

**JNTUA COLLEGE OF ENGINEERING (AUTONOMUS): PULIVENDULA****Department of Computer Science & Engineering****B.Tech IV Year I Semester****Subject: SOFTWARE TESTING****Lesson plan**

<b>Course Code</b>	:	15ACS51			
<b>Course Title</b>	:	<b>SOFTWARE TESTING</b>			
<b>Course Structure</b>	:	Lectures	Tutorials	Practicals	Credits
		3	1	-	3
<b>Course Coordinator</b>	:	Dr.S.Jessica Saritha, Asst.Professor, Dept of CSE.			
<b>Team of Instructor</b>	:	Dr. G. Murali Asst.Prof & HOD of CSE			

**I. Course Overview**

The course introduces the main principles underlying Software testing: Purpose of testing, model for testing , consequences of bugs, taxonomy of bugs, path testing, Test Case Design Strategies-Black-Box Approach, White-Box Approach, decision tables, path expressions, kv charts, state testing, Testability tips, On the completion of the units, students will understand the fundamentals of Software testing and be able to test the applications with the help of testing tools like Win runner, QTP, Selenium, Jmeter.

**II. Prerequisite(s):**

<b>Level</b>	<b>Credits</b>	<b>Periods / Week</b>	<b>Prerequisites</b>
UG	3	4	Object Oriented Analysis and Design with UML, Software Engineering

**III. Assessment:**

<b>FORMATIVE ASSESMENT</b>	
Mid Semester Test I (Theory) for 20 Marks in first two and half units is conducted at the end of 9 <sup>th</sup> week.	20 Marks
Mid Semester Test II (Theory) for 20 Marks in last two and half units is conducted at the end of the course work.	
Multiple Choice Test I for 10 Marks in first two and halfunits is conducted at along with Theory exam.	10 Marks
Multiple Choice Test II for 10 Marks in last two and halfunits is conducted at along with Theory exam.	
Note: After evaluating these tests 30 marks are calculated as 80% from best marks and 20% from other.	
Total ( Formative)	30 Marks

<b>SUMMATIVE ASSESMENT</b>	
End Semester Examination in all units is conducted for 70 Marks	70 marks
<b>Grand Total</b>	100 Marks

#### **IV. Course objectives:**

**In Software testing this course, you will learn a range of Analysis of various testing techniques in Software Testing. The learning objectives for Software Testing are:**

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. Analysis of various testing methodologies and procedure to design test cases.

#### **V. Course Outcomes:**

1. Analyze requirements to determine appropriate testing strategies.
2. Apply a wide variety of testing techniques in an effective and efficient manner.
3. Compute test coverage and yield according to a variety of criteria.
4. Evaluate the limitations of a given testing process and provide a succinct summary of those limitations.

#### **VI. Program outcomes:**

- A An ability to apply knowledge of computing, mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems to real-world problems (fundamental engineering analysis skills)
- B An ability to design and conduct experiments, as well as to analyze and interpret data (information retrieval skills)
- C An ability to design , implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints such as economic, environmental, social, political, health and safety, manufacturability, and sustainability (Creative Skills)
- D An ability to function effectively on multi-disciplinary teams (team work)
- E An ability to analyze a problem, identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution (engineering problem solving skills)
- F An understanding of professional, ethical, legal, security and social issues and responsibilities (professional integrity)
- G An ability to communicate effectively both in writing and orally (speaking / writing skills)
- H The broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society (engineering impact assessment skills)
- I Recognition of the need for, and an ability to engage in continuing professional development and life-long learning (continuing education awareness)

- J A Knowledge of contemporary issues (social awareness)
- K An ability to use current techniques, skills, and tools necessary for computing and engineering practice (practical engineering analysis skills)
- L An ability to apply design and development principles in the construction of software and hardware systems of varying complexity (software hardware interface)
- M An ability to recognize the importance of professional development by pursuing postgraduate studies or face competitive examinations that offer challenging and rewarding careers in computing (successful career and immediate employment).

## VII. Syllabus:

### UNIT I

**Introduction:** Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.

**Flow graphs and Path testing:** Basics concepts of path testing, predicates, path predicates and Achievable paths, path sensitizing, path instrumentation, application of path testing.

### UNIT II

Introduction to Testing Design Strategies – The Smarter Tester – Test Case Design Strategies

**Test Case Design Black-Box Approach** - Using Black Box Approach to Test Case Design Random Testing – Requirements based testing –Boundary Value Analysis – decision tables - Equivalence Class Partitioning- state based testing– cause effect graphing – error guessing - compatibility testing – user documentation testing – domain testing.

### UNIT III

**Test Case Design White-Box Approach**-Using White-Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing - Coverage and Control Flow Graphs – Covering Code Logic – Paths – Their Role in White-box Based Test Design – code complexity testing – Evaluating Test Adequacy Criteria.

### UNIT IV

**Paths, Path products and Regular expressions:** path products & path expression, reduction Procedure, applications, regular expressions & flow anomaly detection. **Logic Based Testing:** overview, decision tables, path expressions, kv charts, specifications.

### UNIT V

**State, State Graphs and Transition testing:** state graphs, good & bad state graphs, state testing, Testability tips. **Graph Matrices and Application:** Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to tools like Win runner, QTP, Selenium, Jmeter).

**TEXT BOOKS:**

1. Software Testing techniques, Boris Beizer, Dreamtech, Second Edition
2. SrinivasanDesikan and Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson education, 2006.
3. Software Testing Tools, Dr.K.V.K.K.Prasad, Dreamtech.

