LOOP CONTROL

TABLE OF CONTENTS

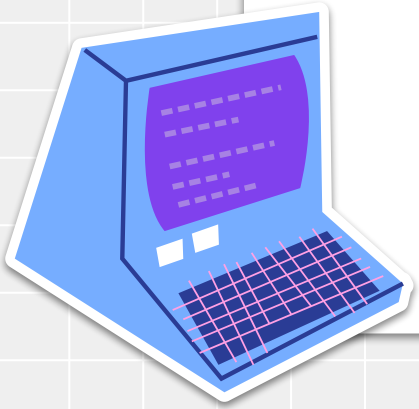
TEAM 9!

SEY HAK, HENG LEABHENG
EAB PISEY, LONG CHANLEAB
LIM KIMHOUNG

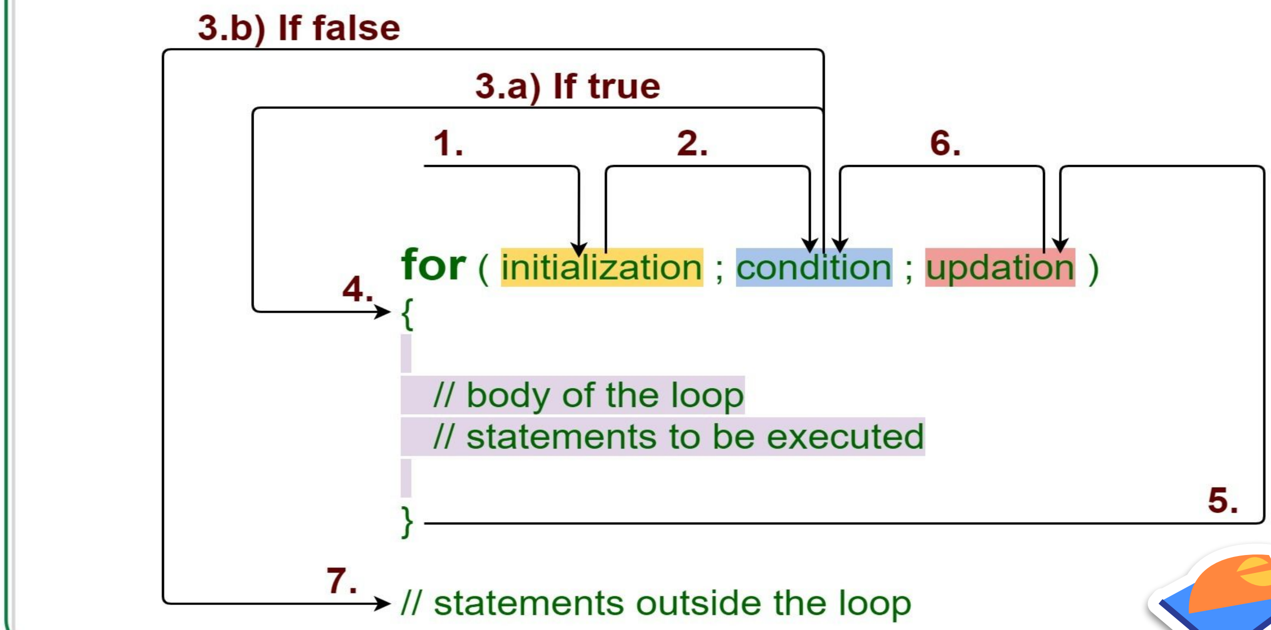
01

FOR LOOP

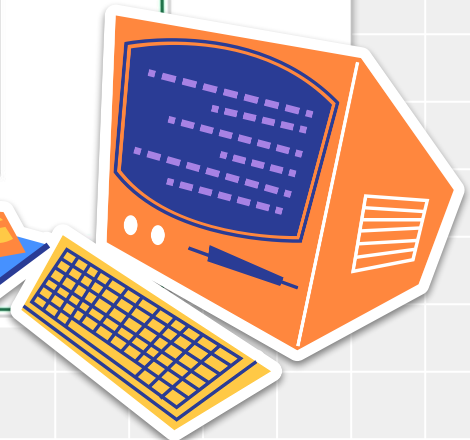
We use for loop only **when we exactly know the number of times we want to execute a block of code**



