Lecture 23

While loop & do-while loop

While loop:- This loop will continue executing while a particular condition is true.

- It will keep executing as long as the given condition is true

Syntax:-

```
while (condition) {
    // code to execute as long as condition is true
}
```

→ condition → it should always be a boolean expression.

→ It is generally used when number of iterations are not known in advance.

Que. print numbers from 1 to 5 using while loop

```
int i = 1;
while (i <= 5) {
    System.out.println(i);
    i++;
}
```

initialization
termination condition
increment / Decrement

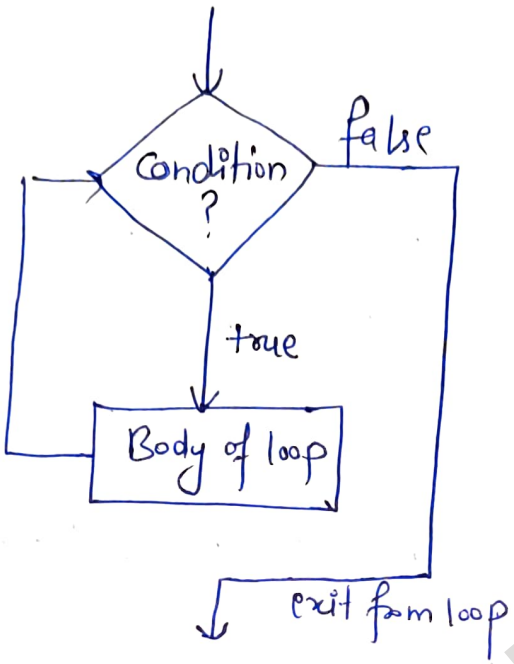
In this case if we don't do i++ then loop would be infinite loop

In while it is mandatory to specify the condition / boolean expression otherwise we will get compilation error.
But initialization & update expressions are optional.

Que: Print numbers until user enters a negative number.

```
Scanner scanner = new Scanner(System.in);
System.out.println("Enter a number:");
while (true) {
    int num = scanner.nextInt();
    if (num < 0) {
        break;
    }
    System.out.println(num);
}
```

while loop is entry controlled loop



Use case :-

(1) When you want to read data from files until there is no more data available

(2) To check input validation,

e.g:-
```
int num = scanner.nextInt();
while( num<=2 || num>=10) {
    System.out.println("Enter a valid number!");
    num = scanner.nextInt();
}
System.out.println("thanks");
```

(3) Checking Password / Authentication until success

Practice Time

```
int n=1;
while(n==1){
  System.out.println(n); n++;
}
```

output
1

But in above case if we replace (n==1) with (n=1) then?

```
while(n=1){
  - -
}
```

⇒ output ? will give error because condition should always be boolean expression.

do-while loop :- exit controlled loop

↳ do-while loop executes a block of statements at least once & then continue executing as long as the specified condition is true

→ The main difference between while & do-while loop is that the do-while loop will always run at least once, regardless of the condition, because the condition is evaluated after the loop body executes.

Syntax :-

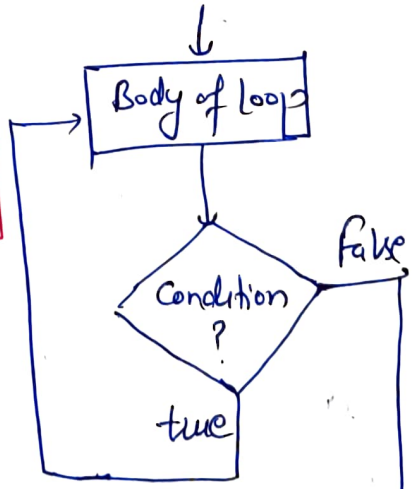
```
do{
  // Code to execute
}while(condition);
```

semicolon is must

↓

it is always a boolean expression, means the condition/expression must evaluate to boolean values.

Flowchart :-



e.g. print numbers from 1 to 5

```
int i = 1;
do {
    System.out.println(i);
    i++;
} while(i<=5);
```

Coding Exercise:-

(i) WAP that prompts the user to enter an integer between 1 and 50, inclusive. If the user enters a value outside this range, display an error message and prompt them again. It should continue to ask for input until the user provides a valid number within the specified range [use do-while loop]

Output! - (Example)

```
Enter a number between 1 and 50 : 51
Invalid input. Valid range is between 1 and 50.
Enter a number between 1 and 50 : -1
Invalid input. Valid range is between 1 and 50.
Enter a number between 1 and 50 : 10
Thank you. You entered 10 which is a valid number.
```

NOTE:- do-while loop is used when we want to execute the loop body atleast once without checking the condition. In Menu-driven programs.

(2) Write a program that provides a menu-driven calculator with the following options:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

The program should:

- Display a menu with options listed above.
- prompt the user to enter a choice between 1 and 5.
- Based on users choice, perform the selected arithmetic operation on 2 numbers entered by the user.
- Display an error message if the user attempts to divide by zero.
- Use a do-while loop to keep displaying the menu until the user selects option 5 to exit.

(3) Write a Program that prompts the user to enter a password. The program should:

1. Ask the user to input a password.
2. Verify if the entered password matches a predefined password (e.g. Password123)
3. If the password is incorrect, display an error message and allow the user to try again.
4. Use a do-while loop to continue prompting the user until the correct password is entered.
5. Display a message "Access Granted!" once the user enters the correct password.