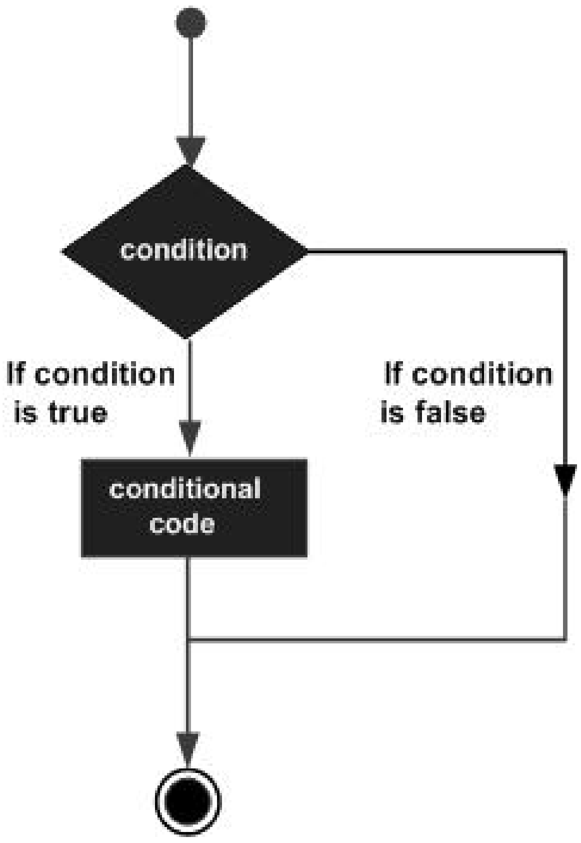
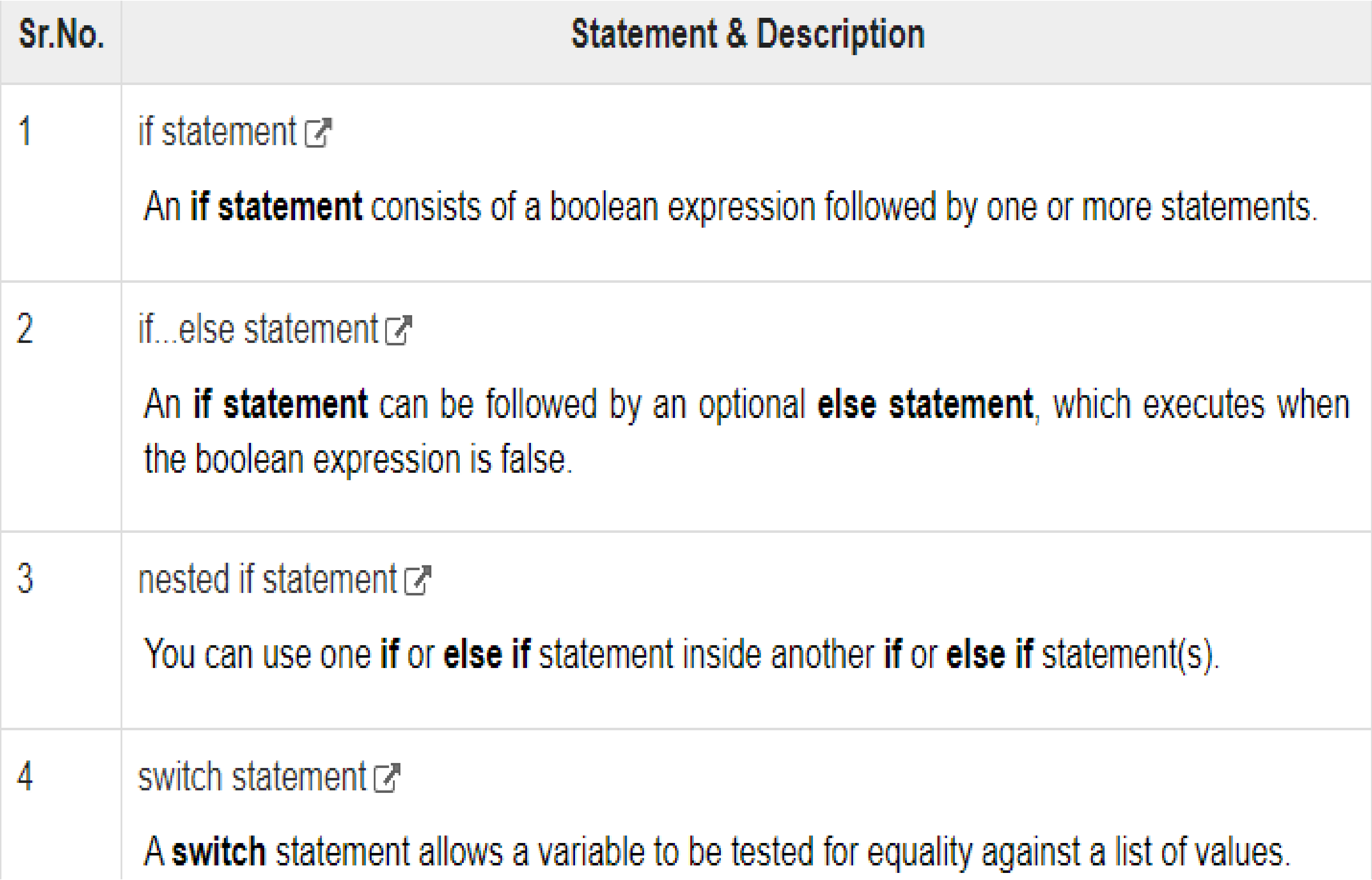
Decision Making, Control Instructions and Array

Reference: Liang, Y., 2015. Intro To Java Programming. 10th ed. Pearson Education Limited.

Java - Decision Making

Decision making structures have one or more conditions to be evaluated or tested by the program, along with a statement or statements that are to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

