Control statements

In this Lecture we Learn

- Relation operators in Java
- Logical operators in Java
- Bitwise Operators in Java
- Assignment Operators in Java
- Miscellaneous Operators in Java
- Java Selection Statements
  - If Statement
  - If else Statement
  - If else if Statement
  - Switch Statement

Relation Operators

The following table shows all relation operators supported by Java.

Operator	Description
==	Check if two operand are equal
!=	Check if two operand are not equal.
>	Check if operand on the left is greater than operand on the right
<	Check operand on the left is smaller than right operand
>=	check left operand is greater than or equal to right operand
<=	Check if operand on left is smaller than or equal to right operand

Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
== (equal to)	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!= (not equal to)	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
> (greater than)	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.

Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
< (less than)	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>= (greater than or equal to)	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<= (less than or equal to)	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Relational Operators

Which can be used to check the Condition, it always return true or false. Lets suppose variable **A** hold 8 and **B** hold 3.

Operators	Example (int A=8, B=3)	Result
<	A<B	False
<=	A<=10	True
>	A>B	True
>=	A<=B	False
==	A== B	False
!=	A!=(-4)	True

Logical Operators

Assume Boolean variables A holds true and variable B holds false, then

Operator	Description	Example
Operator	Description	Example
&& (logical and)	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false
|| (logical or)	Called Logical OR Operator. If any of the two operands are non-zero, then the condition becomes true.	(A || B) is true
! (logical not)	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true

Logical Operators (Example)

Java supports following 3 logical operator. Suppose a=1 and b=0;

Operator	Description	Example
&&	Logical AND	(a && b) is false
||	Logical OR	(a || b) is true
!	Logical NOT	(!a) is false

Bitwise Operators

Java defines several bitwise operators that can be applied to the integer types long, int, short, char and byte

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
<<	left shift
>>	right shift

Now lets see truth table for bitwise &, | and ^

a	b	a & b	a | b	a ^ b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

The bitwise shift operators shifts the bit value. The left operand specifies the value to be shifted and the right operand specifies the number of positions that the bits in the value are to be shifted. Both operands have the same precedence. **Example**

a = 0001000

b= 2

a << b= 0100000

a >> b= 0000010

Assignment Operators

Operator	Description	Example
=	assigns values from right side operands to left side operand	a=b
+=	adds right operand to the left operand and assign the result to left	a+=b is same as a=a+b
-=	subtracts right operand from the left operand and assign the result to left operand	a-=b is same as a=a-b
*=	mutiply left operand with the right operand and assign the result to left operand	a*=b is same as a=a*b
/=	divides left operand with the right operand and assign the result to left operand	a/=b is same as a=a/b
%=	calculate modulus using two operands and assign the result to left operand	a%=b is same as a=a%b

Assignment operators (Example)

Which can be used to assign a value to a variable.
Lets suppose variable **A** hold 8 and **B** hold 3.

Operator	Example (int A=8, B=3)	Result
+=	A+=B or A=A+B	11
-=	A-=3 or A=A+3	5
*=	A*=7 or A=A*7	56
/=	A/=B or A=A/B	2
%=	A%=5 or A=A%5	3
=a=b	Value of b will be assigned to a	

Assignment operators Program Example

```
public class JavaApplication1 {

    public static void main(String[] args) {
            //+= Assignment Operator
                    int value1 = 4;
                    int value2 = 8;
                    value2 += value1;
                    System. out .println( "value1 is " +value1);
                    System. out .println( "value2 is " +value2);
    }

}
```

Assignment operators Program Example

```
public class JavaApplication1 {

    public static void main(String[] args) {
            //+= Assignment Operator
                    int value1 = 4;
                    int value2 = 8;
                    value2 += value1;
                    System. out .println( "value1 is " +value1);
                    System. out .println( "value2 is " +value2);
    }
}
```

value2 += value1
value2 = value2+value1
Value2=8+4=12

Output is :
value1 is 4
value2 is 12

Miscellaneous Operators

- There are few other operators supported by Java Language
- **Conditional Operator (? :)**
- Conditional operator is also known as the **ternary operator**. This operator consists of three operands and is used to evaluate Boolean expressions. The goal of the operator is to decide, which value should be assigned to the variable. The operator is written as :

```
variable x = (expression) ? value if true : value if false
```

Program Example of Conditional Operator (? :)

```
14    public class JavaApplication1 {
15
16        public static void main(String[] args) {
17            int a, b;
18            a = 10;
19            b = (a == 1) ? 20: 30;
20            System.out.println( "Value of b is : " +  b );
21
22            b = (a == 10) ? 20: 30;
23            System.out.println( "Value of b is : " + b );
24
25        }
26
27    }
```

javaapplication1.JavaApplication1 > main >

Output ×

JavaApplication1 (run) × | JavaApplication1 (run) #2 ×

```
run:
Value of b is : 30
Value of b is : 20
```

```
public class JavaApplication1 {

    public static void main(String[] args) {
      int a, b;
      a = 10;
      b = (a == 1) ? 20: 30;
      System.out.println( "Value of b is : " +  b );

      b = (a == 10) ? 20: 30;
      System.out.println( "Value of b is : " + b );

    }

}
```

Condition .Checking if Condition is true Print 20 if false print 30

```
public class JavaApplication1 {

   public static void main(String[] args) {
     int a, b;
     a = 10;
     b = (a == 1) ? 20: 30;
     System.out.println( "Value of b is : " +  b );

     b = (a == 10) ? 20: 30;
     System.out.println( "Value of b is : " + b );

   }
}
```

Condition is False

Condition is true

Output is :

Value of b is : 30

Value of b is : 20

Miscellaneous Operators

- **instanceof Operator**
- This operator is used only for object reference variables. The operator checks whether the object is of a particular type (class type or interface type). instanceof operator is written as:

(Object reference variable) instanceof (class/interface type)

- If the object referred by the variable on the left side of the operator passes the IS-A check for the class/interface type on the right side, then the result will be true

Program Example of instanceof Operator

```
14    public class JavaApplication1 {
15
16        public static void main(String[] args) {
17            String name = "Adil";
18            // following will return true since name is type of String
19            boolean result = name instanceof String;
20            System.out.println( result );
21        }
22
23    }
```

utput ×

JavaApplication1 (run) × | JavaApplication1 (run) #2 ×

```
run:
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program Example of instanceof Operator

```
public class JavaApplication1 {


   public static void main(String[] args) {
     String name = "Adil";
 // following will return true since name is type of String
     boolean result = name instanceof String;
     System.out.println( result );
   }


}
```

Output is :

true

(% = Remainder) Operator

- **Examples**
- 15 % 2 = 1
- 20 % 5 = 0
- 38 % 6 = 2
- 45 % 7 = 3
- 52 % 8 = 4
- 20 % 3 = 2

Precedence of Operators

- All operations inside of () are evaluated first
- *, /, and % are at the same level of precedence and are evaluated next
- + and – have the same level of precedence and are evaluated last
- When operators are on the same level
  - Performed from left to right (associativity)
- `3 * 7 - 6 + 2 * 5 / 4 + 6` means

```
(((3 * 7) - 6) + ((2 * 5) / 4 )) + 6
```

Precedence of Operators

- Java has a large operator set.
- When an expression involves two or more operators, two rules are applied to determine the meaning of the expression:-
  - **Priority** - Higher priority operators take priority over lower priority ones.
  - 2 + 3 * 5 is 17 not 25.

Precedence of Operators

Precedence	Operators
Highest	( )
Next	* , / , %
Lowest	+ , -

Same Precedence Example

- For addition, subtraction, division and multiplication:
  - **Left-to-right** rule applies
- A+B+C **means** (A+B)+C

- For exponentiation
  - **Right-to-left** rule applies
- A ^ B ^ C means A ^ (B ^ C)

Activity

- **10+2*3+5*4+12/6+5*20/2**
- Here highest Precedence first is * then / then last Precedence of +

Java If-else Statements

- The Java *if statement* is used to test the condition. It returns *true* or *false*. There are various types of if statement in java.

- if statement
- if-else statement
- nested if statement
- if-else-if ladder

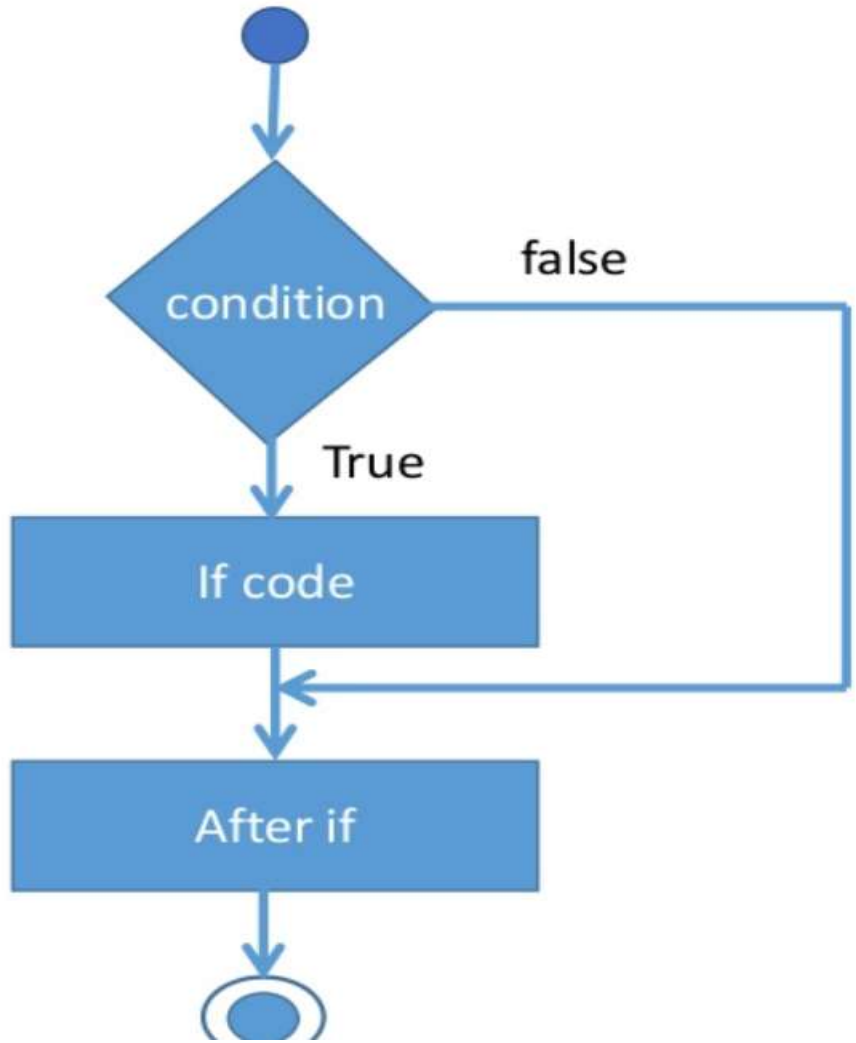
Java if Statement

- The if statement tests the condition. It executes the if statement if condition is true.
- **Syntax:**

```
if(condition){
        //code to be executed
}
```

- If the condition (or Boolean expression) evaluates to true then the block of code inside the if statement will be executed. If not the first set of code after the end of the if statement (after the closing curly brace) will be executed.

Flow Chart of if Statement



Java if Statement Example

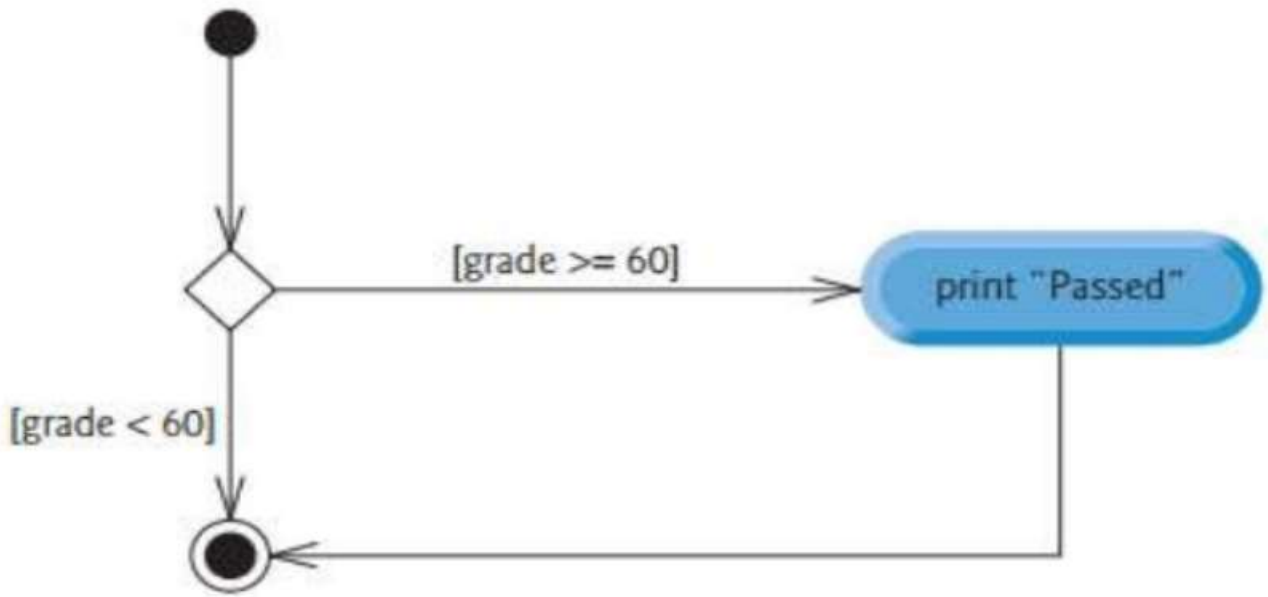
- if (bank balance is zero)
  - Borrow money

- if (room is dark)
  - Put on lights
- if (today is Sunday)
  - Lets play football
- if (student marks is 60)
  - You are pass
- if(code is 0)
  - Person is male
- if (age is more than 55)
  - Person is retired

Java if Statement Example

- The preceding pseudo code If statement can be written in java as