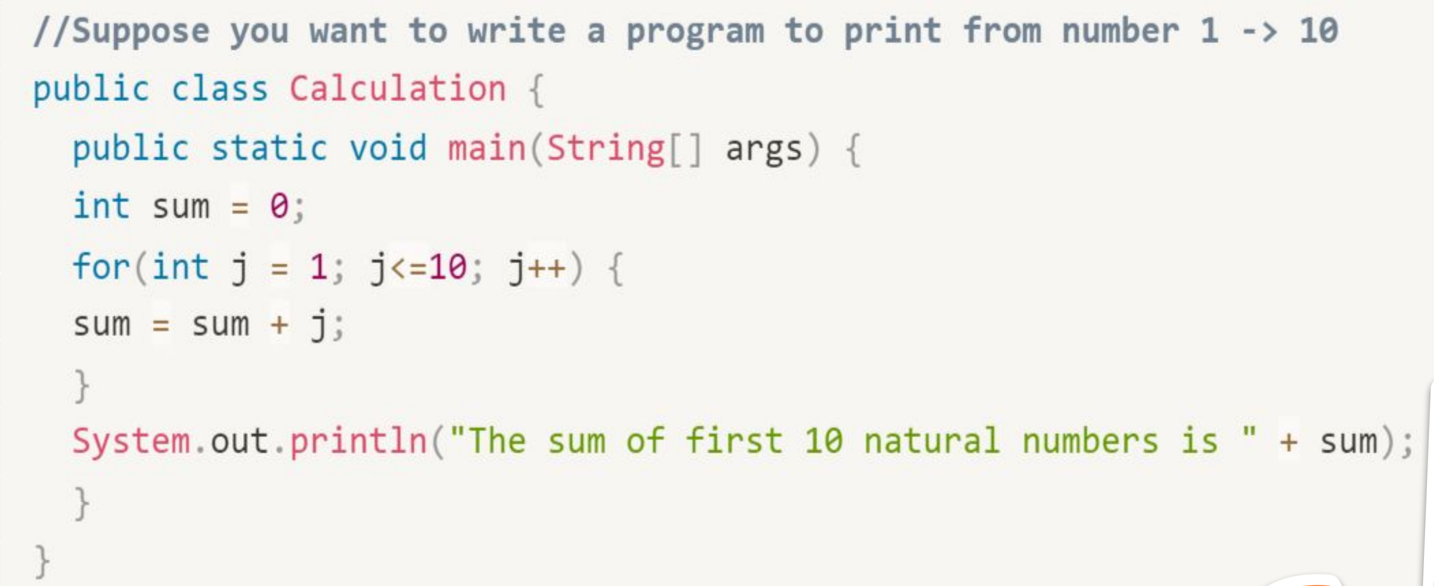
syntax



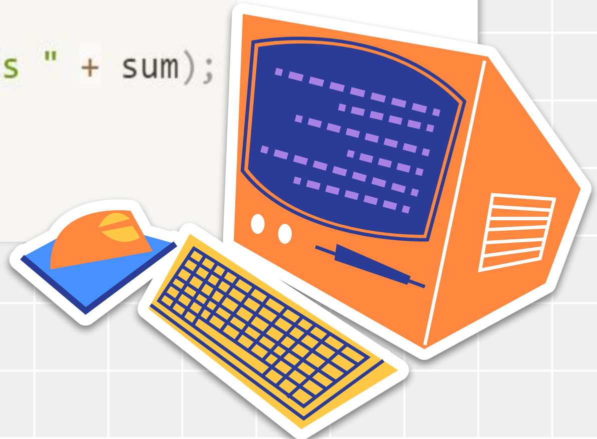
```
for ( initialization ; condition ; updation )
{

    // body of the loop
    // statements to be executed

}
// statements outside the loop
```



Example



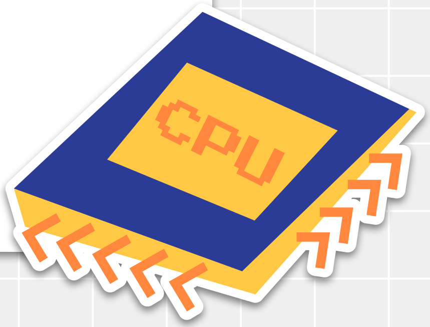
```
//Suppose you want to write a program to print from number 1 -> 10
public class Calculation {
  public static void main(String[] args) {
  int sum = 0;
  for(int j = 1; j<=10; j++) {
  sum = sum + j;
  }
  System.out.println("The sum of first 10 natural numbers is " + sum);
  }
}
```