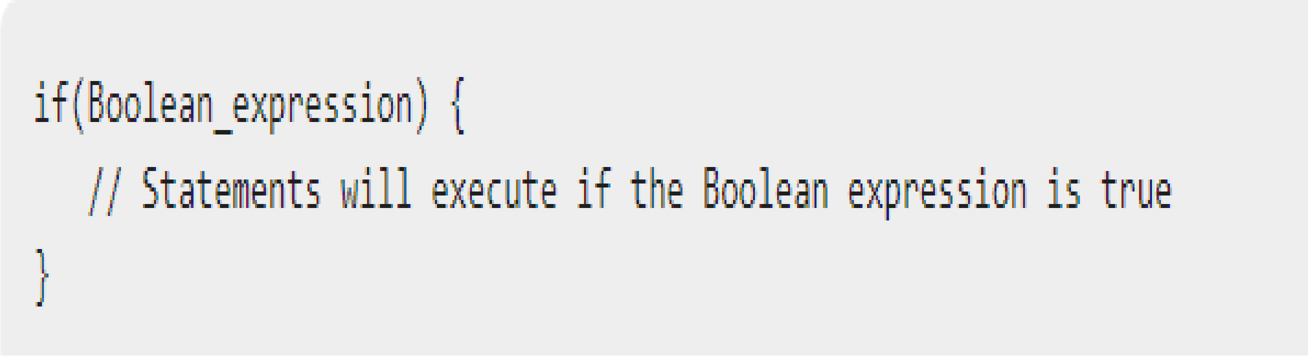




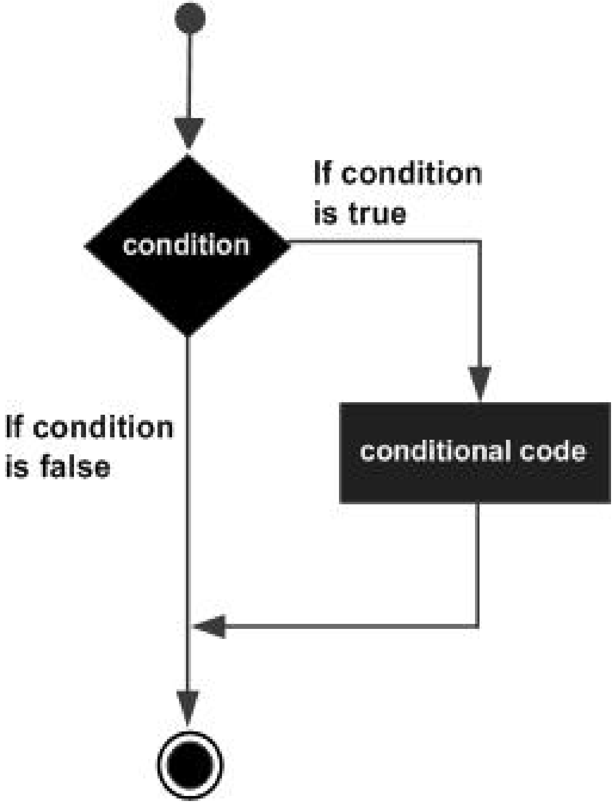
Sr.No.	Statement & Description
1	if statement An **if statement** consists of a boolean expression followed by one or more statements.
2	if...else statement An **if statement** can be followed by an optional **else statement**, which executes when the boolean expression is false.
3	nested if statement You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).
4	switch statement A **switch** statement allows a variable to be tested for equality against a list of values.

if statement

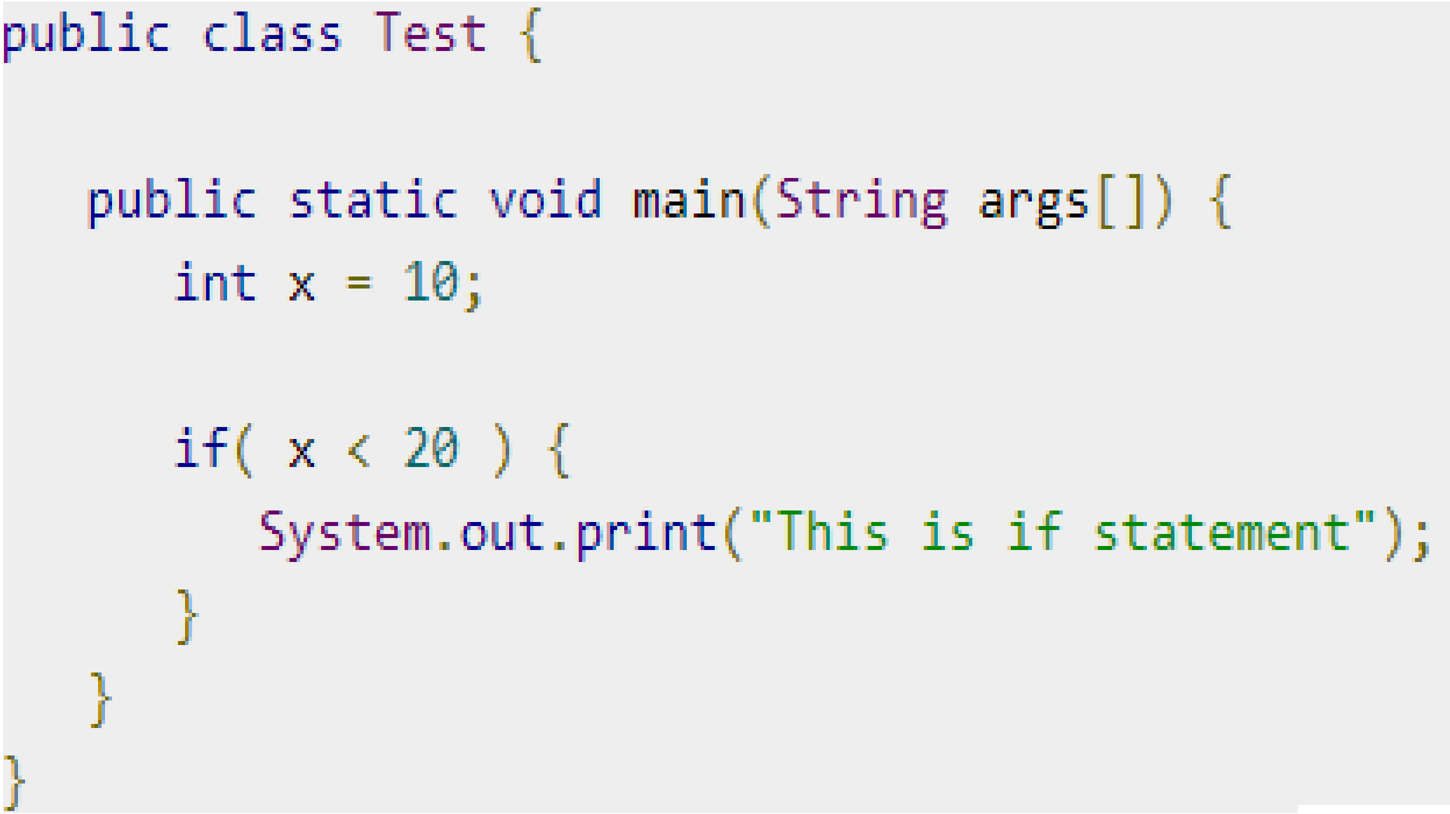
- An **if** statement consists of a Boolean expression followed by one or more statements.
- Syntax
- Following is the syntax of an if statement −



```
if(Boolean_expression) {
   // Statements will execute if the Boolean expression is true
}
```



Example

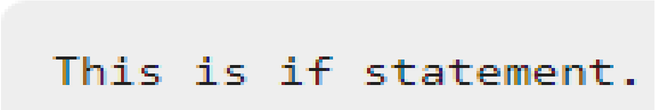


```
public class Test {

   public static void main(String args[]) {
      int x = 10;

      if( x < 20 ) {
         System.out.print("This is if statement");
      }
   }
}
```

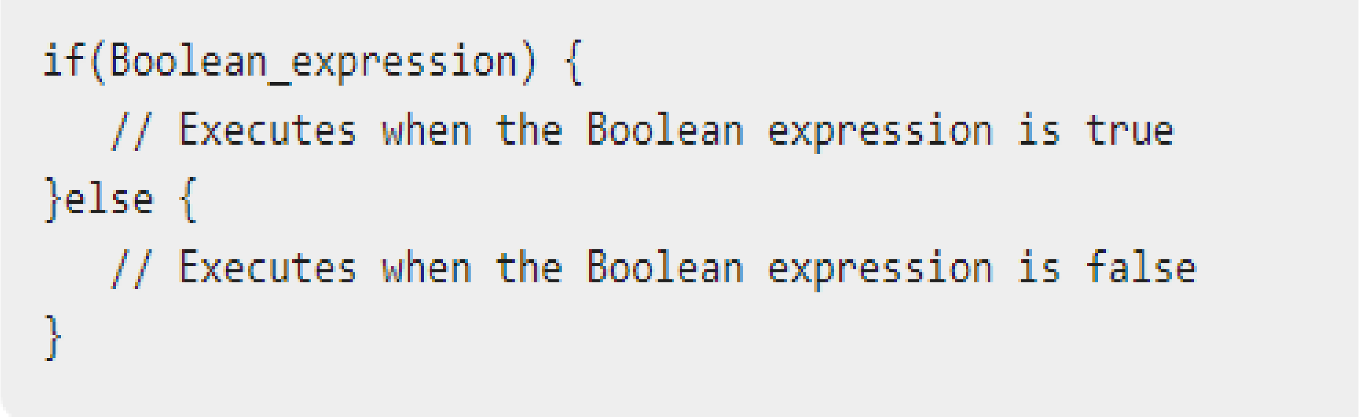
Output



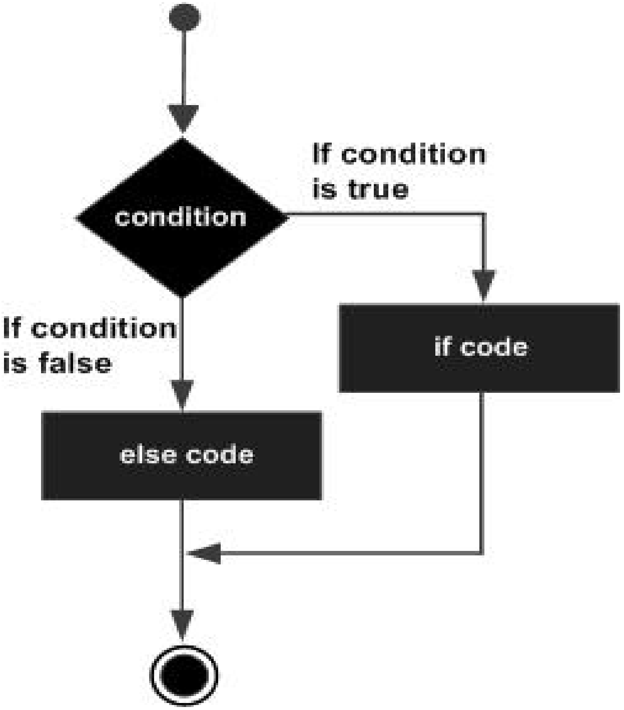
```
This is if statement.
```

if-else statement in java

- An **if** statement can be followed by an optional **else** statement, which executes when the Boolean expression is false.
- Syntax
- Following is the syntax of an if...else statement −

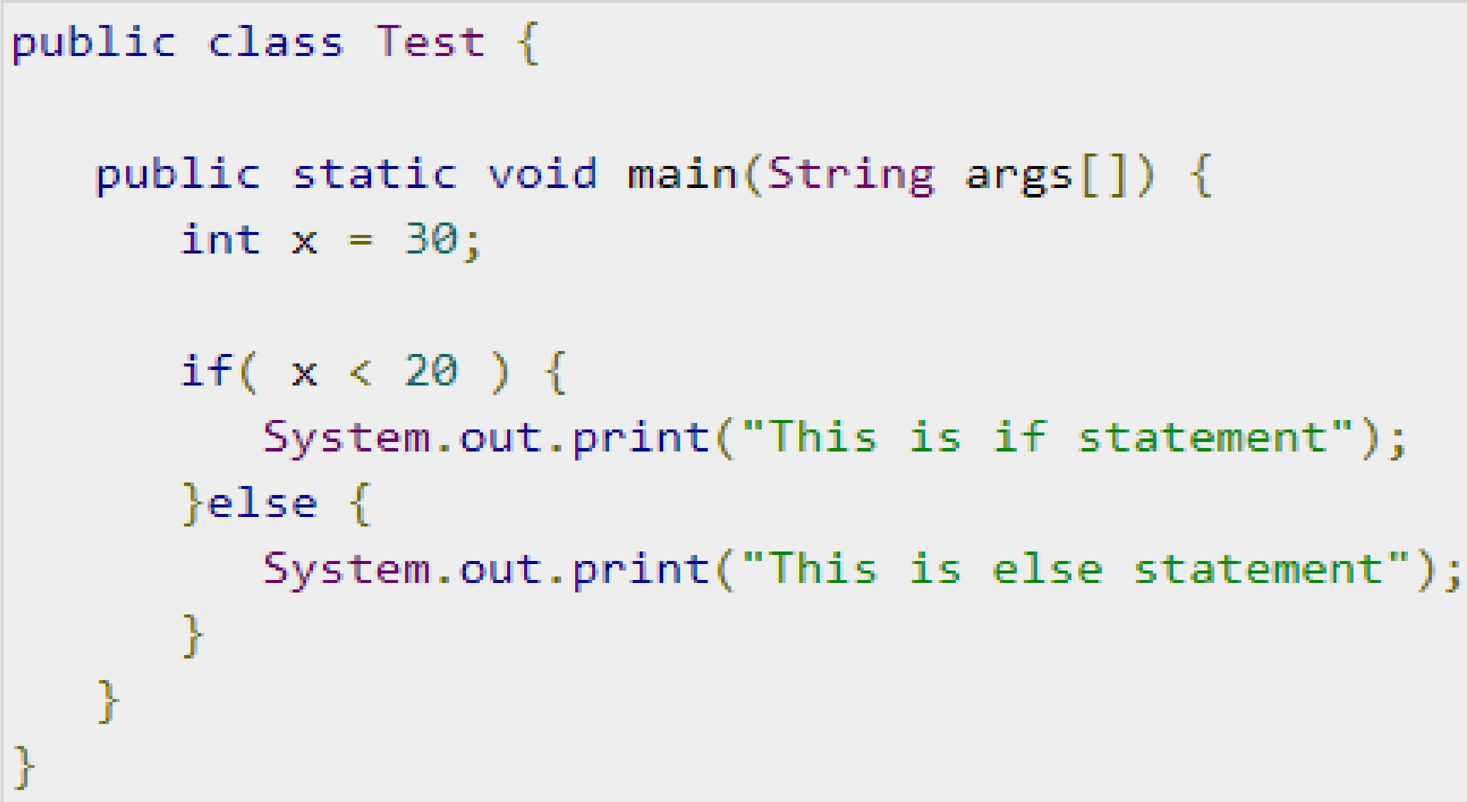


```
if(Boolean_expression) {
   // Executes when the Boolean expression is true
}else {
   // Executes when the Boolean expression is false
}
```



IQRA UNIVERSITY IU

Example

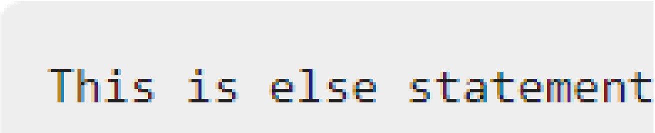


```java
public class Test {

   public static void main(String args[]) {
      int x = 30;

      if( x < 20 ) {
         System.out.print("This is if statement");
      }else {
         System.out.print("This is else statement");
      }
   }
}
```