```
if (grade >= 60 )
System.out.println("Passed");
```



Program Example of if Statement

```
12      public class MyFirstProgram {
13
14          public static void main(String[] args) {
15              int number=20;
16              if(number>10)
17                  System.out.println("i am in if Statement");
18
19          }
20
21      }
```

myfirstprogram.MyFirstProgram > main >

Output - MyFirstProgram (run)

```
run:
i am in if Statement
BUILD SUCCESSFUL (total time: 0 seconds)
```

Another Example of is Statement

```
12  public class MyFirstProgram {
13
14      public static void main(String[] args) {
15          int a=10,b=20;
16              if (a<b) {
17
18                  System.out.println("This is if statement example");
19
20              }
21
22              }
23
24  }
```

myfirstprogram.MyFirstProgram

Output - MyFirstProgram (run)

```
run:
This is if statement example
BUILD SUCCESSFUL (total time: 0 seconds)
```

Another Example of is Statement

```
public class MyFirstProgram {

    public static void main(String[] args) {
        int a=10,b=20;
        if (a<b) {

            System.out.println("This is if statement example");

        }

        }
}
```

Declare two variables and both initialized

Checking condition if a is less than b then return true otherwise return false

One more Example of if statement

```
14      public static void main(String[] args) {
15          int a=20,b=10;
16              if (a<b) {
17
18                  System.out.println("a is less than b");
19
20              }
21              if(a>b){
22
23                  System.out.println("a is greater than b ");
24              }
25  }
26
27  }
```

myfirstprogram.MyFirstProgram

Output - MyFirstProgram (run)

```
run:
a is greater than b
BUILD SUCCESSFUL (total time: 0 seconds)
```

One more Example of if statement

```
public class MyFirstProgram {

   public static void main(String[] args) {
      int a=20,b=10;
        if (a<b) {

          System.out.println("a is less than b");
        }
        if(a>b){

           System.out.println(“a is greater than b ");
        }
}

}
```

Write a java program which gets the value of two integers from user and display whether first number is lower than, equal to or greater than the second number.

```java
public static void main(String[] args) {
    int number_1,number_2;
   Scanner user_input = new Scanner(System.in);
   System.out.println("Please Enter First Number: ");
   number_1=user_input.nextInt();
   System.out.println("Please Enter Second Number: ");
   number_2=user_input.nextInt();
    if(number_1==number_2)
    {
          System.out.println("Both Numbers are Equal");
    }
    if(number_1<number_2)
    {
          System.out.println("First Number is Less than Second");
    }
    if(number_1>number_2)
    {
          System.out.println("First Number is Greater than Second");
    }
    if(number_1!=number_2)
    {
          System.out.println("Both Number are Not Equal");
    }
}
```

Nested if Statement in java

- It is always legal to nest if-else statements which means you can use one if or else if statement inside another if or else if statement.
- **Syntax**
- The syntax for a nested if statement is as follows

```
if(Boolean_expression 1) {
   // Executes when the Boolean expression 1 is true
      if(Boolean_expression 2) {
            // Executes when the Boolean expression 2 is true
   }
}
```

If Boolean Expression is true

```
if(Boolean_expression 1) {
   // Executes when the Boolean expression 1 is true
      if(Boolean_expression 2) {
            // Executes when the Boolean expression 2 is true
   }
}
```

Example of Nested if Statement