**REFERENCES:**

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing, Third Edition, P.C.Jorgensen, Aurbach Publications (Dist.by SPD).
3. Software Testing, N.Chauhan, Oxford University Press.
4. Introduction to Software Testing, P.Ammann and J.Offutt, Cambridge Univ. Press.
5. Effective methods of Software Testing, Perry, John Wiley, Second Edition, 1999.
6. Software Testing Concepts and Tools, P.NageswaraRao, Dreamtech Press.
7. Software Testing, M.G.Limaye, TMH.
8. Software Testing, Desikan, G.Ramesh, Pearson.
9. Foundations of Software Testing, D.Graham and Others, Cengage Learning.
10. Foundations of Software Testing, A.P.Mathur, Pearson.
11. Ilene Burnstein, “Practical Software Testing”, Springer International Edition, Chennai, 2003

**IX. Course Plan:**

The course plan is meant as a guideline. There may probably be changes.

Course Learning Outcomes	Topics to be covered	Reference
This concept tells about How to work real time projects in industry	<b>UNIT I</b> <b>Introduction:</b> Introduction about Software Testing in Real world	T1:page1
Here learn about phases and their purpose	Purpose of testing in real time project assessment	T1:page1
Dichotomies tells about the difference in software testing	Purpose of testing, Dichotomies	T1:page9
Learn the characteristics of the project	model for testing	T1:page15
Learn about test cases of real time applications	Test cases for Atm System, Banking system	T1:page15
Learn about characteristics of the bugs	consequences of bugs	T1:page27
Learn about different types of bugs. Learn about Path testing	Taxonomy of bugs  <b>Path testing:</b> Basics concepts of path testing	T1:page33  T1:page59
Learn about predicates, path predicates and Achievable paths,	predicates, path predicates and Achievable paths, path sensitizing	T1:page92

path sensitizing with the examples		
Learn about applications of path testing	path instrumentation, application of path testing	T1:page109,115
Learn about Smarter Tester and Test Case Design Strategies	<b>UNIT II</b> Introduction to Testing Design Strategies – The Smarter Tester ,Test Case Design Strategies	T2:page47
Learn about Requirements based testing and Random testing	<b>Test Case Design Black-Box Approach -</b> Using Black Box Approach to Test Case Design Random Testing , Requirements based testing	T2:page76
Learn about decision tables , Equivalence Class Partitioning, state based testing, cause effect graphing	Boundary Value Analysis ,decision tables , Equivalence Class Partitioning, state based testing, cause effect graphing	T2:page84,87,90,93,
Learn about compatibility testing, user documentation testing in black box approach	Error guessing, compatibility testing, user documentation testing, domain testing.	T2:page96,99,101
Learn about black box approach	<b>UNIT III</b> <b>Test Case Design White-Box Approach-</b> Using White-Box Approach to Test design	T2:page47
Learn about Test Adequacy Criteria in black box approach	Test Adequacy Criteria	T2:page47
Learn about differences between static testing vs. structural testing	static testing vs. structural testing	T2:page48,56
Learn about code functional testing	code functional testing	T2:page56
Learn about Control Flow Graphs ,Covering Code Logic	Coverage and Control Flow Graphs ,Covering Code Logic	T2:page57
Learn about main role in white box based test design	Paths , Their Role in White-box Based Test Design	T2:page57
Learn about code complexity testing	code complexity testing	T2:page63
Learn about Evaluating Test Adequacy Criteria	Evaluating Test Adequacy Criteria	T2:page67
Learn about path products & path expression	<b>UNIT IV</b> <b>Paths, Path products and Regular expressions:</b> path products & path expression	T1:page244
Learn about applications	reduction Procedure, applications	T1:page251,257
Learn about regular expressions	regular expressions & flow anomaly detection	T1:page278
Learn about decision tables, path expressions	<b>Logic Based Testing:</b> overview, decision tables, path expressions	T1:page320,322,332

Learn about kv charts and their specifications	kv charts, specifications.	T1:page343,352
Learn about state graphs, good & bad state graphs	<b>UNIT V</b> <b>State, State Graphs and Transition testing:</b> state graphs, good & bad state graphs	T1:page364,373
Learn about Testability tips	state testing, Testability tips	T1:page387,391
Learn about matrix of graph	<b>Graph Matrices and Application:</b> Motivational overview, matrix of graph	T1:page397,399
Learn about relations of graphs and their power of a matrix	relations, power of a matrix	T1:page402,405
Learn about node reduction algorithm and their building tools	node reduction algorithm, building tools.	T1:page415,421

#### **X. Mapping course outcomes leading to the achievement of the programme outcomes:**

<b>Course Outcomes</b>	<b>Program Outcomes</b>												
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
<b>1</b>	<b>H</b>										<b>S</b>		
<b>2</b>				<b>S</b>							<b>H</b>		
<b>3</b>	<b>S</b>					<b>S</b>							
<b>4</b>			<b>H</b>								<b>S</b>		

S= Supportive

H=Highly Related

#### **Justification of Course syllabus covering Course Outcomes:**

By covering the syllabus a student can understand the designing of algorithm and flowcharts. Student is able to develop applications using C Program Constructs.

**Justification of CO's –PO's Mapping Table:**

By mapping CO-1 to the PO's A & K which are related to the course CO1: The student is able to analyze requirements to testing strategies and know the techniques, skills, and tools.

By mapping CO-2 to the PO's D & K which are related to the course CO2: The student is able to testing techniques in an effective and efficient manner.

By mapping CO-3 to the PO's A & F which are related to the course CO3: The student is able to do testing techniques in professional, ethical, legal, security issues.

By mapping CO-4 to the PO's C & K which are related to the course CO4: The student is able to to design , implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints