

## Experiment - 2

**Aim :** Design and write test cases for a simple calculator application.

### Theory :

- **Black Box Testing :**

It is a software testing method in which the functionalities of software method in which the functionalities of software applications are test without having knowledge of the internal code structure, implementation details, & internal paths.

- More focus is given on functional requirements.
- The application is tested from the user's perspective.
- Behavior of application is test rather than single modules.

- **Three types of black box testing include:**

1. **Functional Testing :** Testing functional requirements of the application.
2. **Non-Functional Testing :** Tests the readiness, performance, scalability, flexibility & efficiency of the application.
3. **Regression Testing :** It is performed after upgrade, code fixes, or any other system maintenance to check that the new changes have not affected any other existing functionality.

- **Analysis of Simple Calculator Applications :**

Requirements for a simple calculator application will be :-

1. Take input through buttons
  - a. Number buttons 1 2 3 4 5 6 7 8 9 0.
  - b. C , AC , <- buttons for clearing input
  - c. Decimal point button
  - d. E(x10<sup>x</sup>) for exponent values
2. Take input operations from buttons
  - a. Basic mathematical operations using buttons + - \* /
  - b. If an operation exists, then replace it with the newly created operations.
3. Calculate the result using the BODMAS & law of indices
4. Display result.
  - a. The preview calculation result of the input expression if the result (=) button is not pressed.
  - b. If the result (=) button is pressed then display the result as the final result.

### **Analysis :**

1. As input is taken only through buttons provided by the application only, there will not be any unexpected inputs.
2. Input or output can be unexpected only during the data overflow. Hence equivalence class partitioning & boundary value analysis are required to check the valid range of inputs & outputs.

- **Equivalence partitioning :**

Maximum Input : 999,999,999,999,999

Minimum Input : -999,999,999,999,999

Valid class :

Partition 1 : From 999,999,999,999,999 to -999,999,999,999,999

Invalid classes :

Partition 2 : Number greater than 999,999,999,999,999

Partition 3 : Number smaller than -999,999,999,999,999

• **Observation/Results :**

| Case ID | Description   | Pre-condition                             | Input       | Expected output                                   | Actual Output                   | Remark |
|---------|---|---|-------------|---|---------------------------------|--------|
| TC-1    | Simple Addition                                     | Calculator is on                          | 14 + 2      | 16  | 16                              | Pass   |
| TC-2    | Simple Subtraction                                  | Calculator is on                          | 14 – 2      | 12  | 12                              | Pass   |
| TC-3    | Simple Multiplication                               | Calculator is on                          | 12 * 12     | 144   | 144                             | Pass   |
| TC-4    | Simple Division                                     | Calculator is on                          | 36/6        | 6   | 6                               | Pass   |
| TC-5    | Test <- to clear the last enter number or operation | Some number or operator are on the screen | <- button   | Should clear last enter number                    | Clears last entered number      | Pass   |
| TC-6    | Test C to clear input text                          | Input text on the screen                  | C button    | Clears input screen                               | Input screen was cleared        | Pass   |
| TC-7    | Test AC to clear the input & previous result        | Input & previous results on the screens   | AC button   | Clear entered input screen & also previous result | Entire input screen was cleared | Pass   |
| TC-8    | Addition of negative                                | Calculation is on                         | -14+(-12)   | -26   | -26                             | Pass   |
| TC-9    | Subtraction of Negative                             | Calculation is on                         | -14-(-12)   | -2  | -2                              | Pass   |
| TC-10   | Multiplication of negative                          | Calculation is on                         | -14 * (-12) | 168   | 168                             | Pass   |
| TC-11   | Division of negative                                | Calculation is on                         | -14/(-12)   | 1.1667  | 1.166667                        | Pass   |
| TC-12   | Division by zero                                    | Calculation is on                         | 1/0         | Math error  | “Can’t divide by zero”          | Pass   |

**Conclusion :**

Thus, I have completely write test cases & their design of simple calculator application.