Output



```
This is else statement
```

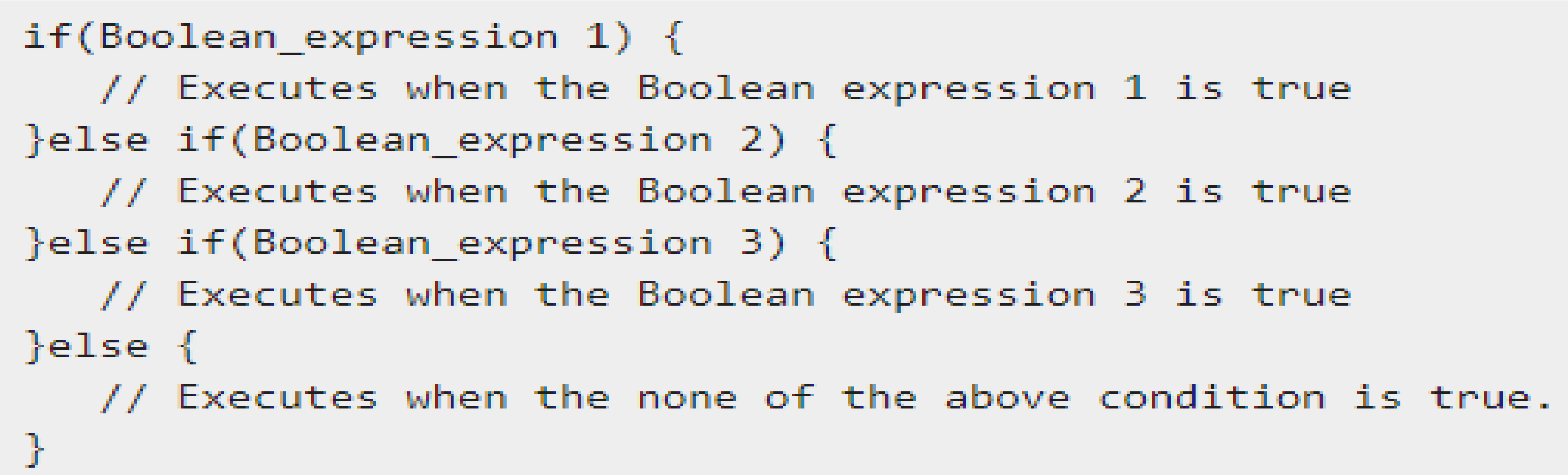
Write a program that takes a number as input and print whether it is odd or even.

```
import java.util.Scanner;
public class IFELSE {
public static void main(String[] args) {
// TODO Auto-generated method stub
     int num;
     Scanner sc= new Scanner(System.in);
    System.out.println("Enter an integer: ");
    num = sc.nextInt();
if(num % 2 == 0)
System.out.println(num +" is Even");
    else
    System.out.println(num + " is Odd");
}
}
```

```
Enter an integer: 22
22 is Even
```

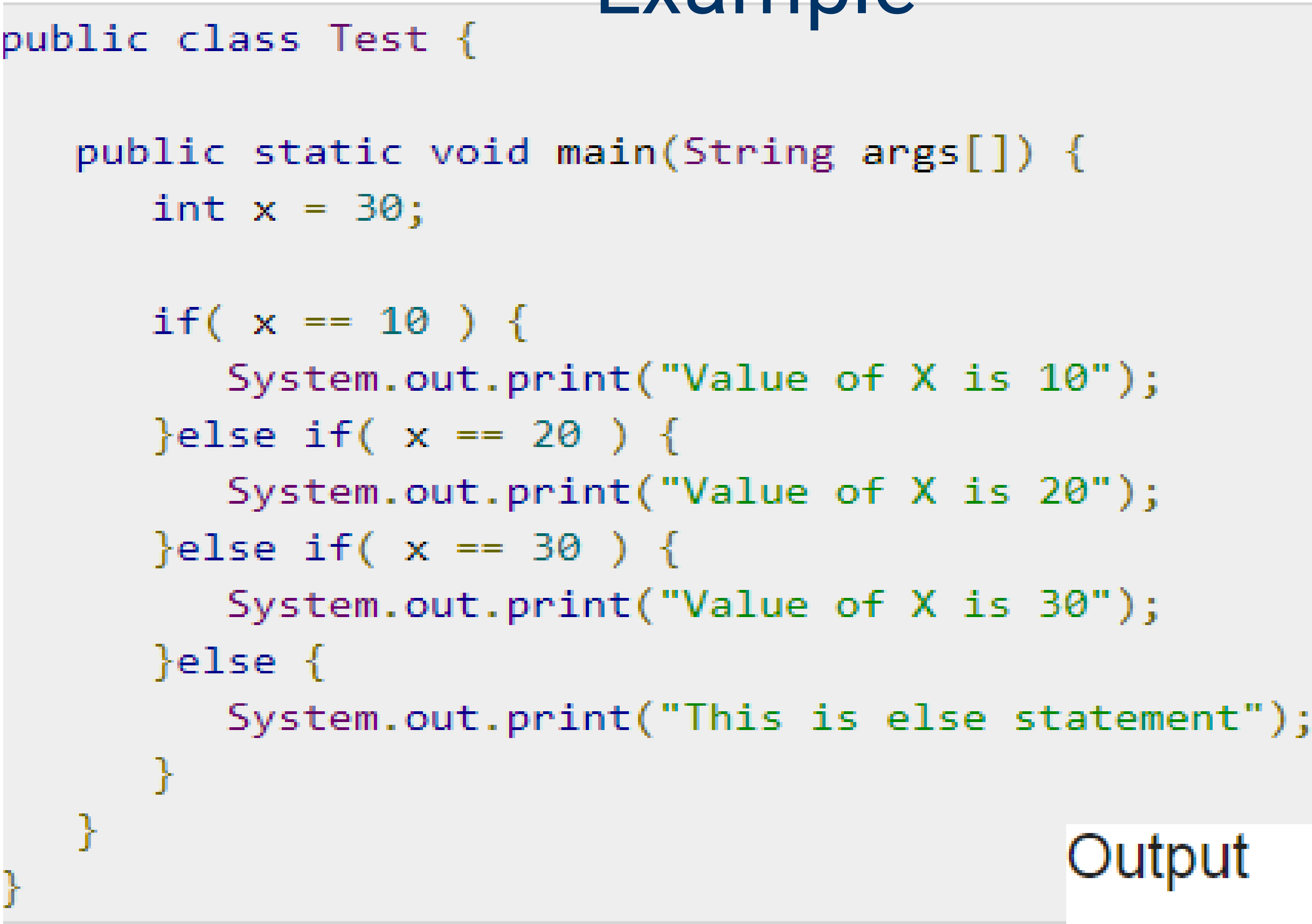
The if...else if...else Statement

- if statement can be followed by an optional *else if...else* statement, which is very useful to test various conditions using single if...else if statement.
- When using if, else if, else statements there are a few points to keep in mind.
- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of the remaining else if's or else's will be tested.



```
if(Boolean_expression 1) {
   // Executes when the Boolean expression 1 is true
}else if(Boolean_expression 2) {
   // Executes when the Boolean expression 2 is true
}else if(Boolean_expression 3) {
   // Executes when the Boolean expression 3 is true
}else {
   // Executes when the none of the above condition is true.
}
```

Example

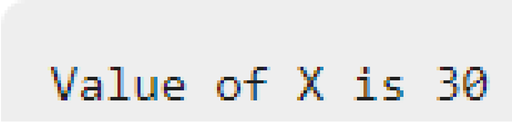


```
public class Test {

   public static void main(String args[]) {
      int x = 30;

      if( x == 10 ) {
         System.out.print("Value of X is 10");
      }else if( x == 20 ) {
         System.out.print("Value of X is 20");
      }else if( x == 30 ) {
         System.out.print("Value of X is 30");
      }else {
         System.out.print("This is else statement");
      }
   }
}
```

Output



```
Value of X is 30
```

Write a program that take month number as input (from 1 to 12) and print number of days in that month. If wrong number is given then show error message.

