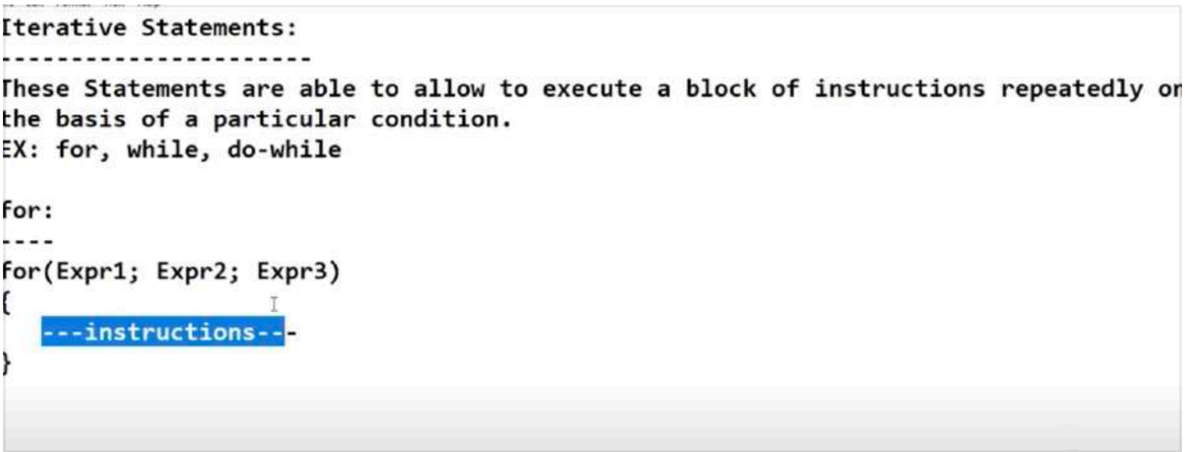
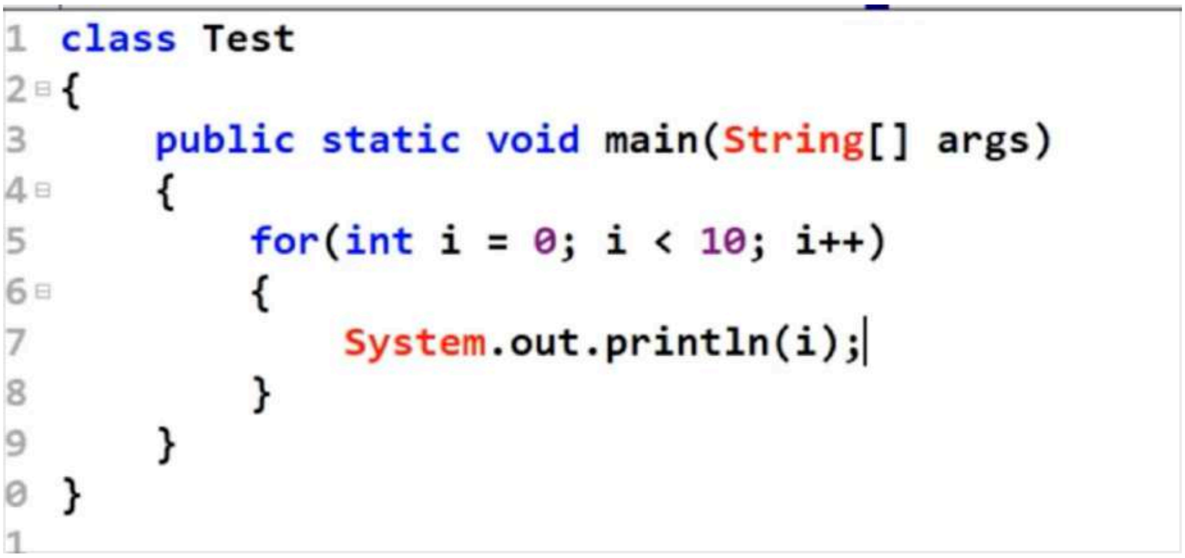
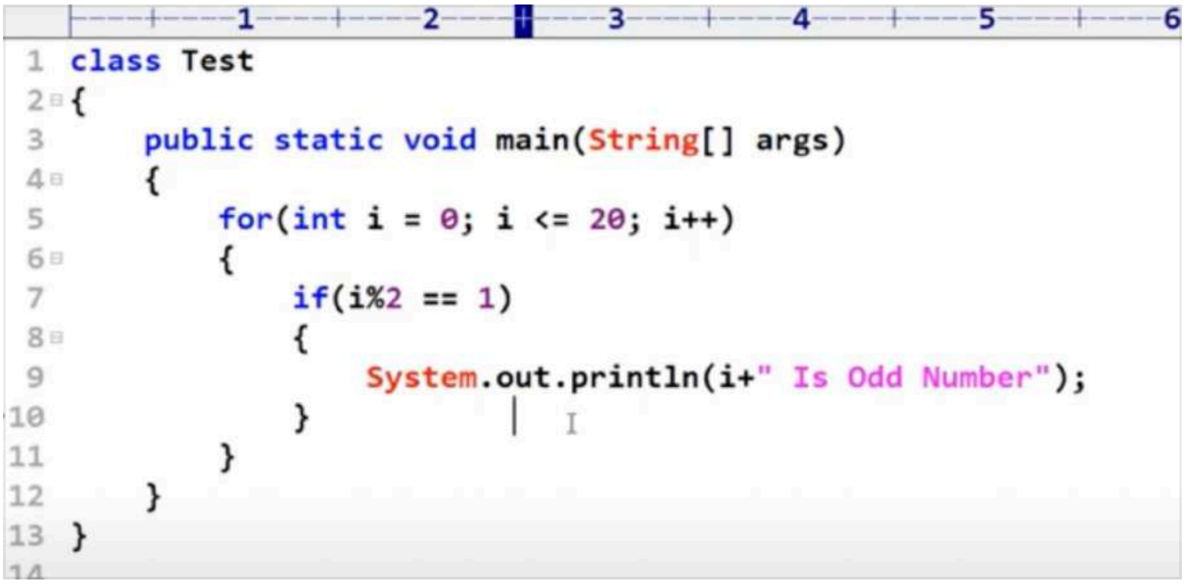


