INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING

COURSE DESCRIPTION FORM

Course Title	**SOFTWARE TESTING METHODOLOGIES**			
Course Code	**A60525**			
Regulation	**R13 - JNTUH**			
Course Structure	Lectures	Tutorials	Practical	Credits
	4	1	-	4
Course Coordinator	Ms N Shalini , Assistant Professor, CSE			
Team of Instructors	Dr. N Rajasekhar, Professor, CSE Ms. J Hareesha, Assistant Professor, CSE Ms. M Siva Swetha Reddy, Assistant Professor, CSE			

I. COURSE OVERVIEW:

The software testing is a process of executing a program or application with the intent of finding the bugs. This course will help students learn catch bugs and break software as you discover different testing methods that will help build better software. It will teach and make students think like a software tester and help in finding bugs in code earlier and write better code. The course demonstrates an in-depth understanding of the tools and technologies for software testing and do better programming and test the programs efficiently.

II. PREREQUISITE(S):

Level	Credits	Periods/ Week	Prerequisites
UG	4	5	Software Engineering

III. MARKS DISTRIBUTION:

Sessional Marks	University End Exam marks	Total marks
Midterm Test There shall be two midterm examinations. Each midterm examination consists of essay paper, objective paper and assignment. The essay paper is for 10 marks of 60 minutes duration and shall contain 4 questions. The student has to answer 2 questions, each carrying 5 marks. The objective paper is for 10 marks of 20 minutes duration. It consists of 10 multiple choice and 10 fill-in-the blank questions, the student has to answer all the questions and each carries half mark. First midterm examination shall be conducted for the first two and half units of syllabus and second midterm examination shall be conducted for the remaining portion. Five marks are earmarked for assignments. There shall be two assignments in every theory course. Assignments are usually issued at the time of commencement of the semester. These are of problem solving in nature with critical thinking. Marks shall be awarded considering the average of two midterm tests in each course.	75	100

IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1.	I Mid Examination	80 minutes	20
2.	I Assignment	-	5
3.	II Mid Examination	80 minutes	20
4.	II Assignment	-	5
5.	External Examination	3 hours	75

V. COURSE OBJECTIVES:

At the end of the course, the students will be able to:

I. Understand the concepts of software testing objectives, process criteria, strategies and methods.

II. Learn various software testing issues and solutions in software unit testing, integration, regression and system testing.

III. Explore on advanced software testing topics such as object oriented software testing methods and component based software testing issues, challenges and solutions.

IV. Identify the techniques and skills on how to use modern software testing tools to support software testing projects.

V. Describe the important concepts of complexity metrics and object oriented metrics.

VI. Observe the foundations, techniques, and tools in area of software testing and its practice in the industry.

VI. COURSE OUTCOMES:

After completing this course the student must demonstrate the knowledge and ability to:

1. Understand the importance and purpose of testing and its applications in software development life cycle.
2. Compare and contrast the various terminologies used in dichotomies of testing.
3. Understand the model for testing by taking a project, environment, program and bugs.
4. Describe the consequences, Nightmare list and taxonomy of bugs in software testing.
5. Describe path instrumentation, path sensitization implementation and their applications.
6. List transaction flows ,their techniques and implementation comments in software testing.
7. State basics of data flow testing and strategies in dataflow testing and their testability tips
8. Compare domain testing and path testing and explains various domain techniques.
9. Understand domains and interface testing and their testability tips.
10. State path product, path expression and procedure to minimize path expressions.
11. Design reduction procedure and its applications and lists regular expressions and flow anomaly detection.
12. Describe logic based testing, decision tables and path expressions for logic based testing.
13. Write path expression, KV charts, specifications and more testing strategies.
14. Design and implement state graph, state testing, good state graph, bad state graph and their testability tips.
15. Describe graph matrices, matrix properties and node reduction algorithm.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

	Program Outcomes	Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Assignments and Tutorials
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	H	-----
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	H	Assignment and Exams
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	H	Practicals
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Tutorials
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	N	----
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	N	----
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	----
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	S	----
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	N	----
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply **these to one's own** work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	S	Assignments
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	N	----
N – None	**S – Supportive**	**H - Highly Related**	

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

	Program Specific Outcomes	Level	Proficiency assessed by
PSO1	**Professional Skills:** The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	H	Lectures, Assignments
PSO2	**Problem-Solving Skills:** The ability to apply standard practices and Strategies in software project development using open-ended programming environments to deliver a quality product for business success.	H	Laboratories
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	H	Guest Lectures
N – None	**S – Supportive**	**H - Highly Related**	

IX. 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

Transaction Flow Testing: Transaction flows, transaction flow testing techniques.
Dataflow testing : Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

UNIT-III

Domain Testing: Domains and paths, Nice and ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

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, k v charts, specifications.