Using if … else if … else

```
import java.util.Scanner;
public class IFELSE {
public static void main(String[] args) {
// TODO Auto-generated method stub
 Scanner sc= new Scanner(System.in);
int month;
  System.out.print("Enter month number (1-12):");
  month=sc.nextInt();
  if(month == 1)
  {
  System.out.println("Entered month : January " );
  System.out.println("No. of days : 31 days" );
  }
  else if(month == 2)
  {
  System.out.println("Entered month : February ");
  System.out.println("No. of days :  28 or 29 days");
  }
  else if(month == 3)
  {
  System.out.println("Entered month : March " );
  System.out.println("No. of days :  31 days");
  }
else if(month == 4)
  {
  System.out.println("Entered month : April " );
  System.out.println( "No. of days :  30 days");
  }
```

```
 else if(month == 5)
   {
   System.out.println("Entered month : May ");
   System.out.println("No. of days :  31 days");
   }
else if(month == 6)
   {
   System.out.println("Entered month : June " );
   System.out.println("No. of days :  30 days");
   }
   else if(month == 7)
   {
   System.out.println( "Entered month : July " );
   System.out.println( "No. of days :  31 days" );
   }
   else if(month == 8)
   {
   System.out.println("Entered month : August " );
   System.out.println("No. of days :  31 days" );
   }
   else if(month == 9)
   {
   System.out.println( "Entered month : September
" );
   System.out.println( "No. of days :  30 days" );
   }
else if(month == 10)
   {
   System.out.println("Entered month : October " );
   System.out.println("No. of days :  31 days" );
   }
```

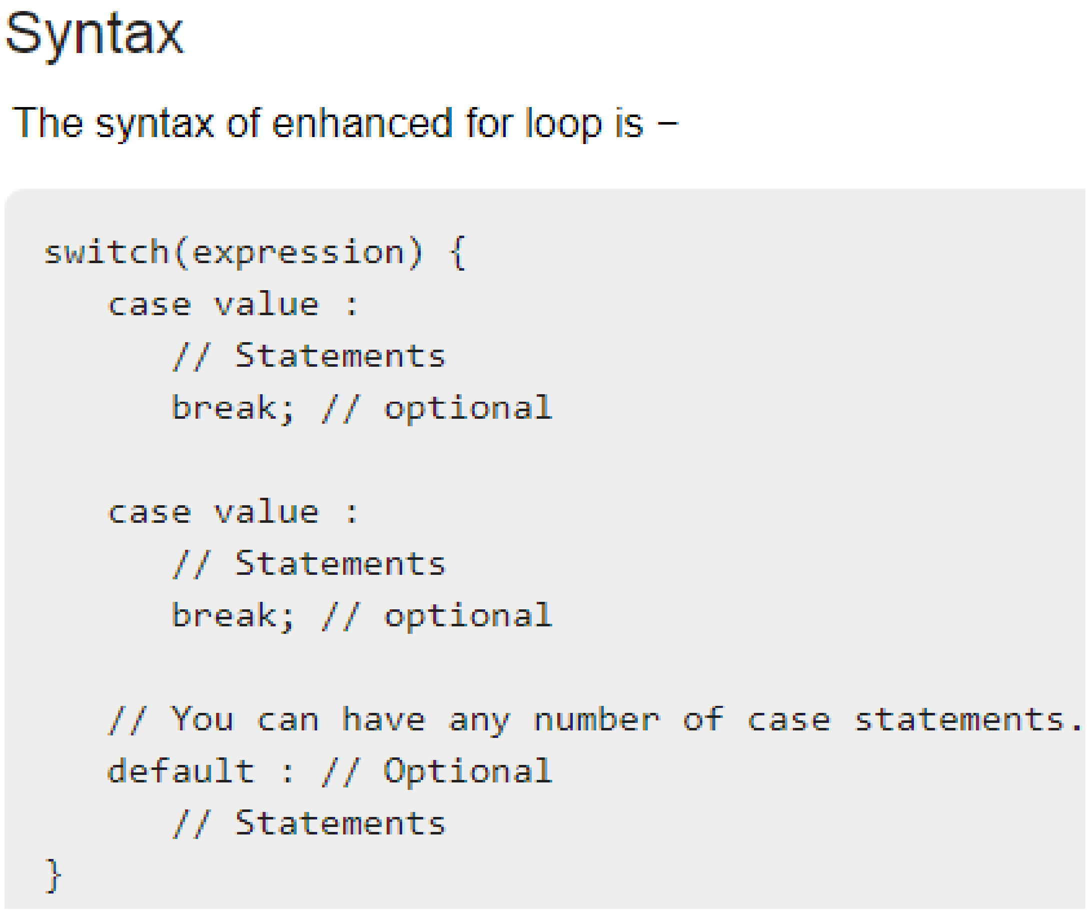
```
 else if(month == 11)
   {
   System.out.println("Entered month :
November " );
   System.out.println( "No. of days :  30 days
" );
   }
else if(month == 12)
   {
   System.out.println( "Entered month :
December " );
   System.out.println( "No. of days :  31 days
" );
   }
   else
   {
   System.out.println("Invalid input! Please
enter month number between (1-12).");
   }

}
}
```

```
Enter month number (1-12):12
Entered month : December
No. of days :  31 days
```

Switch Statement In Java

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.