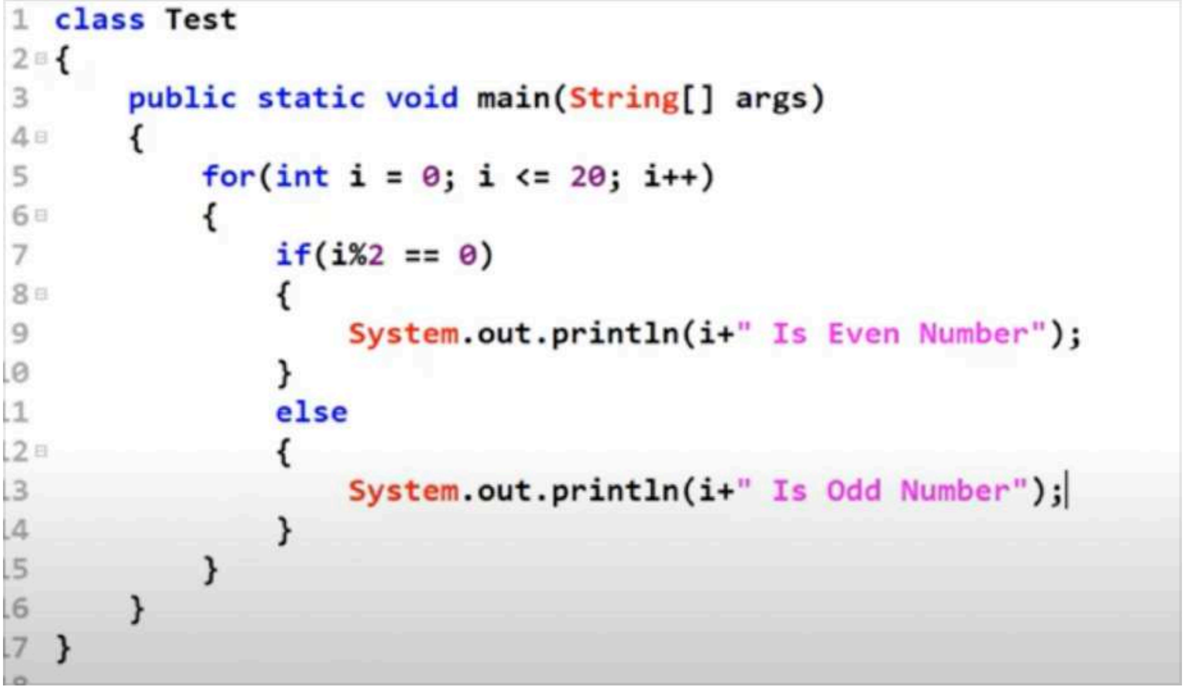
```
Iterative Statements:
----------------------
These Statements are able to allow to execute a block of instructions repeatedly on
the basis of a particular condition.
EX: for, while, do-while

for:
----
for(Expr1; Expr2; Expr3)
{
   ---instructions---
}
```



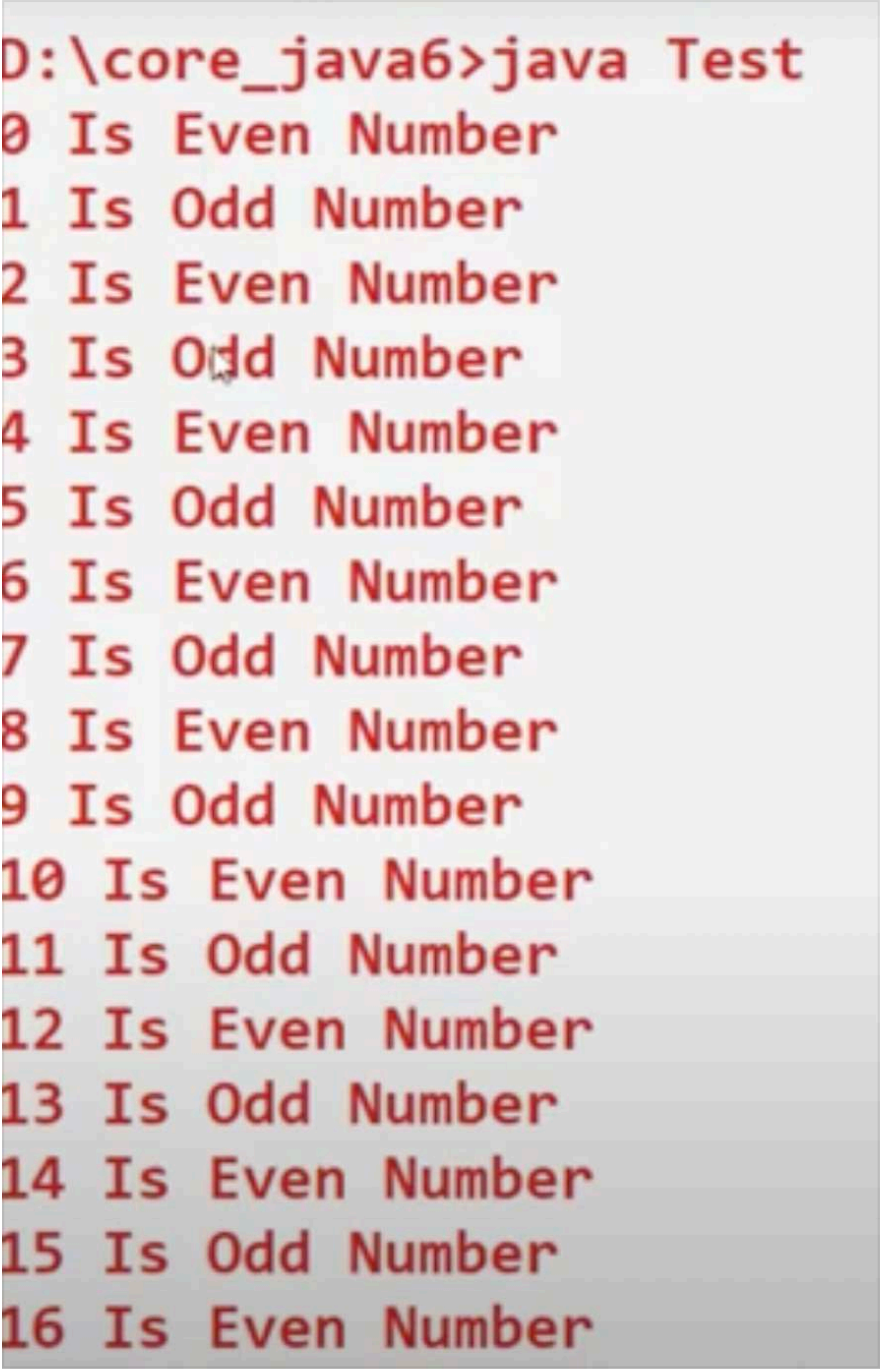
```java
class Test
{
	public static void main(String[] args)
	{
		for(int i = 0; i < 10; i++)
		{
			System.out.println(i);
		}
	}
}
```



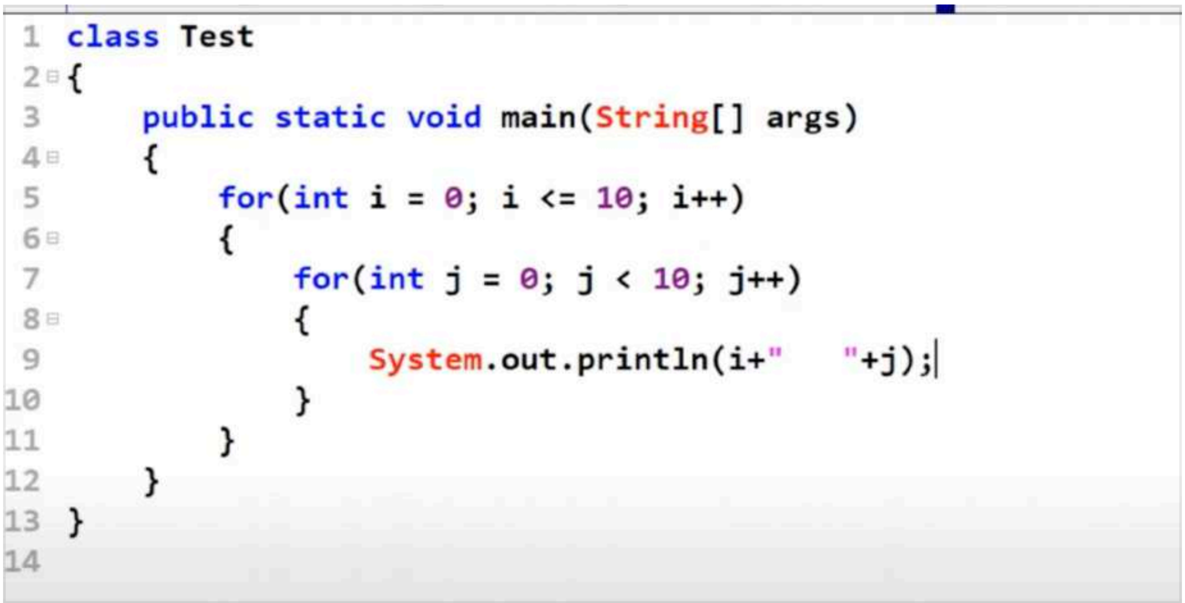
```java
class Test
{
	public static void main(String[] args)
	{
		for(int i = 0; i <= 20; i++)
		{
			if(i%2 == 1)
			{
				System.out.println(i+" Is Odd Number");
			}
		}
	}
}
```



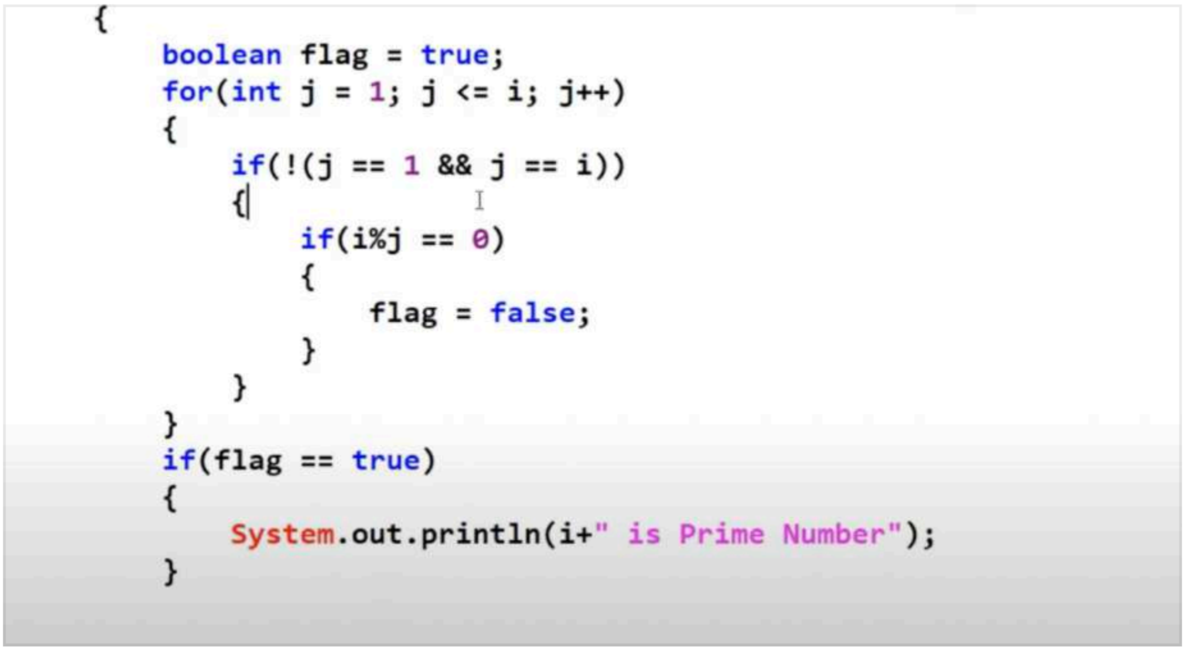
```
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i <= 20; i++)
        {
            if(i%2 == 0)
            {
                System.out.println(i+" Is Even Number");
            }
            else
            {
                System.out.println(i+" Is Odd Number");
            }
        }
    }
}
```



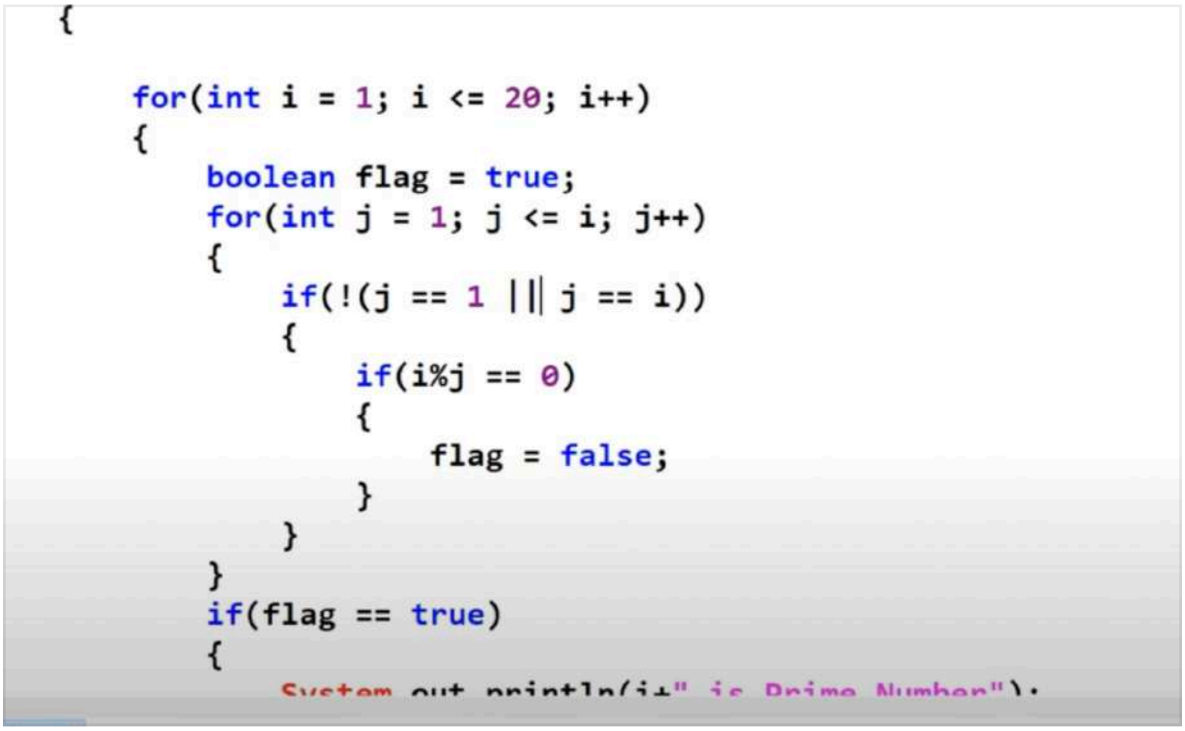
```
D:\core_java6>java Test
0 Is Even Number
1 Is Odd Number
2 Is Even Number
3 Is Odd Number
4 Is Even Number
5 Is Odd Number
6 Is Even Number
7 Is Odd Number
8 Is Even Number
9 Is Odd Number
10 Is Even Number
11 Is Odd Number
12 Is Even Number
13 Is Odd Number
14 Is Even Number
15 Is Odd Number
16 Is Even Number
```



