

## EXPERIMENT NO.:08

**Date of Performance:**

**Date of Submission:**

**Aim:** Write test cases for white box testing

Software Used: : **Selenium/GitHub/Jira**

### **Theory:-**

**White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

White box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

Here, the test engineers will not include in fixing the defects for the following reasons:

- Fixing the bug might interrupt the other features. Therefore, the test engineer should always find the bugs, and developers should still be doing the bug fixes.
- If the test engineers spend most of the time fixing the defects, then they may be unable to find the other bugs in the application.

## Techniques Used in White Box Testing

<u>Data Flow Testing</u>	Verify that the variable used to store the product price is correctly initialized before any calculations. This test ensures that the application correctly captures and retains the initial input.
<u>Control Flow Testing</u>	Test the execution flow of the program when a user searches for a product. This test ensures that the correct functions are called, and the application navigates through the search logic appropriately.
<u>Branch Testing</u>	Test the execution flow of the program when a user searches for a product. This test ensures that the correct functions are called, and the application navigates through the search logic appropriately.
<u>Statement Testing</u>	Confirm that every executable statement within the price calculation function is executed at least once, ensuring complete coverage of all statements.
<u>Decision Testing</u>	Verify that the application correctly evaluates the decision points when determining discount eligibility based on user status. This ensures that all logical paths are tested based on conditions.

*Let us consider the following code :*

INPUT A & B

C = A + B

IF C>100

PRINT "ITS DONE"

Now in the first, line, we assign the value of A and B. Let us suppose A = 60 and B = 50. Moving on to the second line, now C is assigned a value of A+B, here A = 60 and B = 50, hence C = 110. Moving on to the third line, we will check if C > 100, here the condition is true and hence we should get our result as ITS DONE

**Conclusion :** Thus we have written test cases for black box testing .

**Sign and Remark:**

R1	R2	R3	Total Marks	Signature
(5)	(5)	(5)	(15)	