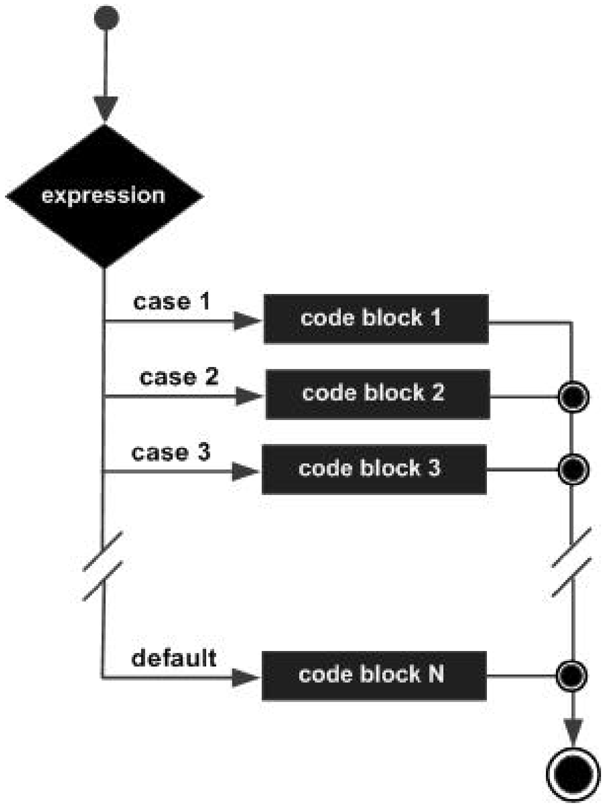
Syntax

The syntax of enhanced for loop is –

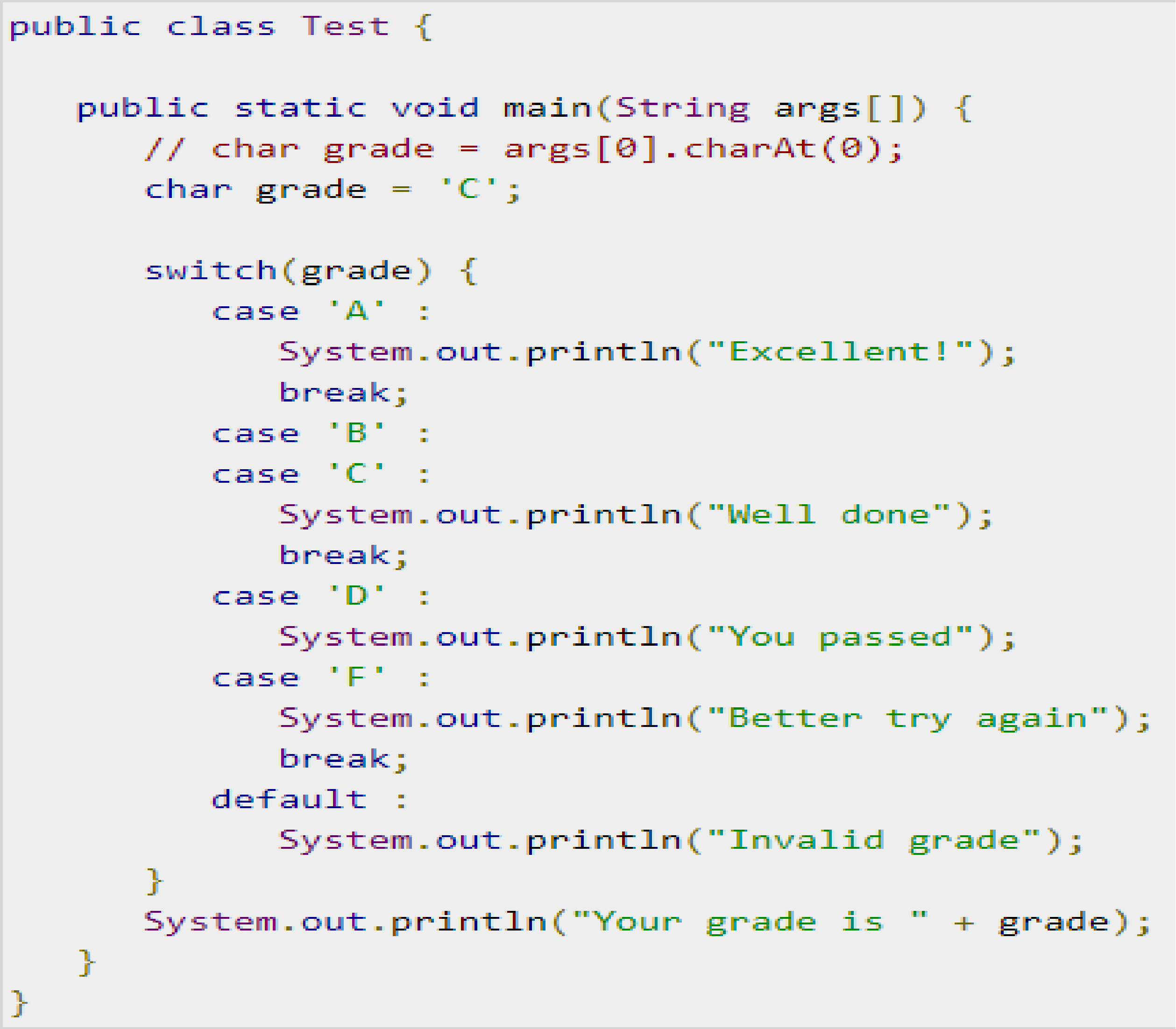
```
switch(expression) {
   case value :
      // Statements
      break; // optional

   case value :
      // Statements
      break; // optional

   // You can have any number of case statements.
   default : // Optional
      // Statements
}
```



Example

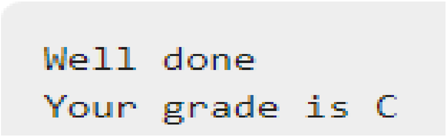


```
public class Test {

   public static void main(String args[]) {
      // char grade = args[0].charAt(0);
      char grade = 'C';

      switch(grade) {
         case 'A' :
            System.out.println("Excellent!");
            break;
         case 'B' :
         case 'C' :
            System.out.println("Well done");
            break;
         case 'D' :
            System.out.println("You passed");
         case 'F' :
            System.out.println("Better try again");
            break;
         default :
            System.out.println("Invalid grade");
      }
      System.out.println("Your grade is " + grade);
   }
}
```

Output



```
Well done
Your grade is C
```

Design a Java program to use an Switch to determine which season a particular month is in.

```
import java.util.Scanner;
public class IFELSE {
public static void main(String[] args) {
// TODO Auto-generated method stub
  Scanner sc= new Scanner(System.in);
int month;
   System.out.print("Enter month number (1-12):");
   month=sc.nextInt();
   String season;
   if(month == 11 || month == 12 || month == 1)
   season = "Winter";
   else if(month == 2 || month == 3 || month == 4)
   season = "Spring";
   else if(month == 5 || month == 6 || month == 7)
   season = "Summer";
   else if(month == 8 || month == 9 || month == 10)
   season = "Autumn";
   else
   season = "Invalid Month";
   System.out.println(season);
}
}
```

```
import java.util.Scanner;
public class IFELSE {
public static void main(String[] args) {
// TODO Auto-generated method stub
  Scanner sc= new Scanner(System.in);
int month;
System.out.print("Enter month number (1-12) :");
   month=sc.nextInt();
   String season;
switch(month)
case 11:
case 12:
case 1:
season = "Winter";
break;
case 2:
case 3:
case 4:
season = "Spring";
break;
case 5:
case 6:
case 7:
season = "Summer";
break;
case 8:
case 9:
case 10:
season = "Autumn";
break;
default:
season = "Invalid Month";
break;
System.out.println(season);
}
}
```

Write a program that take month number as input (from 1 to 12) and print number of days in that month. If wrong number is given then show error message.

Using switch case

```
import java.util.Scanner;
public class switch {
public static void main(String[] args) {
// TODO Auto-generated method stub
  Scanner sc= new Scanner(System.in);
int month;
   System.out.print("Enter month number (1-12):");
   month=sc.nextInt();
   switch(month)
   {
 case 1:
   System.out.println("Entered month : January " );
   System.out.println("No. of days : 31 days" );
   break;

 case 2:
    System.out.println("Entered month : February ");
   System.out.println("No. of days :  28 or 29 days");
   break;

 case 3:
     System.out.println("Entered month : March " );
   System.out.println("No. of days :  31 days");
   break;

 case 4:
   System.out.println("Entered month : April " );
   System.out.println( "No. of days :  30 days");
  break;

case 5:
   System.out.println("Entered month : May ");
   System.out.println("No. of days :  31 days");
   break;

case 6:
     System.out.println("Entered month : June " );
   System.out.println("No. of days :  30 days");
   break;

case 7:
   System.out.println( "Entered month : July " );
   System.out.println( "No. of days :  31 days" );
   break;

case 8:
   System.out.println("Entered month : August " );
   System.out.println("No. of days :  31 days" );
   break;

case 9:
   System.out.println( "Entered month : September
" );
   System.out.println( "No. of days :  30 days" );
   break;

case 10:
   System.out.println("Entered month : October " );
   System.out.println("No. of days :  31 days" );
   break;

 case 11:
   System.out.println("Entered month :
November " );
   System.out.println( "No. of days :  30 days
" );
   break;

 case 12:
   System.out.println( "Entered month :
December " );
   System.out.println( "No. of days :  31 days
" );
   break;

 default:
   System.out.println("Invalid input! Please
enter month number between (1-12).");
    }
    }
}
```

```
Enter month number (1-12):12
Entered month : December
No. of days :  31 days
```