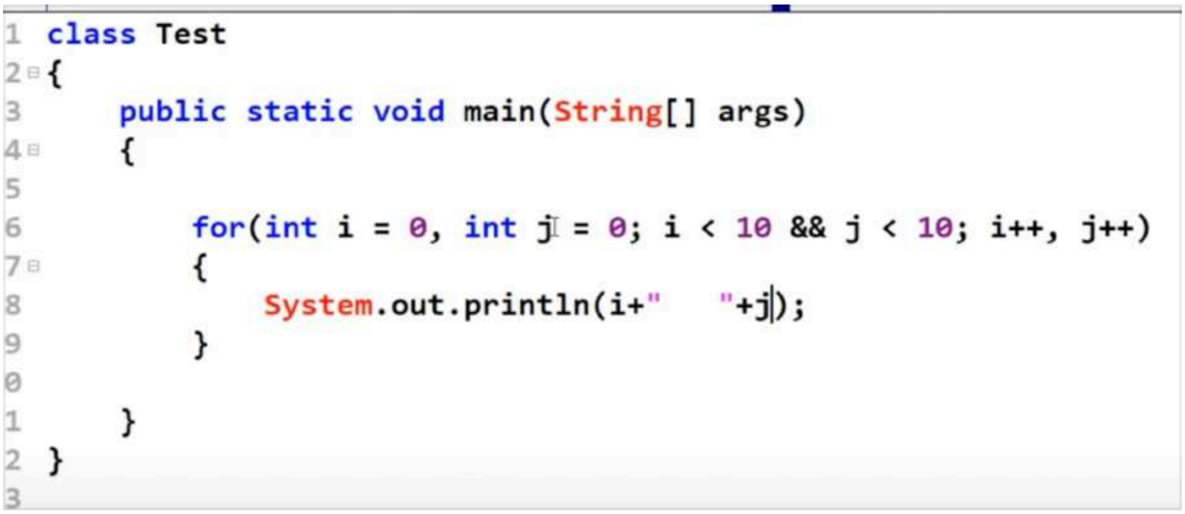
```
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i <= 10; i++)
        {
            for(int j = 0; j < 10; j++)
            {
                System.out.println(i+"    "+j);
            }
        }
    }
}
```



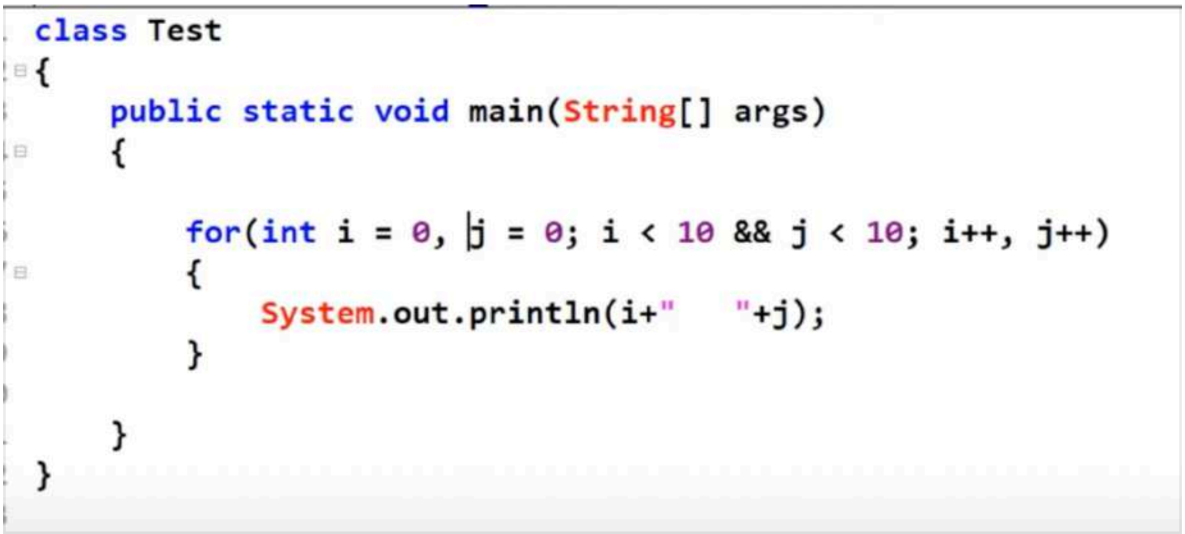
```
{
    boolean flag = true;
    for(int j = 1; j <= i; j++)
    {
        if(!(j == 1 && j == i))
        {
            if(i%j == 0)
            {
                flag = false;
            }
        }
    }
    if(flag == true)
    {
        System.out.println(i+" is Prime Number");
    }
```



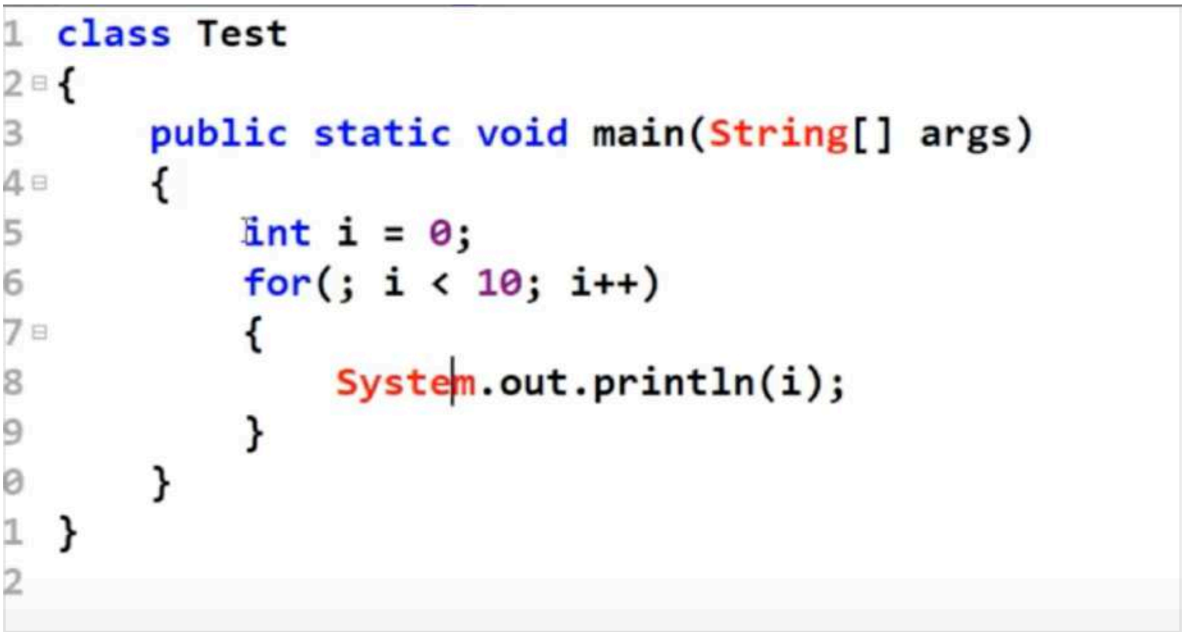
```
    {

        for(int i = 1; i <= 20; i++)
        {
            boolean flag = true;
            for(int j = 1; j <= i; j++)
            {
                if(!(j == 1 || j == i))
                {
                    if(i%j == 0)
                    {
                        flag = false;
                    }
                }
            }
            if(flag == true)
            {
                System.out.println(i+" is Prime Number");
```



