

**COURSE TITLE** : **SOFTWARE TESTING**  
**COURSE CODE** : **5069**  
**COURSE CATEGORY** : **E**  
**PERIODS/WEEK** : **4**  
**PERIODS/SEMESTER** : **72**  
**CREDITS** : **4**

**TIME SCHEDULE**

<b>MODULE</b>	<b>TOPICS</b>	<b>PERIODS</b>
1	Testing Methodology	17
	Test I	1
2	Dynamic Testing Techniques	17
	Test II	1
3	Static Testing Techniques	17
	Test III	1
4	Test Automation and Bug Tracking	17
	Test IV	1
	<b>Total</b>	<b>72</b>

**OBJECTIVES**

**MODULE I. Testing Methodology**

**1.1.0 Introduction to software**

- 1.1.1 Goals of software testing
- 1.1.2 Model for software testing
- 1.1.3 Software testing as a process

**1.2.0 Software testing terminology and methodology**

- 1.2.1 Software testing terminology – failure, fault, error etc.
- 1.1.0 Life cycle of a bug
- 1.1.1 States of a bug
- 1.1.2 Bug classification based on criticality
- 1.1.3 Bug classification based on SDLC
- 1.1.4 Testing principles
- 1.1.5 Software Testing Life Cycle (STLC) and different phases
- 1.1.6 Software testing methodology

**1.3.0 Verification and validation**

- 1.3.1 Verification and validation activities
- 1.3.2 Verification requirements

- 1.3.3 Verification of high level design
- 1.3.4 Verification of low level design
- 1.3.5 Verification of code
- 1.3.6 Validation

## **MODULE II: DYNAMIC TESTING TECHNIQUES**

### **2.1.0 Black box testing techniques**

- 2.1.1 Boundary value analysis
- 2.1.2 Equivalence class testing
- 2.1.3 State table based testing
- 2.1.4 Decision table based testing
- 2.1.5 Cause effect graphing based testing
- 2.1.6 Error guessing

### **2.2.0 White box testing techniques**

- 2.2.1 Need of white box testing
- 2.2.2 Logic coverage criteria
- 2.2.3 Basis path testing
- 2.2.4 Data flow testing
- 2.2.5 Mutation testing

## **MODULE III. STATIC TESTING TECHNIQUES**

### **3.1.0 Static testing**

- 3.1.1 Inspections
- 3.1.2 Structured walkthroughs
- 3.1.3 Technical reviews

### **3.2.0 Validation Testing**

- 3.2.1 Unit validation testing
- 3.2.2 Integration testing
- 3.2.3 Function testing
- 3.2.4 System testing
- 3.2.5 Acceptance testing

### **3.3.0 Regression Testing**

- 3.3.1 Progressive Vs. Regression testing
- 3.3.2 Regression testability
- 3.3.3 Objectives of regression testing
- 3.3.4 Regression testing techniques

## **MODULE IV. Test Automation and Bug Tracking**

### **4.1.0 Automation and Testing tools**

- 4.1.1 Need for automation

- 4.1.2 Categories of testing tools
- 4.1.3 Selection of testing tools
- 4.1.4 Guidelines for automated testing
- 4.1.5 Understand open source testing tools – CUT, Cgreen, Emma and Findbugs
- 4.1.6 Overview of commercial testing tools – WinRunner, SilkTest, LoadRunner, Jmeter, TestDirector
- 4.1.7 Understand Object Oriented Testing
- 4.1.8 Understand testing of web based systems

#### **4.2.0 Bug Tracking**

- 4.2.1 Understand debugging and the process of debugging
- 4.2.2 Debugging techniques
- 4.2.3 Correcting the bugs
- 4.2.4 Debugging tools and different types of debuggers

#### **4.3.0 Familiarizing with Testing Tools (Practical)**

- 4.3.1 Familiarize with atleast one open source testing tool for C programs – CUT, Cgreen, cfix etc.
- 4.3.2 Familiarize with atleast one open source testing tool for Java programs – Emma, Findbugs etc.

### **CONTENT OUTLINE**

#### **MODULE I**

Introduction, goals, models, testing as a process  
Terminology, life cycle, states, bug classification, STLC, testing methodology, verification and validation

#### **MODULE II**

Black box testing, white box testing

#### **MODULE III**

Static testing, validation testing, regression testing

#### **MODULE IV**

Automation testing tools, need, selection, open source testing tools, commercial testing tools, object oriented testing, web based testing, bug tracking, familiarization with testing tools (practical)

**TEXT BOOK : Software Testing, Principles and Practices – Naresh Chauhan (Oxford)**