The marks obtained by a student in 5 different subjects are input through the keyboard. The student gets a division as per the following rules:

- Percentage above or equal to 60 - First division
- Percentage between 50 and 59 - Second division
- Percentage between 40 and 49 - Third division
- Percentage less than 40 - Fail

Write a program to calculate the division obtained by the student.

```
import java.util.Scanner;
class HelloWorld {
public static void main(String[] args)
{double m1, m2, m3, m4, m5, per ;
Scanner sc=new Scanner(System.in);
System.out.println("Enter marks in five subjects ") ;
m1=sc.nextDouble();
m2=sc.nextDouble();
m3=sc.nextDouble();
m4=sc.nextDouble();
m5=sc.nextDouble();
per = ( m1 + m2 + m3 + m4 + m5 ) / 5 ;
if ( per >= 60 )
System.out.println("First division " + per) ;
else if ( ( per >= 50 ) && ( per < 60 ) )
System.out.println(" Second division " + per) ;
else if ( ( per >= 40 ) && ( per < 50 ) )
System.out.println(" Third division " +per )  ;
else if ( per < 40 )
System.out.println("Fail "  + per)   ;
else
System.out.println("Invalid input")   ;
}
}
```

Java Program to Make a Simple Calculator Using switch...case

```
import java.util.Scanner;
public class prac {
public static void main(String[] args)
{
char operator;
double number1, number2, result;
Scanner sc= new Scanner(System.in);
System.out.println("Choose an operator: +, -, *, or /");
operator = sc.next().charAt(0);
System.out.println("Enter first number");
 number1 = sc.nextDouble();
 System.out.println("Enter second number");
 number2 = sc.nextDouble();
 switch (operator) {
case '+':
result = number1 + number2;
System.out.println(number1 + " + " + number2 + " = " + result);
break;
case '-':
result = number1 - number2;
System.out.println(number1 + " - " + number2 + " = " + result);
break;
case '*':
result = number1 * number2;
System.out.println(number1 + " * " + number2 + " = " + result);
 break;
case '/':
if (number2==0)
System.out.println("Cannot do Division");
else {
result = number1 / number2;
 System.out.println(number1 + " / " + number2 + " = " + resul
break;
  default:
   System.out.println("Invalid operator!");
    break;
    }
    sc.close();
    }
}
```

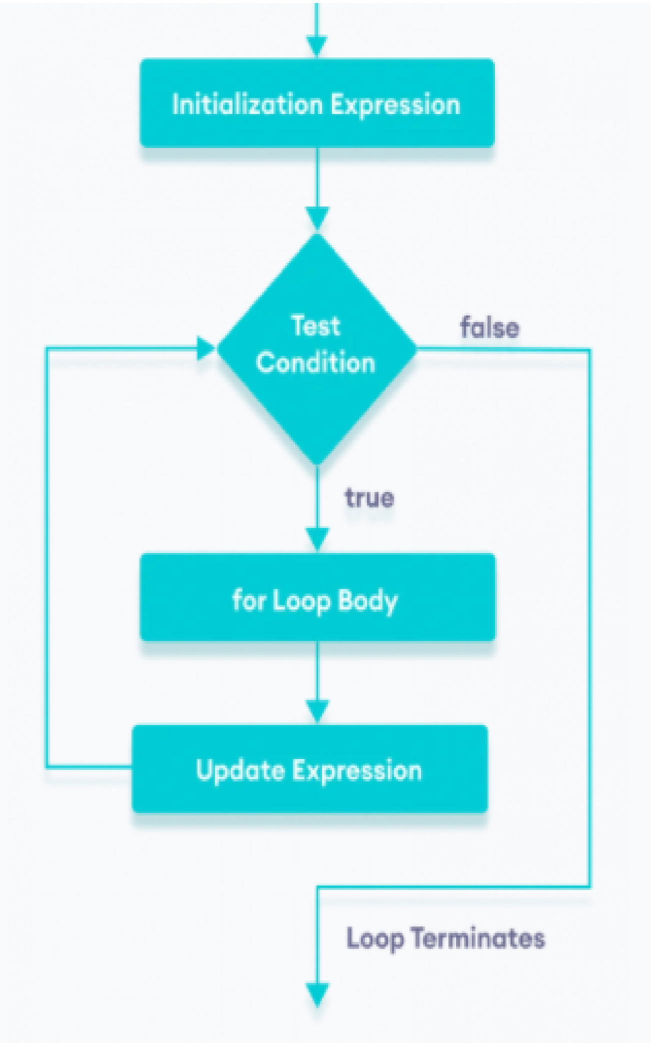
For Loop

- Java for loop is used to run a block of code for a certain number of times. The syntax of for loop is:

```
for (initialExpression; testExpression; updateExpression) {
   // body of the loop
}
```

Here,

- The initialExpression initializes and/or declares variables and executes only once.
- The condition is evaluated. If the condition is true, the body of the for loop is executed.
- The updateExpression updates the value of initialExpression.
- The condition is evaluated again. The process continues until the condition is false.



<u>Note</u>

Infinite loop

for(;;)

```
public class prac {
public static void main(String[] args) {
    int n = 5;
for (int i = 1; i <= n; ++i)
      System.out.println(i+" Java is fun");
}
}
```

Output:

```
1 Java is fun
2 Java is fun
3 Java is fun
4 Java is fun
5 Java is fun
```

Method 1:

```
public class prac {
public static void main(String[] args) {
    int[] num = {1, 2, 3, 4};
     for (int i=0;i<num.length;i++) {
       System.out.println(num [i]); }
  }
}
```

Method 2:

```
class prac {
  public static void main(String[] args) {
  int[] num = {1, 2, 3, 4};
for (int i: num) {
       System.out.println(i);    }
  }
}
```

Output:

```
1
2
3
4
```

Method 1:

```
public class prac {
public static void main(String[] args) {
char[] vowels = {'a', 'e', 'i', 'o', 'u'};
for (int i = 0; i < vowels.length; ++ i)
System.out.println(vowels[i]);
}
}
```

Method 2:

```
class prac {
  public static void main(String[] args) {
char[] vowels = {'a', 'e', 'i', 'o', 'u'};
for (char item:vowels)
System.out.println(item);
}
  }
}
```