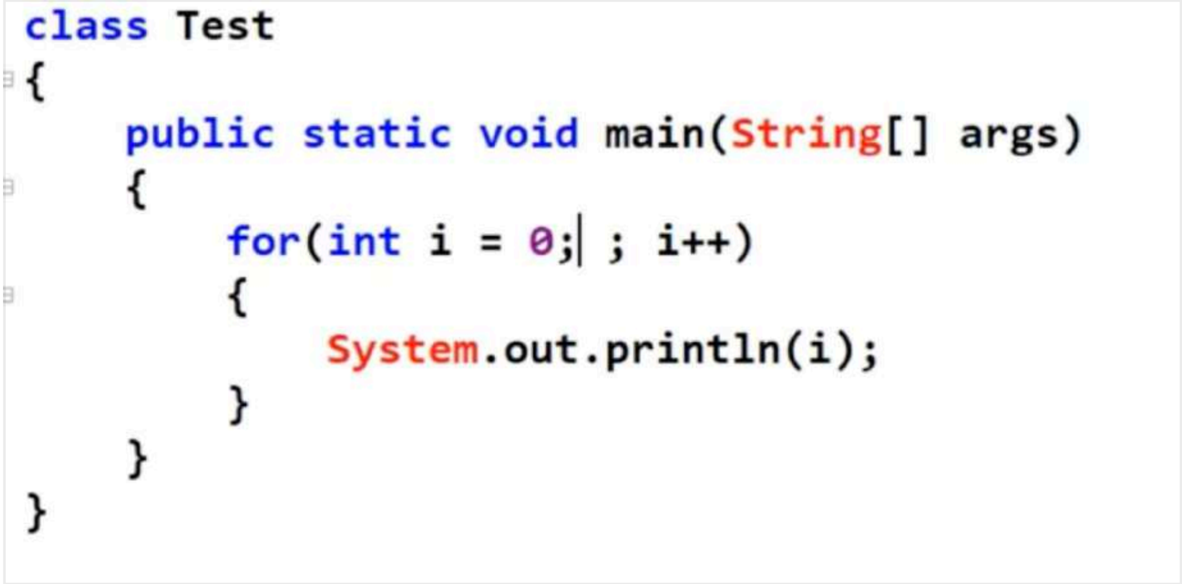
```
class Test
{
    public static void main(String[] args)
    {

        for(int i = 0, int j = 0; i < 10 && j < 10; i++, j++)
        {
            System.out.println(i+"    "+j);
        }

    }
}
```



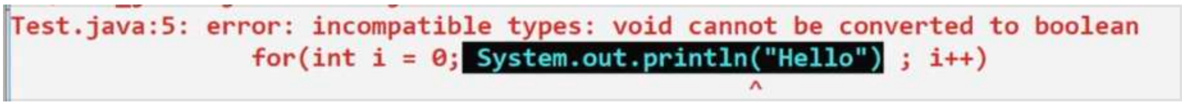
```
class Test
{
    public static void main(String[] args)
    {

        for(int i = 0, j = 0; i < 10 && j < 10; i++, j++)
        {
            System.out.println(i+"    "+j);
        }

    }
}
```



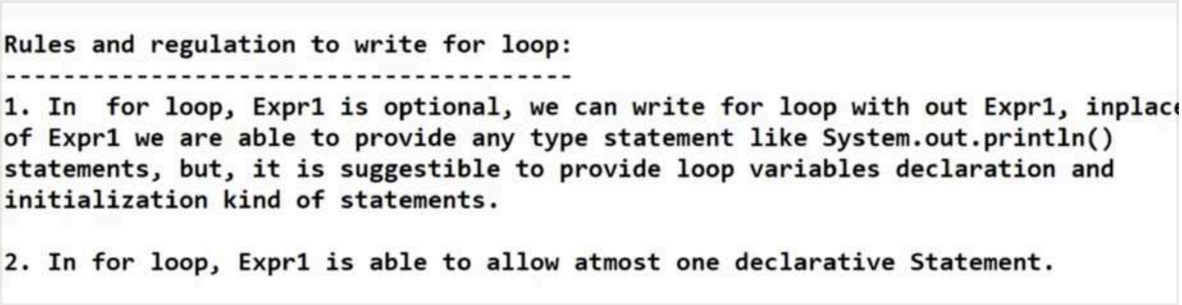
```
class Test
{
    public static void main(String[] args)
    {
        int i = 0;
        for(; i < 10; i++)
        {
            System.out.println(i);
        }
    }
}
```



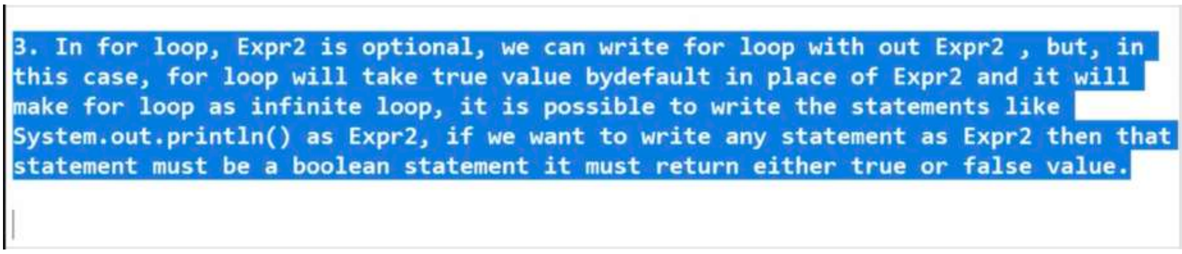
```
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; ; i++)
        {
            System.out.println(i);
        }
    }
}
```



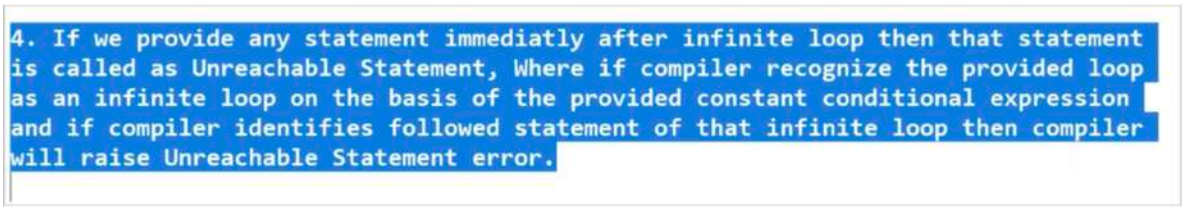
```
Test.java:5: error: incompatible types: void cannot be converted to boolean
                for(int i = 0; System.out.println("Hello") ; i++)
                                                  ^
```