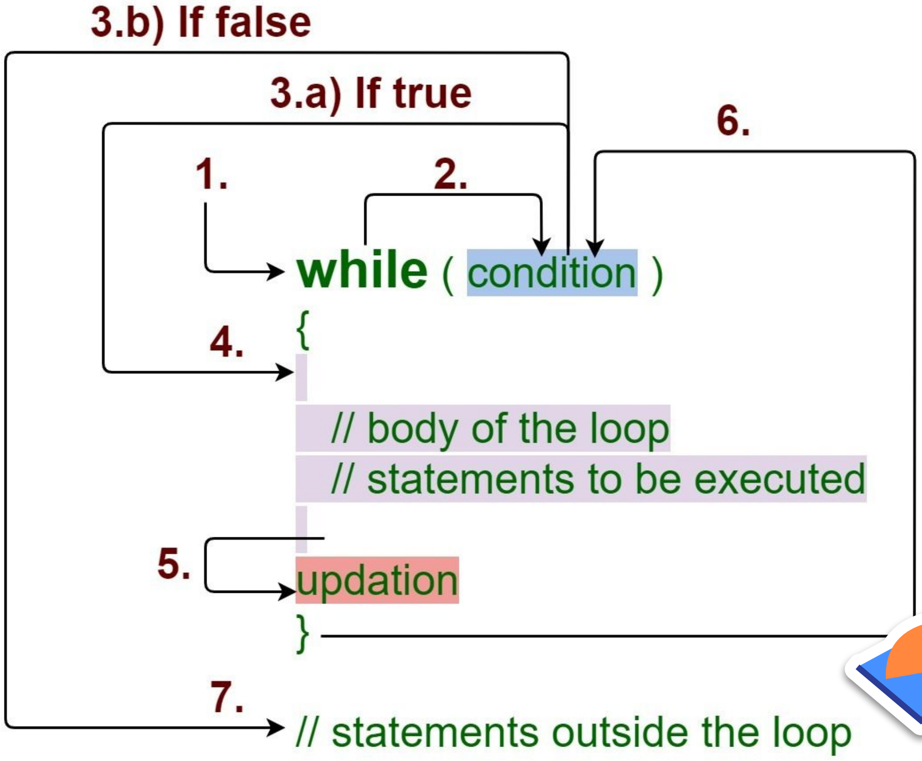
02

While Loop

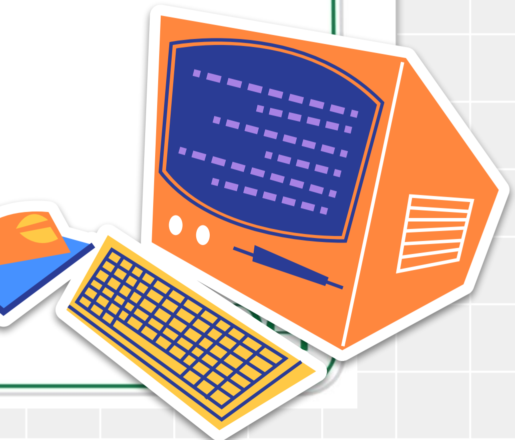
we use while loop **If we don’t know the number of execution in a block of code.**



