

Chirea Short Questions
LIVIU
032

Q1: Black-Box Testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. Equivalence classes refers to a technique that divides the input data of a software unit into partitions of equivalent data from which tests can be derived, by trying to cover each partition at least once.

Example: A Login functionality: we can divide usernames into 3 partitions:

- Valid usernames
john123

- Invalid usernames
@john11

- Edge cases
such as trying
the min. & max. length

Q2: White-Box Testing is a software testing technique that involves examining the internal structure, code, and logic of the system being tested. With the knowledge of the internal workings, we can design test cases that exercise specific paths in the code. Predicate coverage is a criterion used in white box testing to ensure that every boolean condition in the code is evaluated to both true and false at least once.

Example: if a function checks whether a number is prime, we need:

- Prime Case
- 7, 11, 5 => true

- Non-Prime Case
- 2, 4, 12 => false

Q3: Symbolic execution is a White-Box Testing technique that explores the different paths of a program by using symbolic values instead of concrete inputs. It allows for systematic exploration of all conditions and branches by treating inputs as symbols and tracking their constraints. A tree is a representation of these paths, put together in a tree-like structure. Each path in the tree represents a unique scenario based on inputs and the conditions encountered.

Example: We buy groceries. If we have no fruits at all, we buy 2 bananas. If we have one banana, we buy only one more. If we have > 2 bananas, we buy an apple:

Q4: Model checking is a testing technique used to verify whether a given system or model satisfies a set of specified properties. It tries to explore all possible states of a system to check if certain properties hold or if there are any violations.

Liveness properties are a type of property that specify desirable

behaviors in a system, focusing on the occurrence of events or the progress of the system over time.

Example: if we have $L_1 \rightarrow L_2$ (when L_1 is true then L_2 is true), we use model checking to verify the transition holds true, looking for any possible counter-example that would indicate a bug.