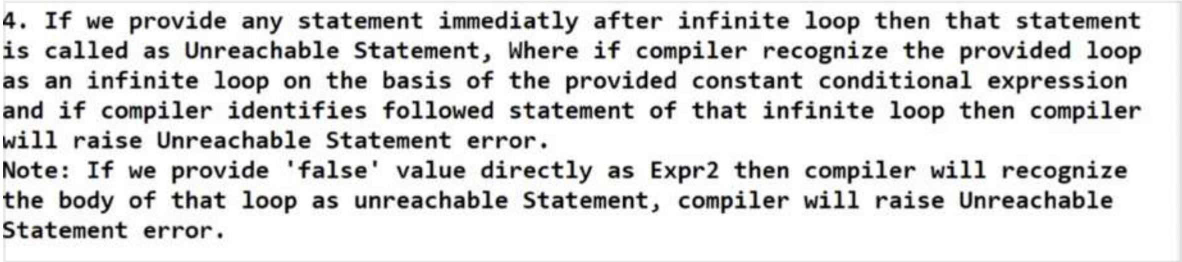
```
Rules and regulation to write for loop:
---------------------------------------
1. In  for loop, Expr1 is optional, we can write for loop with out Expr1, inplace
of Expr1 we are able to provide any type statement like System.out.println()
statements, but, it is suggestible to provide loop variables declaration and
initialization kind of statements.

2. In for loop, Expr1 is able to allow atmost one declarative Statement.
```



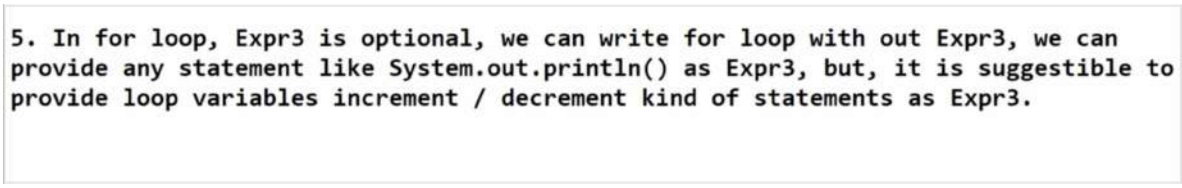
3. In for loop, Expr2 is optional, we can write for loop with out Expr2 , but, in this case, for loop will take true value bydefault in place of Expr2 and it will make for loop as infinite loop, it is possible to write the statements like System.out.println() as Expr2, if we want to write any statement as Expr2 then that statement must be a boolean statement it must return either true or false value.



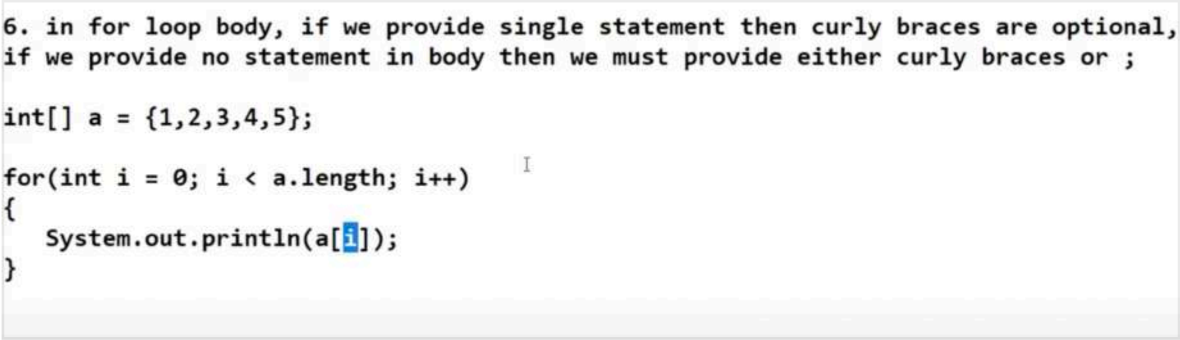
4. If we provide any statement immediatly after infinite loop then that statement is called as Unreachable Statement, Where if compiler recognize the provided loop as an infinite loop on the basis of the provided constant conditional expression and if compiler identifies followed statement of that infinite loop then compiler will raise Unreachable Statement error.



4. If we provide any statement immediatly after infinite loop then that statement is called as Unreachable Statement, Where if compiler recognize the provided loop as an infinite loop on the basis of the provided constant conditional expression and if compiler identifies followed statement of that infinite loop then compiler will raise Unreachable Statement error.
Note: If we provide 'false' value directly as Expr2 then compiler will recognize the body of that loop as unreachable Statement, compiler will raise Unreachable Statement error.

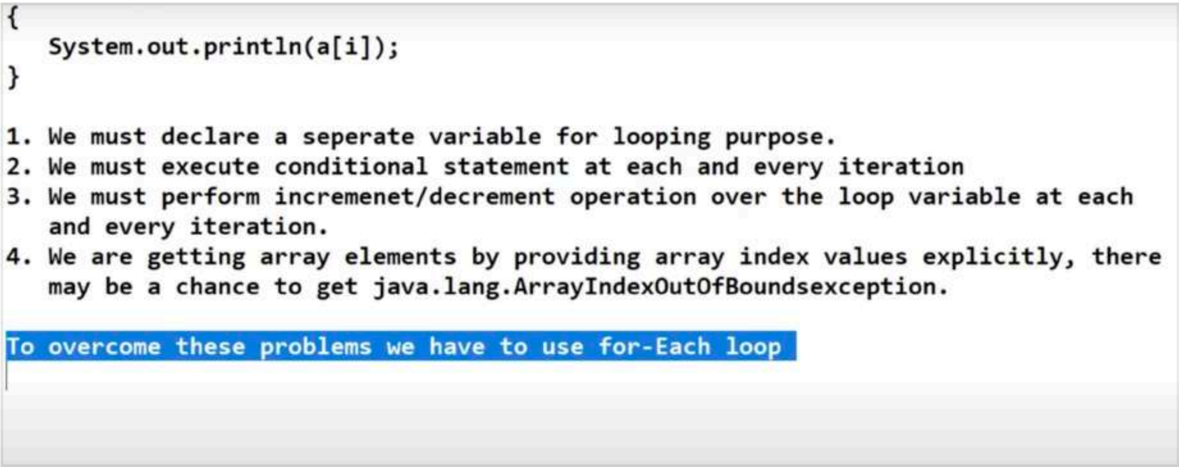


5. In for loop, Expr3 is optional, we can write for loop with out Expr3, we can provide any statement like System.out.println() as Expr3, but, it is suggestible to provide loop variables increment / decrement kind of statements as Expr3.



6. in for loop body, if we provide single statement then curly braces are optional, if we provide no statement in body then we must provide either curly braces or ;