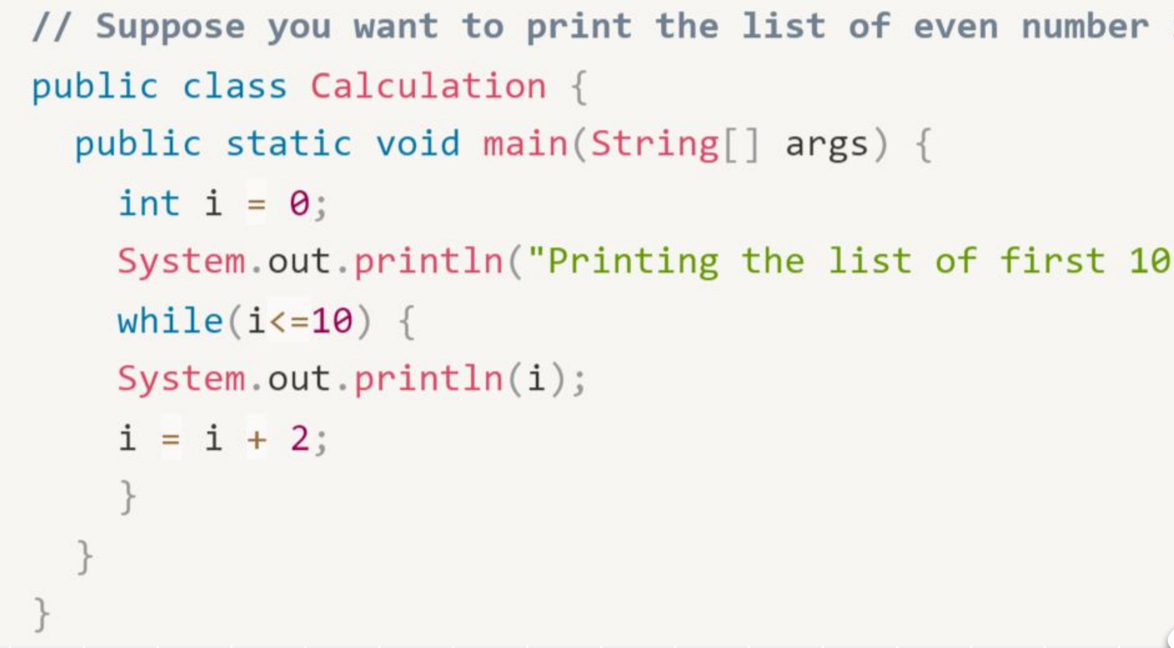
syntax



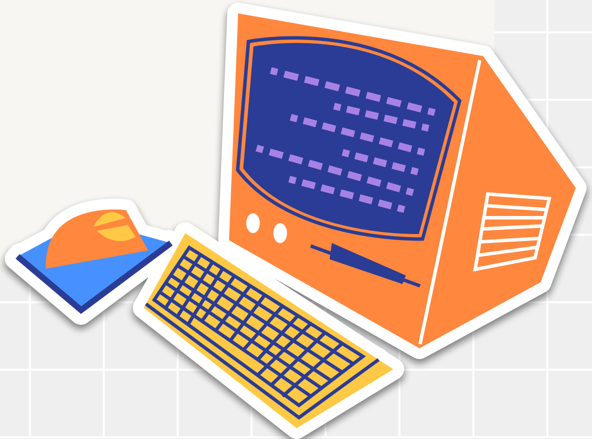
```
while ( condition )
{

   // body of the loop
   // statements to be executed

updation
}
// statements outside the loop
```



Example



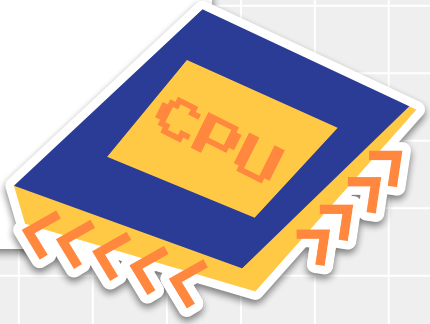
```
// Suppose you want to print the list of even number in range from 1 -> 10
public class Calculation {
  public static void main(String[] args) {
    int i = 0;
    System.out.println("Printing the list of first 10 even numbers \n");
    while(i<=10) {
    System.out.println(i);
    i = i + 2;
    }
  }
}
```



