	AutoIT	Free Ware
2	Ruby	Free Ware
3	Water	Free Ware
4	Sahi	Free Ware
5	Bugzilla	Licensed Software
6	Test Track	Licensed Software

**Learning Resources:****1. Books:**

Sr. No.	Author	Title	Publication
01	Ron Patton	Software Testing	SAMS Techmedia
02	Srinivasan Desikan Gopaldaswamy Ramesh	Software Testing : Principals and Practical	Pearson Education

**2. Sources of Information –**1) [www.autoitv3.com](http://www.autoitv3.com)2) [www.selenium.com](http://www.selenium.com)