=> Jump Statements :-

-> Jump statements are also known as Transfer Statements

-> Jump statements are used to skip some statements inside the loop or used to terminate the loop immediately without checking the condition

-> Examples are break, continue, return

1. break :-

-> It is used to terminate the loop

-> Whenever we use break statement, then loop gets teminated without checking the condition and first statement after the loop will be executed

-> Use :-

1. Used to terminate the loop
2. Used to terminate the switch sequence

-> Cases :-

1. There should not be any statement just after the break statement
2. If there is inner for loop and we are using break statement inside inner for loop, then it will break only inner for loop

-> If we want to terminate nested loop according to our needs then we can use labelled break statement.

```
public class Demo {
    public static void main(String[] args) {
        for(int i=1  ; i<=10  ; i++){
            System.out.println(i);
            if (i == 5) {
                break;
            }
        }
    }
}
```

Labbeled Break :

```
public class Demo {
    public static void main(String[] args) {
        outer:
        for(int i=1 ; i<=5 ; i++){
            System.out.println("i: " + i);
            inner:            //optional
            for(int j=1 ; j<=5 ; j++){
                if(i==3){
                    break outer ;       //if inner is written, then it will break from inner loop only
                }
                System.out.println(": " + j);
            }
        }
    }
```

2. continue :-

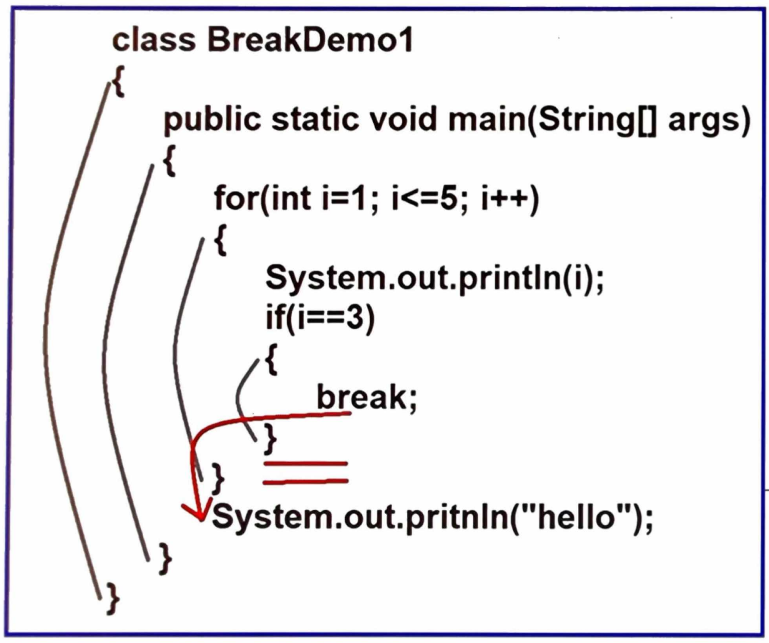
-> continue is used to skip the current iteration in the loop

-> Case :-

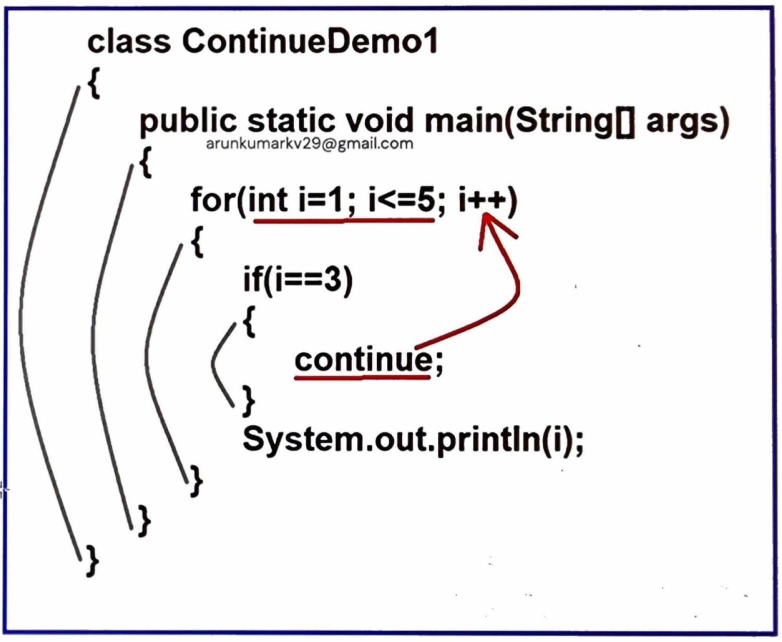
1. There should not be any statement after continue statement

-> If there are nested for loop and we want to skip the current iteration for outer for loop then we can use labelled continue statement

```
public class Demo {
    public static void main(String[] args) {
        for(int i=1 ; i<=10 ; i++){
            if(i==5){
                continue ;
            }
            System.out.println(i);
        }
    }
}
```



```
class BreakDemo1
{
    public static void main(String[] args)
    {
        for(int i=1; i<=5; i++)
        {
            System.out.println(i);
            if(i==3)
            {
                break;
            }
        }
        System.out.pritnln("hello");
    }
}
```



```
class ContinueDemo1
{
    public static void main(String[] args)
    {
        for(int i=1; i<=5; i++)
        {
            if(i==3)
            {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

3. return

-> return is used to exit the method with or without a value

-> return can be used in the method by two

types :-

1. method returning value
2. method not returning any value