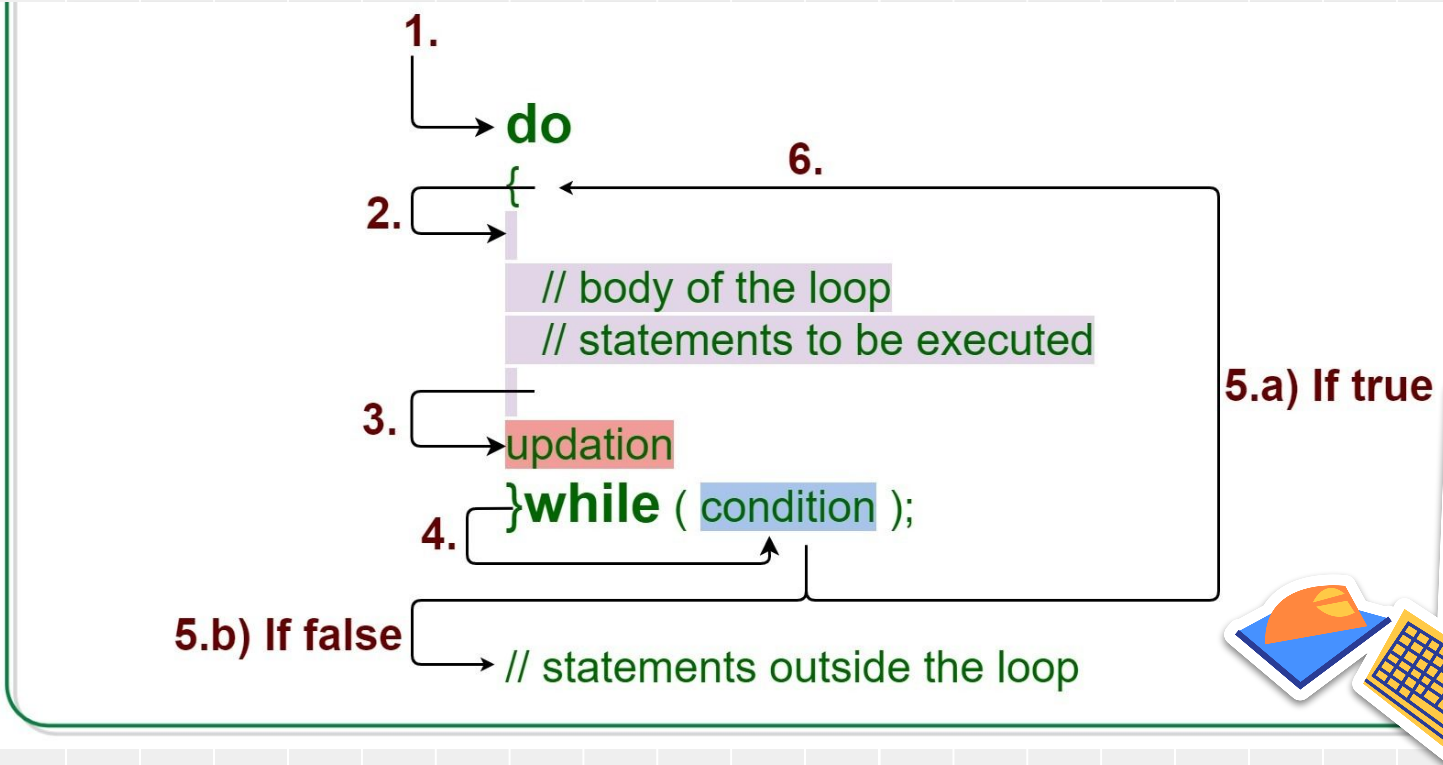
03

DO-WHILE LOOP

do-while loop **checks the condition at the end of the loop after executing the block instruction in do**. When the number of iteration is **not known and we have to execute the loop at least once**, we can use do-while loop.

