SOFTWARE TESTING

PROF. MEENAKSHI D’SOUZA
Dept. of Computer Science and Engineering
IIIT Bangalore

TYPE OF COURSE : Rerun | Elective | UG/PG
COURSE DURATION : 12 weeks (26 Jul’21 - 15 Oct’21)
EXAM DATE : 24 Oct 2021

INTENDED AUDIENCE : Any Interested Learner
PRE-REQUISITES : Programming, Algorithms, Discrete Mathematics (basics)
INDUSTRIES APPLICABLE TO : The material of this course has been used to offer training for Samsung, ABB and Mindtree. The course will be useful for any firm that does tests their software.

COURSE OUTLINE :
This course will cover various techniques for test case design, as used for testing of software artifacts including requirements, design and code. We will discuss algorithms and techniques for test case design based on graphs, logic, syntax of programming languages and on inputs. Special techniques for testing object-oriented features and web applications will also be discussed. The course will end with symbolic testing techniques. These broadly will cover test cases for both white-box and black-box.

ABOUT INSTRUCTOR :
Prof. Meenakshi D Souza is currently an Associate Professor at IIIT-Bangalore. Meenakshi did her Master’s in Mathematics from University of Madras, Chennai and her Ph. D. in Theoretical Computer Science from The Institute of Mathematical Sciences, Chennai. She joined the research department of Honeywell Technology Solutions, Bangalore soon after completing her Ph. D. and worked there in the areas of Formal Verification of Software Design, Model Based Development and Physical Access Control before joining IIIT-Bangalore. Her research interests are in Formal Methods, Model Based Development, Software Testing and Automata Theory.

COURSE PLAN :

Week 01 : Techniques and algorithms for test case design: Graphs based testing- structural coverage criteria.
Week 02 : Graphs based testing: Data flow coverage criteria
Week 03 : Graphs based testing: Data flow coverage criteria (contd)
Week 04 : Graphs coverage for source code, design elements and requirements
Week 05 : Techniques and algorithms for test case design: Logic based testing- Predicates, logic based coverage criteria
Week 06 : Specification based logic coverage, logic coverage on finite state machines
Week 07 : Input space partitioning: Input domain modeling, combination strategies criteria
Week 08 : Syntax based testing: Coverage criteria based on syntax, mutation testing
Week 09 : Test case design (as learnt above) applied to object-oriented applications
Week 10 : Test case design (as learnt above) applied to web applications
Week 11 : Symbolic testing
Week 12 : Concolic testing, Conclusion