UNIT-V

State, State Graphs and Transition testing: State graphs, good and 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 tool like jmeter and win runner)

Text books:

1. Boris Beizer, "Software Testing Techniques", Dreamtech, 2nd Edition .
2. Dr. K. V. K. K. Prasad, "Software Testing Tools", Dreamtech.

References:

1. Brian Marwick, "The Craft of Software Testing", Pearson Education.
2. P. C. Jorgenson, "Software Testing", 3rd Edition, Aurbach Publications (Dist.by SPD).
3. N. Chauhan, "Software Testing", Oxford University Press.
4. P. Ammann, J. Offutt, "Introduction to Software Testing", Cambridge University Press.
5. Perry, "Effective methods of Software Testing", John Wiley, 2nd Edition, 1999.

X. COURSE PLAN:

At the end of the course, the students are able to achieve the following course learning outcomes:

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
1-2	**Introduction**: Purpose of testing.	**Explain** the importance of testing and purpose of testing.	T1:1.1
3-4	Dichotomies, model for testing.	**Illustrate** different and compare dichotomies of testing.	T1:1.2
5-6	Model for testing.	**Demonstrate** the model for testing and different testing levels and role of models.	T1:1.3
7-9	Consequences of bugs, taxonomy of bugs.	**Describe** the consequences and taxonomy of bugs and different bugs in project environment.	T1:2.2
10-13	Path testing and predicate, loops and path sensitization.	**Illustrate** the concepts of path testing and predicate loops and path sensitization.	T1:3.2
14-15	Path instrumentation and their applications and link markers.	**Explain** Path instrumentation and their applications and link markers.	T1:3.5
16-19	Transaction flows techniques ,Transaction flows, transaction flow testing technique	**List** Transaction flows techniques and transaction flow structures and their test databases.	T1:4.3
20-23	Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.	**State** Basics of data flow testing and Strategies in data flow testing, applications of dataflow testing.	T1:5.2
24-27	Domains and paths, Nice and ugly domains, domain testing.	**Describe** Domains and paths and explain about domains and bugs and their tools effectiveness.	T1:6.2
28-31	Domains and interfaces testing, domain and interface testing, domains and testability.	**Demonstrate** Domains and Interfaces testing .explain linearising transformation and coordinate transformation.	T1:6.5
32-33	Path products and path expression.	**State** Path products and path expression, different laws used in path testing.	T1:8.3
34-38	Reduction procedure, applications, regular expressions and flow anomaly detection.	**Demonstrate** Reduction procedure and applications, Regular expressions and Flow anomaly detection.	T1:8.4
39-44	Logic based testing and decision tables.	**Describe** Logic based testing and Decision tables and compare hardware and software testing.	T1:10.2
45-51	Path expressions, k v charts, specifications.	**Illustrate** Path expression and KV Charts and their specifications.	T1:10.4
52-59	State graphs, good & bad state graphs, state testing, Testability tips.	**Explain** State Graphs and state testing and their Testability Tips. Explain finite state behavior in state graphs.	T1:11.3
60-63	Motivational overview, matrix of graph, relations, power of a matrix.	**Describes** Graph Matrices and Node reduction algorithm. Explain break loops and their applications.	T1:12.2
64-65	Node reduction algorithm, building tools.	**Demonstrate m**atrix properties and node reduction algorithm.	T1:12.6

XI. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I.	H				S								H	S	
II.			H		S								S		H
III.			H						S						
IV.				H					S				H	S	H
V.		H									S				
VI.			H										H	H	S
S-Supportive													**H- Highly Related**		

XII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
1.	H				S								H		
2.			H						S				S		
3.	H				S									H	S
4.				H					S				H		
5.	H										S			S	H
6.			H								S				
7.				H					S				S	H	
8.			H		S										
9.				H							S			S	H
10.	H								S						
11.			H		S								S	H	
12.	H								S						
13.			H		S										
14.		H							S						
15.			H								S		H		H
S-Supportive													**H- Highly Related**		

Prepared by: Ms. N Shalini , Assistant Professor, CSE

HOD, CSE