```
public class Test {
   public static void main(String[] args) {
     int x = 30;
     int y = 10;
     if( x == 30 ) {
        if( y == 10 ) {
          System.out.println("X = 30 and Y = 10");
        }
     }
   }
}
```

Example of Nested if Statement

```
public class Test {
   public static void main(String[] args) {
     int x = 30;
     int y = 10;
     if( x == 30 ) {
        if( y == 10 ) {
           System.out.println("X = 30 and Y = 10");
        }
     }
   }
}
```

When both if condition are true then execute this statement

This an outer if

This an inner if

Java if-else Statement

- The if-else statement also tests the condition.
- It executes the *if block* if condition is true otherwise execute *else block*.
- **Syntax:**

```
if(condition){
    //code if condition is true
}
else{
    //code if condition is false
}
```

if-else Statement Example

- For example, suppose the passing grade on an exam is 60. The pseudo code statement

If student 's grade is greater than or equal to 60
Print "**Passed**"
else
Print "**Failed**"

- Prints "Passed" if the student's grade is greater than or equal to 60, but prints "Failed" if the student's grade is less than 60. In either case , after printing occurs , the next pseudo code statement in sequence is "performed.

Program Example of if else Statement

```
12    public class MyFirstProgram {
13
14        public static void main(String[] args) {
15            int number=30;
              if (number<30) {
17
18                System.out.println("This is if statement");
19
20            }
21            else{
22
23                System.out.println("This is else statement");
24            }
25    }
26
27    }
```

myfirstprogram.MyFirstProgram > main > if (number < 30) >

Output - MyFirstProgram (run) ×

```
run:
```

Program Example of if else Statement

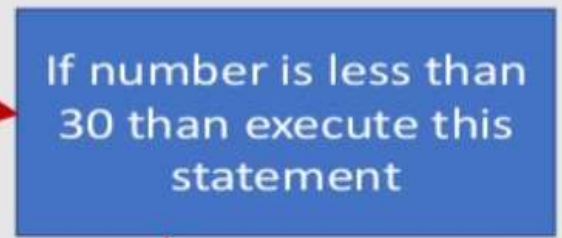
```
public class MyFirstProgram {

   public static void main(String[] args) {
      int number=30;
        if (number<30) {

          System.out.println("This is if statement");

        }
        else{

           System.out.println("This is else statement");
        }
    }
}
```



Output is:
This is else statement

Problem Statement

- Prompt user to enter a value. Check if the number is even then display a message that number is even. If number is odd then display message number is odd.

Solution of the Previous Problem

```
2  import java.util.Scanner;
3  public class MyFirstProgram {
4
5      public static void main(String[] args) {
6          int number;
7         Scanner user_input = new Scanner(System.in);
8         System.out.println("Please Enter a Number: ");
9        number=user_input.nextInt();
0          if(number%2==0)
1          {
2                  System.out.println((+number)+" is an Even Number");
3          }
4          else
5          {
           System.out.println((+number)+" is an odd Number");
7          }
8      }
9  }
```

Explanation of Previous Program

```
import java.util.Scanner;
public class MyFirstProgram {

   public static void main(String[] args) {
      int number;
     Scanner user_input = new Scanner(System.in);
     System.out.println("Please Enter a Number: ");
    number=user_input.nextInt();
     if(number%2==0) {
           System.out.println((+number)+" is an Even Number");
      }
      else {
       System.out.println((+number)+" is an odd Number");
      }
   }
}
```

Import Scanner Class to use its method to take input from user

Create an object using Scanner class

Take input from user

Checking condition

Output is:

Please Enter a Number:
10
10 is an Even Number

Problem Statement

Write a Java program which inputs the marks obtained by student. If the marks are equal to or greater than **50** then display the message that "**You are pass**." else "**Sorry you are fail**."

- Write a Java program to check whether a triangle is valid or not, when the three angles of the triangle are entered by the user. A triangle is valid if the sum of all the three angles is equal to 180 degrees.