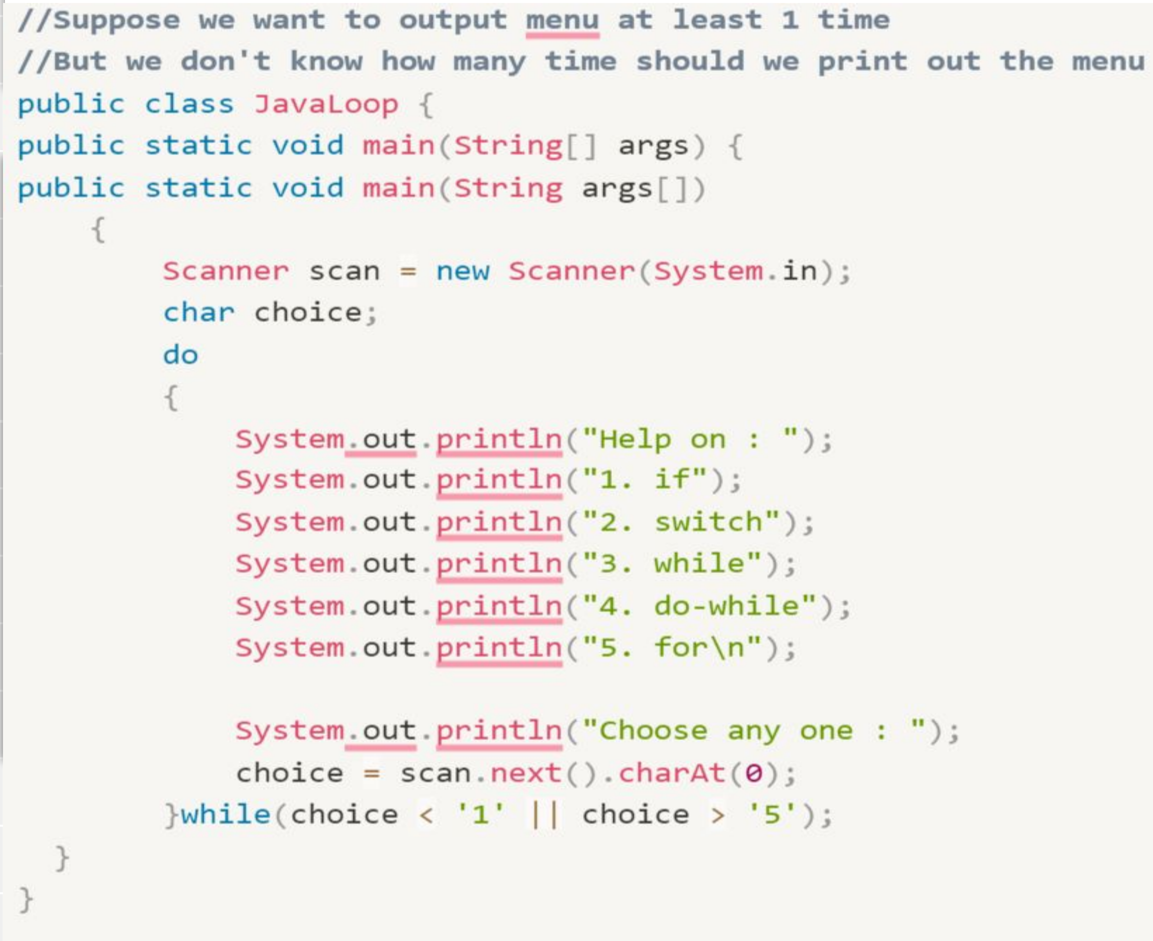


syntax

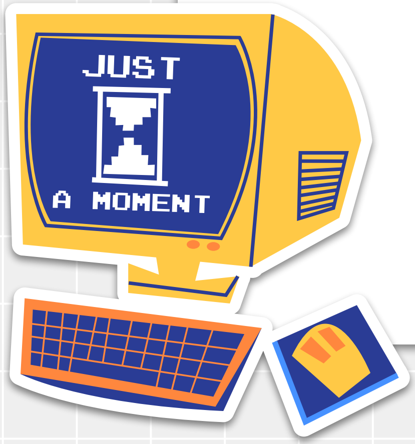


```
do
{

  // body of the loop
  // statements to be executed

updation
}while ( condition );

// statements outside the loop
```

Example



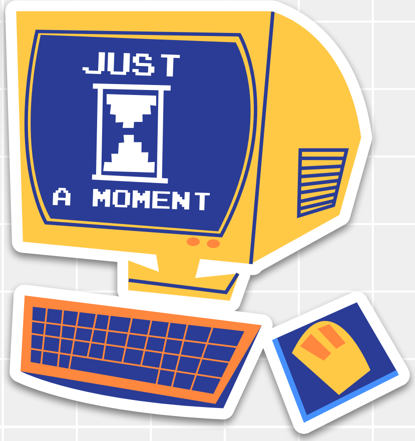
```
//Suppose we want to output menu at least 1 time
//But we don't know how many time should we print out the menu
public class JavaLoop {
public static void main(String[] args) {
public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        char choice;
        do
        {
            System.out.println("Help on : ");
            System.out.println("1. if");
            System.out.println("2. switch");
            System.out.println("3. while");
            System.out.println("4. do-while");
            System.out.println("5. for\n");

            System.out.println("Choose any one : ");
            choice = scan.next().charAt(0);
        }while(choice < '1' || choice > '5');
  }
}
```

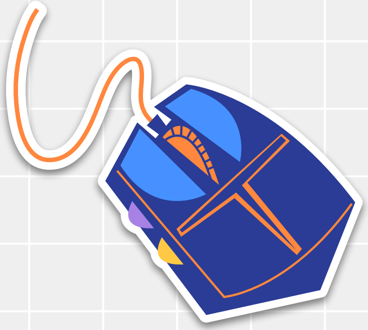


04

LOOP CONTROL STATEMENT

Programs usually run step to step but if you want a program **to stop at a particular step or skip at a particular step** we use loop control statement