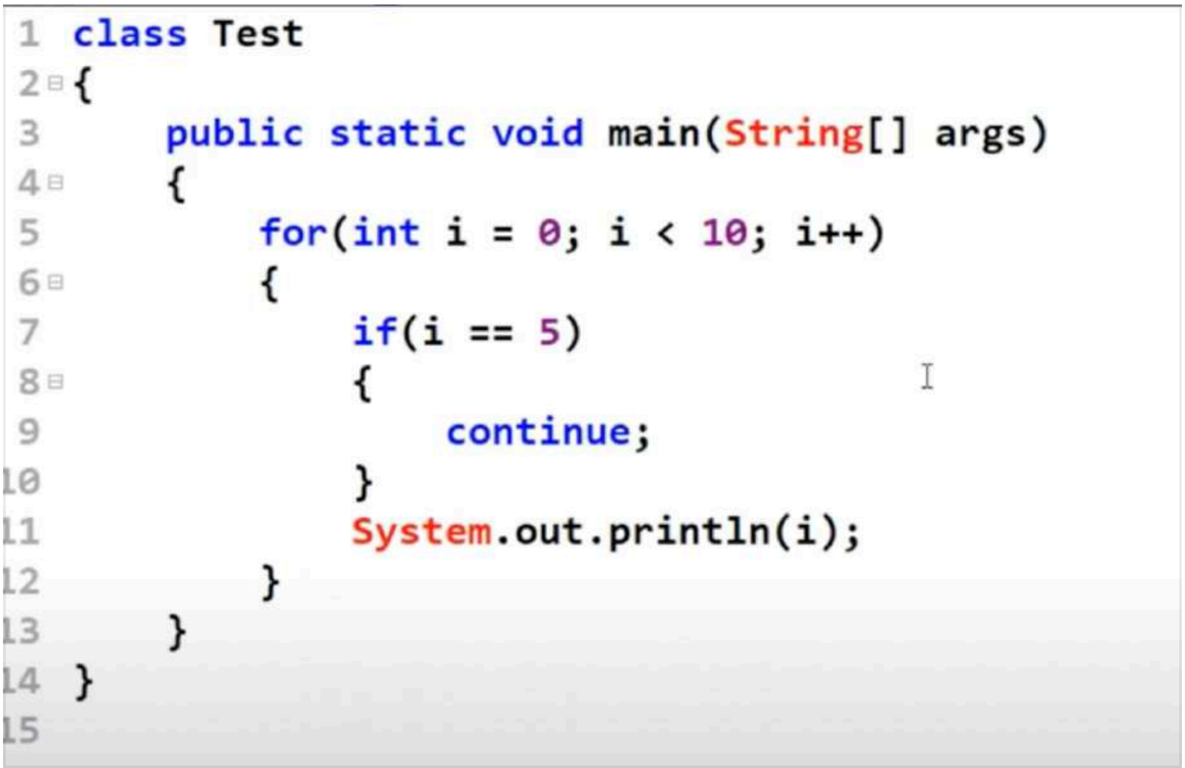
```
int[] a = {1,2,3,4,5};

for(int i = 0; i < a.length; i++)
{
   System.out.println(a[i]);
}
```



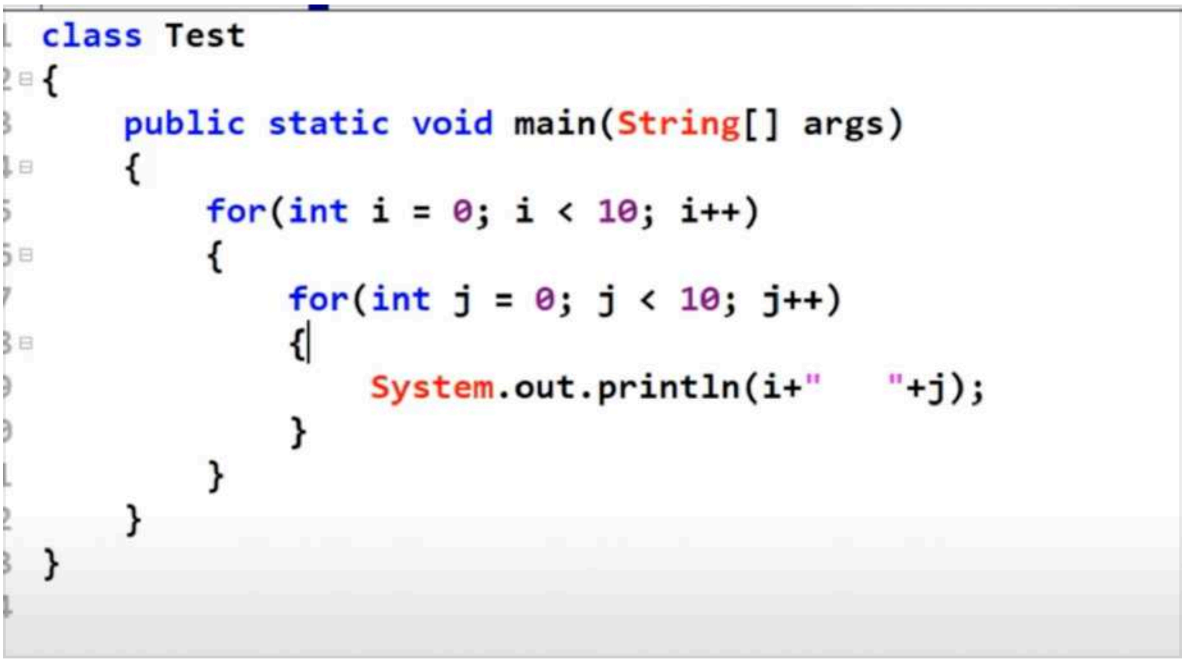
```
{
   System.out.println(a[i]);
}
```

1. We must declare a seperate variable for looping purpose.
2. We must execute conditional statement at each and every iteration
3. We must perform incremenet/decrement operation over the loop variable at each and every iteration.
4. We are getting array elements by providing array index values explicitly, there may be a chance to get java.lang.ArrayIndexOutOfBoundsexception.

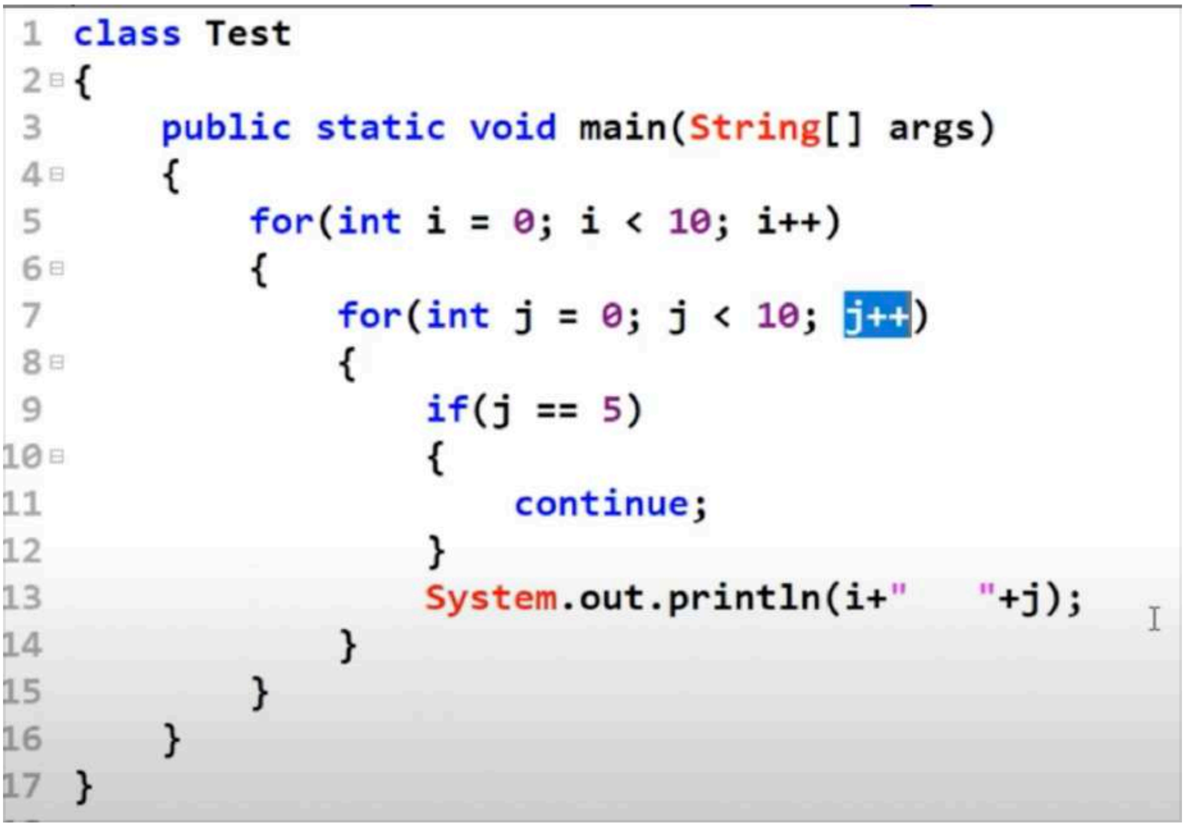
To overcome these problems we have to use for-Each loop