Output:

```
a
e
i
o
u
```

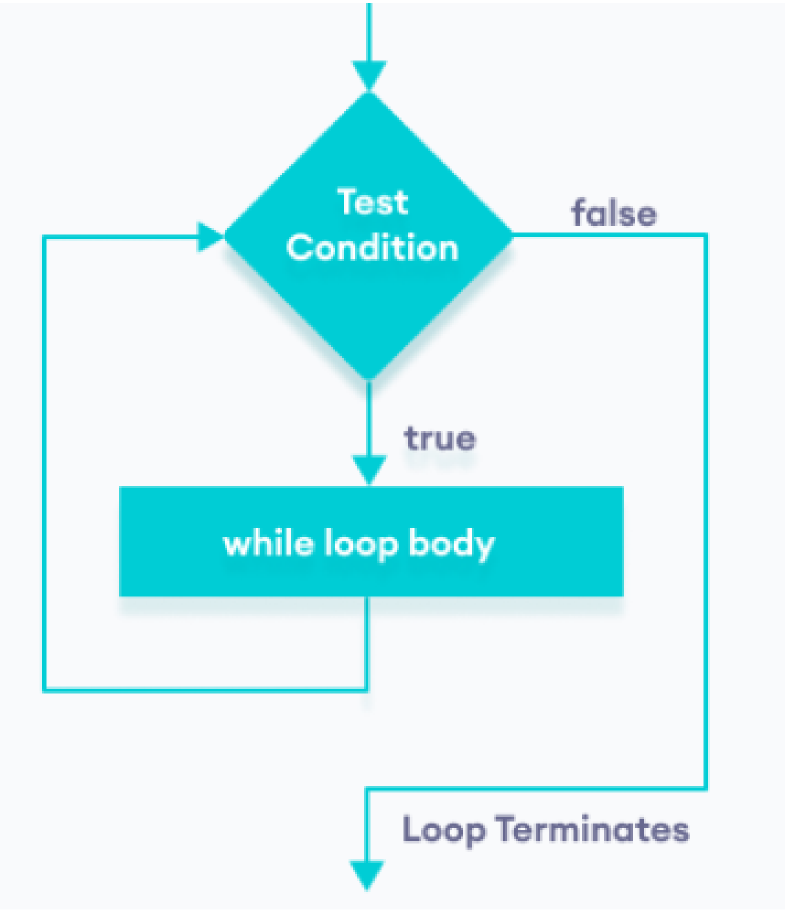
While loop

Java while loop is used to run a specific code until a certain condition is met. The syntax of the while loop is:

```
while (testExpression) {
   // body of loop
}
```

Here,

- A while loop evaluates the textExpression inside the parenthesis ().
- If the textExpression evaluates to true, the code inside the while loop is executed.
- The textExpression is evaluated again.
- This process continues until the textExpression is false.
- When the textExpression evaluates to false, the loop stops.



Method 1:

```
public class prac {
public static void main(String[] args) {
int num =1;
while(num<=20){
System.out.println(num);
num++;}
  }
}
```

Method 2:

```
public class prac {
public static void main(String[] args) {
int num =1;
while(num<=20){
System.out.println(num++);
}
}
}
```

OUTPUT:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

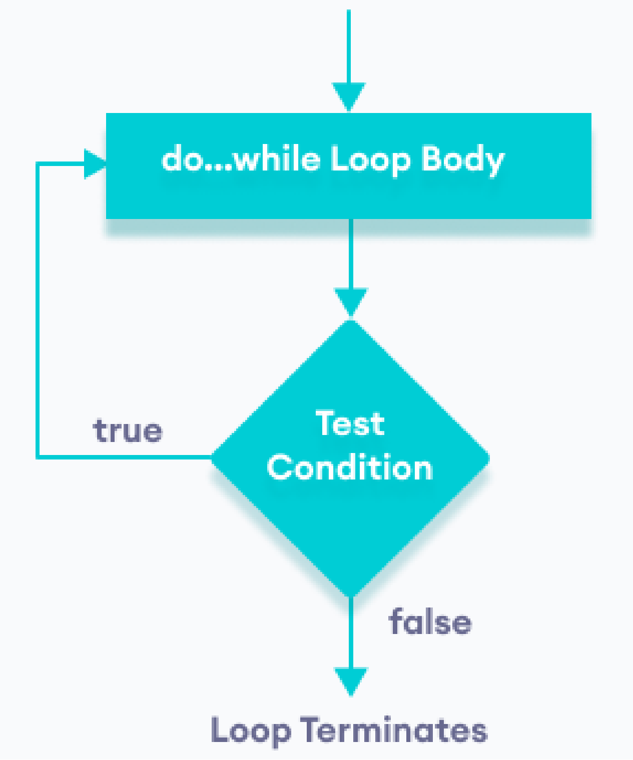
Do While Loop

The do...while loop is similar to while loop. However, the body of do...while loop is executed once before the test expression is checked. For example,

```
do {
   // body of loop
} while(textExpression);
```

Here,

- The body of the loop is executed at first. Then the textExpression is evaluated.
- If the textExpression evaluates to true, the body of the loop inside the do statement is executed again.
- The textExpression is evaluated once again.
- If the textExpression evaluates to true, the body of the loop inside the do statement is executed again.
- This process continues until the textExpression evaluates to false. Then the loop stops.



Method 1:

```
public class prac {
public static void main(String[] args) {
int num =1;
do{
System.out.println(num);
num++;}
while(num<=20);
 }
}
```

Method 2:

```
public class prac {
public static void main(String[] args) {
int num =1;
do{
System.out.println(num++);
} while(num<=20);
}
}
```

OUTPUT:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Array

An array is a collection of similar types of data.

For example, if we want to store the names of 100 people then we can create an array of the string type that can store 100 names.

```
String[] array = new String[100];
```

Here, the above array cannot store more than 100 names.

How to declare an array in Java?

```
dataType[] arrayName;
```

dataType - it can be primitive data types like int, char, double, byte, etc. or Java objects

arrayName - it is an identifier

For example,

```
double[] data;
data = new double[10];                    or          double[] data = new double[10];
```

How to Initialize Arrays in Java?

In Java, we can initialize arrays during declaration. For example,

```
int[] age = {12, 4, 5, 2, 5};
```

```
Import java.util.Scanner;
public class ArrayInsert {
public static void main(String[] args) {
int[] arr = new int[10];
Scanner sc = new Scanner(System.in);
for(int i=0;i<10;i++){
System.out.println("Enter array element at index : "+ i);
arr[i] = sc.nextInt();
}
System.out.println("Elements contained in array are: ");
for(int j=0; j<arr.length; j++){
System.out.println(arr[j]);
}
}
}
```

Output:

```
Enter array element at index : 0=1
Enter array element at index : 1=2
Enter array element at index : 2=3
Enter array element at index : 3=4
Enter array element at index : 4=5
Enter array element at index : 5=6
Enter array element at index : 6=7
Enter array element at index : 7=8
Enter array element at index : 8=9
Enter array element at index : 9=10
Elements contained in array are:
1
2
3
4
5
6
7
8
9
10
```

```
import java.util.Scanner;
public class prac {
public static void main(String[] args) {
int[] arr = new int[5];
Scanner sc = new Scanner(System.in);
for(int i=0;i<5;i++)
{
System.out.print("Enter array element at index : "+ i+"=");
arr[i] = sc.nextInt();
}
for(int i=0;i<5;i++)
{
System.out.println("Enter array element at index : "+ i +"-->"+ arr[i]);
}
System.out.println("Elements contained in array in reverse order: ");
for(int j=arr.length-1; j>=0; j–)
{
System.out.println(arr[j]);
}
}
}
```

Output:

```
Enter array element at index : 0=1
Enter array element at index : 1=2
Enter array element at index : 2=3
Enter array element at index : 3=4
Enter array element at index : 4=5
Enter array element at index : 0-->1
Enter array element at index : 1-->2
Enter array element at index : 2-->3
Enter array element at index : 3-->4
Enter array element at index : 4-->5
Elements contained in array in reverse order:
5
4
3
2
1
```

THANK YOU