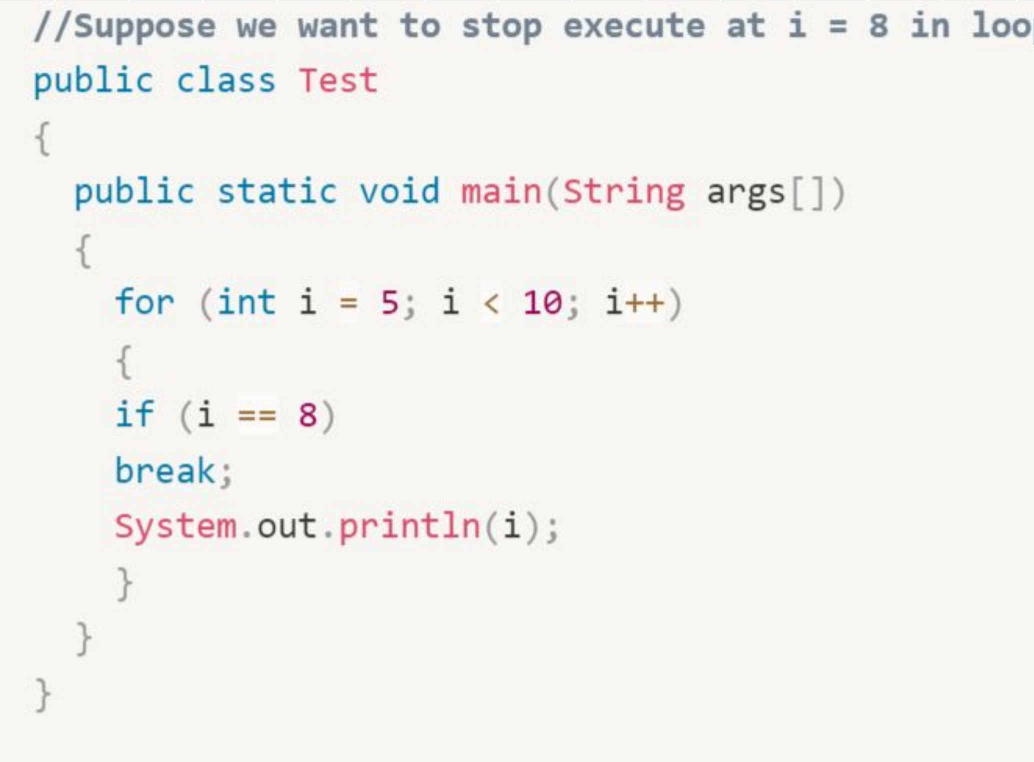




1. Break Statement

The break statement in java **is used to terminate a loop and break the current flow of the program.**

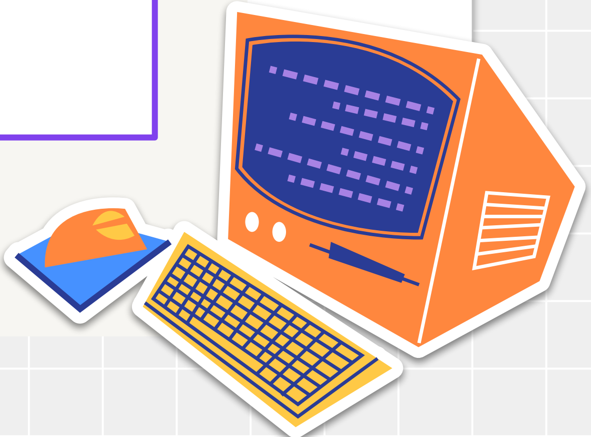
Example

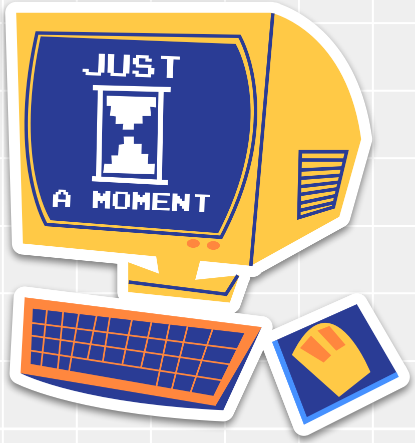


```
//Suppose we want to stop execute at i = 8 in loop range from 5 -> 10
public class Test
{
  public static void main(String args[])
  {
    for (int i = 5; i < 10; i++)
    {
    if (i == 8)
    break;
    System.out.println(i);
    }
  }
}
```

