

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
Course: Software Engineering
Course CODE: CS-304
Max.Marks : 15

Class Test: II

Session: 2017-18
Year: IIIrd
Time: 16:00 to 17:00
Date: 23-01-2018

Instructions to candidates: Attempt Any Three Questions

SI#	Question	Marks	CO MAPPING
1	Explain Verification and Validation.	5	CO4
2	Explain Black Box Testing with Boundary Value Analysis.	5	CO4
3	Explain Integration Testing and System Testing.	5	CO4
4.	Construct a DFD of Loan Department of a Bank.	5	CO3

Q.1 Explain Verification and Validation.

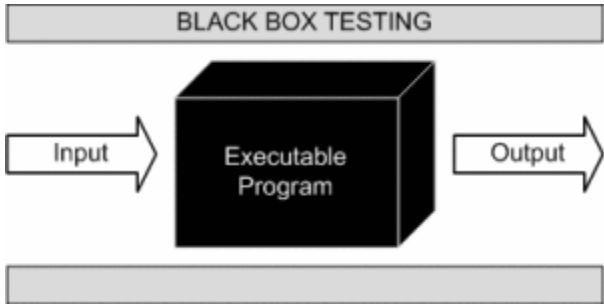
Ans.

Difference between software **Verification and Validation**:

Verification	Validation
Are we building the system right?	Are we building the right system?
Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements.	Validation is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.
The objective of Verification is to make sure that the product being develop is as per the requirements and design specifications.	The objective of Validation is to make sure that the product actually meet up the user’s requirements, and check whether the specifications were correct in the first place.
Following activities are involved in Verification : Reviews, Meetings and Inspections.	Following activities are involved in Validation : Testing like black box testing, white box testing, gray box testing etc.
Verification is carried out by QA team to check whether implementation software is as per specification document or not.	Validation is carried out by testing team.
Execution of code is not comes under Verification .	Execution of code is comes under Validation .
Verification process explains whether the outputs are according to inputs or not.	Validation process describes whether the software is accepted by the user or not.
Verification is carried out before the Validation.	Validation activity is carried out just after the Verification.
Following items are evaluated during Verification : Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc,	Following item is evaluated during Validation : Actual product or Software under test.
Cost of errors caught in Verification is less than errors found in Validation.	Cost of errors caught in Validation is more than errors found in Verification.
It is basically manually checking the of documents and files like requirement	It is basically checking of developed program based on the requirement specifications

specifications etc.	documents & files.
---------------------	--------------------

Q.2 Explain Black Box Testing with Boundary Value Analysis.
BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

For the most part, errors are observed in the extreme ends of the input values, so these extreme values like start/end or lower/upper values are called Boundary values and analysis of these Boundary values is called “Boundary value analysis”. It is also sometimes known as ‘range checking’.

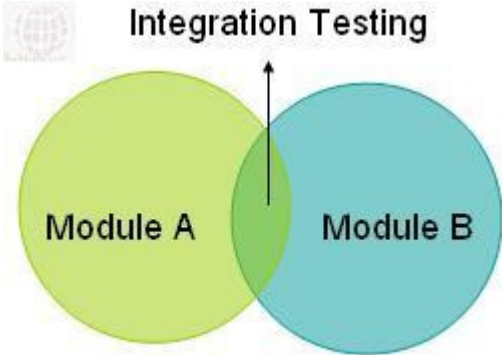
Boundary value analysis is another black box test design technique and it is used to find the errors at boundaries of input domain rather than finding those errors in the center of input.

Equivalence Partitioning and Boundary value analysis are linked to each other and can be used together at all levels of testing. Based on the edges of the equivalence classes, test cases can then be derived.

Each boundary has a valid boundary value and an invalid boundary value. Test cases are designed based on the both valid and invalid boundary values. Typically, we choose one test case from each boundary.

Finding defects using Boundary value analysis test design technique is very effective and it can be used at all test levels. You can select multiple test cases from valid and invalid input domains based on your needs or previous experience but remember you do have to select at least one test case from each input domain.

Ans 3. **Integration testing** (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.



System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.^[1]

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

Ans 4.