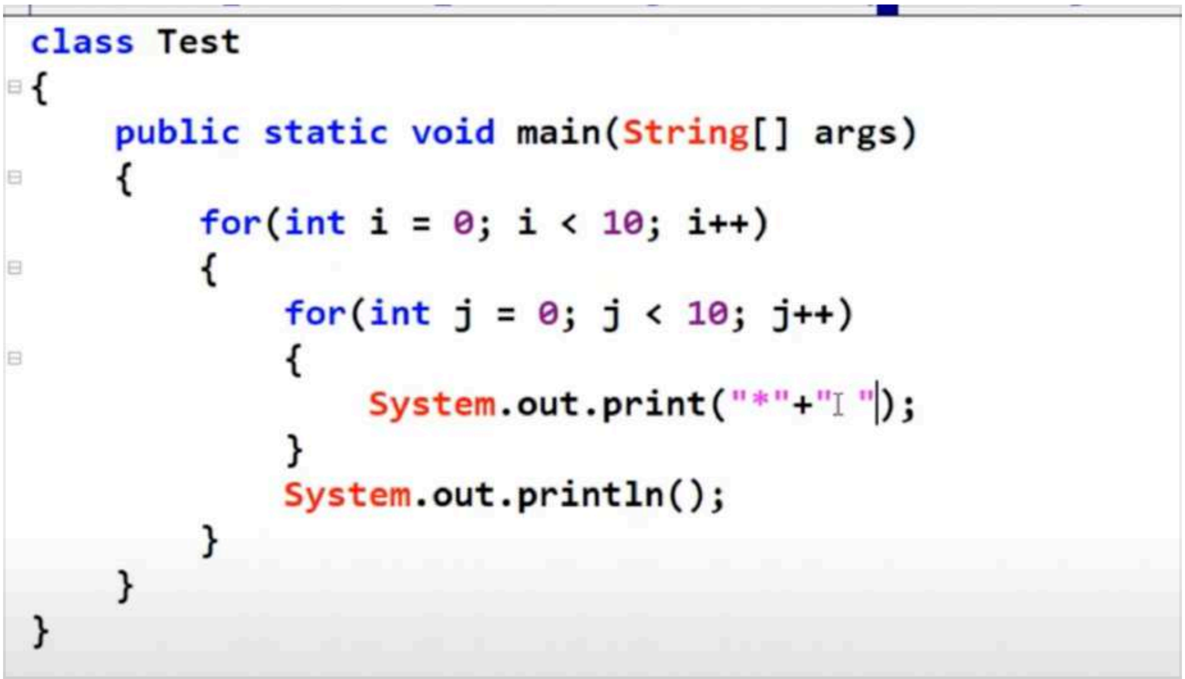
```java
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 10; i++)
        {
            if(i == 5)
            {
                continue;
            }
            System.out.println(i);
        }
    }
}
```



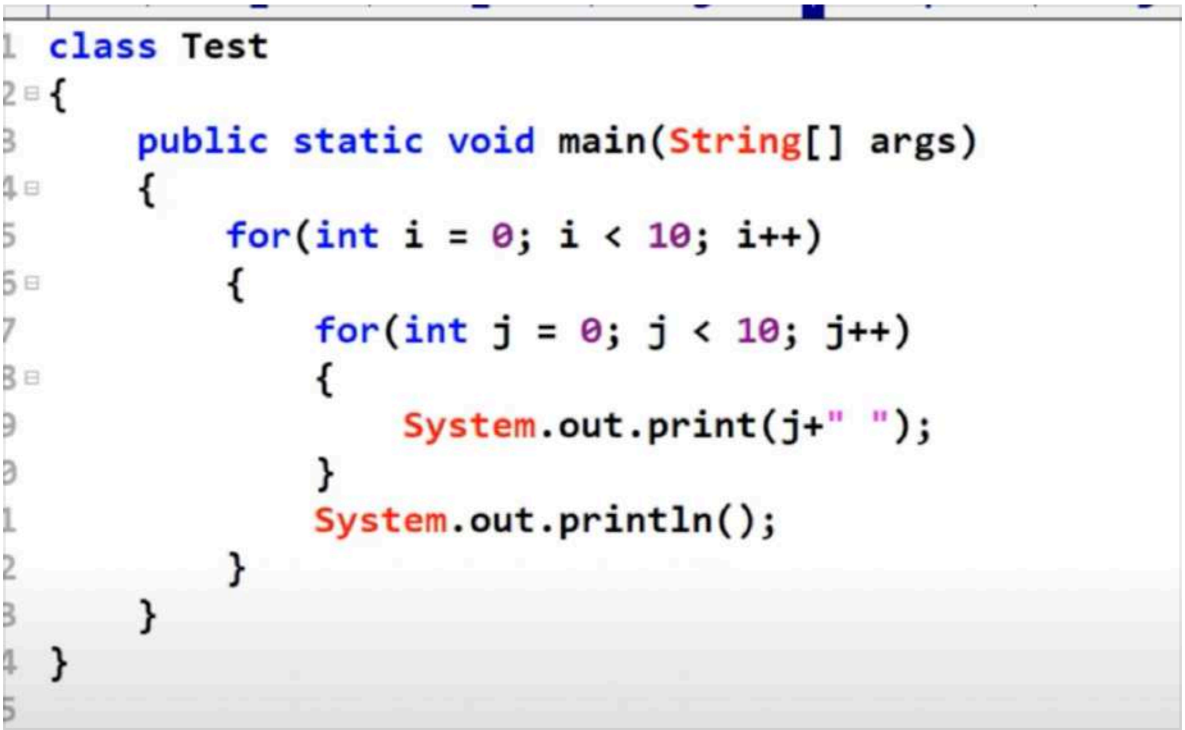
```java
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 10; i++)
        {
            for(int j = 0; j < 10; j++)
            {
                System.out.println(i+"    "+j);
            }
        }
    }
}
```



```java
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 10; i++)
        {
            for(int j = 0; j < 10; j++)
            {
                if(j == 5)
                {
                    continue;
                }
                System.out.println(i+"    "+j);
            }
        }
    }
}
```



```java
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 10; i++)
        {
            for(int j = 0; j < 10; j++)
            {
                System.out.print("*"+"  ");
            }
            System.out.println();
        }
    }
}
```



```
class Test
{
    public static void main(String[] args)
    {
        for(int i = 0; i < 10; i++)
        {
            for(int j = 0; j < 10; j++)
            {
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```