Output:

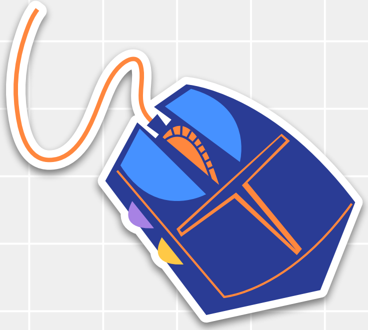
```
5
6
7
```



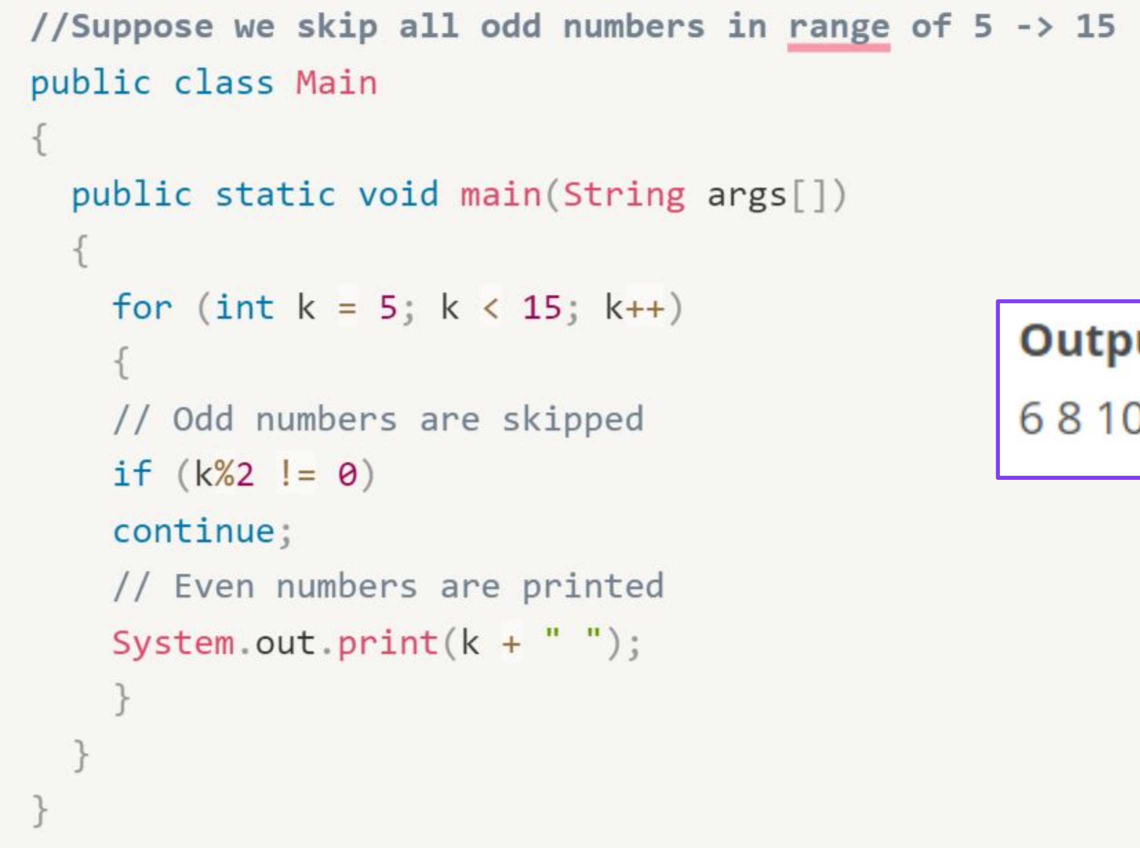


2. CONTINUOUS STATEMENT

This statement **continues the current flow of the program and skips a part of the code** at the specified condition



Example



```java
//Suppose we skip all odd numbers in range of 5 -> 15
public class Main
{
  public static void main(String args[])
  {
    for (int k = 5; k < 15; k++)
    {
    // Odd numbers are skipped
    if (k%2 != 0)
    continue;
    // Even numbers are printed
    System.out.print(k + " ");
    }
  }
}
```

Output:

```
6 8 10 12 14
```

THANK YOU!