The Do-While Loop :-

Syntax :

```
Initialization-expression;
do {
        loop-body;
          update-expression;
    } while (condition-expression);
```

Example :

```
char ch='a';
do {
        cout<<endl<<ch;
        ch++;
    } while (ch<='z');
```

This snippet prints the characters 'a' to 'z' in different lines.

Note :

Unlike the For and While loops, Do-While is an *exit-controlled* loop, i.e., it checks the condition at the end of the loop after executing its loop-body atleast once. This means that a Do-While loop always executes atleast once, even if the condition evaluates to false initially.

The most common use of the Do-While loop is in MENU SELECTION PROGRAMS, where the menu is displayed atleast once. Then, depending upon the user's response, it is either repeated or terminated.

Practice Problem :

1. Write a code to display a menu regarding the different operations which can be performed on a rectangle and perform according to the user's response. The loop should keep on running until the user presses the exit option.

Hint :

The menu can be like :

Rectangle Menu

1. Area
2. Perimeter
3. Diagonal
4. Exit

Enter your choice : 1

Enter length and breadth : 3 4

Area : 6

Press any key to continue…

Rectangle Menu :

1. Area
2. Perimeter
3. Diagonal
4. Exit

4

Observation : So, it is clear from the above problem that for and while loops cannot be used in such cases. That is because we will need to display the menu at least once, even if the user chooses to exit in the very first case. If we try to implement the same program using for or while, we will be trapped if the user enters 4 in the very first case. Because then, the menu will never be displayed as the code will directly exit.