ABHISHEK CHANDRA
(41311503118)
IT-M

SOFTWARE TESTING
QUIZ - (4 MARKS)

A1 In oops programs, control flow is characterized by message passing among objects, and the control flow switches from one object to another by inter object communication. Consequently, there is no control flow within a class requires different approaches for testing. Furthermore, in a function, arguments passed to the function with global data determine the path of execution within the procedure. But, in an object, the state associated with the object also influences the path of execution, and methods of a class can communicate among themselves through this state because this state is persistent accross invocations of methods. Hence, for testing objects, the state of an object has to play an important role. Techniques of OOP Testing are as follows:

(1) Fault based ~~techniques~~ testing:
(2) Class testing based on method testing
(3) Random Testing.
(4) Partition Testing
(5) Scenario-based testing

These tests tend to search out interaction form of error.

~~State Based testing~~

Class testing: This approach is the simplest approach to test classes. Each method of the class peryorms a well defined cohesive function and can, therefore, be related to unit testing of the traditional testing techniques. Therefore all the methods of a class can be involved at least once to test the class.

Composition

Issues with class testing

→ Object of OO is to facilitate easy code reuse in the form of classes, To allow each class has to be rigirously unit tested.

→ Due to classes potentially used in unforeseenable ways when composed in new systems.

→ Example: A XML parser for a web browses

→ Classes must be created in a way promoting loose coupling and strong cohesion.

① Enclapsulation issues → Encapsulation requires that classes are only aware of their own properties, and are able to operate independently. If unit testing is peryormed with the integration testing becomes more important.

② Inheritance & Polymorphism :→ Inheritance is an important part of OOP paradigm, Unit testing a class with a superclass can be impossible to do without the super class methods / variable

→ Repeatedly testing same methods

→ Time can then be wasted is not addressed

→ Potentially can be avoided and actually save time.