```java
    public static void main(String[] args) {
        int angle_1,angle_2,angle_3;
       Scanner user_input = new Scanner(System.in);
       System.out.println("Please Enter First Angle: ");
        angle_1=user_input.nextInt();
        System.out.println("Please Enter Second Angle: ");
        angle_2=user_input.nextInt();
        System.out.println("Please Enter Third Angle: ");
        angle_3=user_input.nextInt();
        if(angle_1+angle_2+angle_3==180)
        {
              System.out.println("Triangle is Valid");
        }
        else
        {
              System.out.println("Trianle is Not Valid");
        }
    }
}
```

myfirstprogram.MyFirstProgram > main >

tput - MyFirstProgram (run) ×

```
run:
Please Enter First Angle:
60
Please Enter Second Angle:
60
Please Enter Third Angle:
60
Triangle is Valid
```

Any year is input by the user. Write a program to determine whether the year is a leap year or not.

Program Example if Nested if else Statement

```
12  public class Test {
13      public static void main(String[] args) {
14          int i=1; int j=4;
15        if (i==1) {
16            System.out.println("i am outer if");
                  if (j>4) {
18                    System.out.println("i am inner if");
19                }
20                else{
21                    System.out.println("i am inner else");
22                 }
23  }
24  else{
25            System.out.println("i am outer else");
26  }
27
28      }
29  }
```

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

The if-else-if ladder statement executes one condition from multiple statements.

Syntax:

```
if(condition1){
        //code to be executed if condition1 is true
}
else if(condition2){
        //code to be executed if condition2 is true
}
else if(condition3){
        //code to be executed if condition3 is true
}
...
else{
        //code to be executed if all the conditions are false
}
```

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

```
If student 's grade is greater than or equal to 90
        Print " A "
else If student 's grade is greater than or equal to 80
        Print " B "
else If student 's grade is greater than or equal to 70
        Print " C "
else If student 's grade is greater than or equal to 60
        Print " D "
else
        Print " F "
```

Program Example of if...else if...else Statement

```
14      public static void main(String[] args) {
15          int x = 30;
16
17          if(x == 10) {
18              System.out.println("Value of X is 10");
19          }
20          else if(x == 20) {
21              System.out.println("Value of X is 20");
22          }
23          else if(x == 30) {
24              System.out.println("Value of X is 30");
25          }
26          else {
27              System.out.println("This is else statement");
28          }
29      }
30      }
```

myfirstprogram.MyFirstProgram > main > if (x == 10) else if (x == 20) else if (x == 30) e

utput - MyFirstProgram (run) ×

Program Example of if...else if...else Statement

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

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

Problem Statement

- Write a program which prompt the user to enter a digit between **1**to **3** and display the digit entered by the user.

```
15      public static void main(String[] args) {
16          int number;
17          Scanner user_input = new Scanner(System.in);
18          System.out.println("Please Enter a Number: ");
19          number=user_input.nextInt();
            if(number==1)
21          {
22              System.out.println("You Pressed 1");
23          }
24          else if(number==2)
25          {
26              System.out.println("You Pressed 2");
27          }
28          else if(number==3)
29          {
30              System.out.println("You Pressed 3");
31          }
32          else
33          {
34              System.out.println("You Pressed other than 1,2,3");
35          }
36          }
```

Problem Statement

- A bank gives loan in **two** situations
  - 1) If customer is **male** and his **age** is more than 25 and his **salary** is more than 25000
  - 2) If customer is a **female** and her **age** is more than 30 and her **income** is 20000
- Program: Ask user to enter **age**, **gender(M/F)** and **income**. Check whether he\she is **eligible** to get loan.

```java
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        int age,income;char gender;
        Scanner input=new Scanner(System.in);
        System.out.println("Please Enter  Your Age: ");
        age=input.nextInt();
        System.out.println("Please Enter Your Income: ");
        income=input.nextInt();
        System.out.println("Please Enter  Your Gender(M/F): ");
        gender=input.next().charAt(0);
        if(age > 25 && gender == 'M'&& income > 25000){
            System.out.println("Sir, You can apply for Loan");
        }
        else if(age > 30 && gender == 'F' && income > 30000){
            System.out.println("Madam, You can apply for Loan");
        }
        else{
            System.out.println("Sorry, You are not eligible for loan");
        }
    }
}
```

Write a Java program to calculate the **monthly telephone** bills as per the following rule:

Minimum Rs. 200 for up to 100 calls.

Plus Rs. 0.60 per call for next 50 calls.

Plus Rs. 0.50 per call for next 50 calls.

Plus Rs. 0.40 per call for any call beyond 200 calls.

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

```
switch(expression) {
  case value :
    // Statements
    break;  // optional
  case value :
    // Statements
    break; // optional
  // You can have any number of case statements.
  default : // Optional
    // Statements
}
```

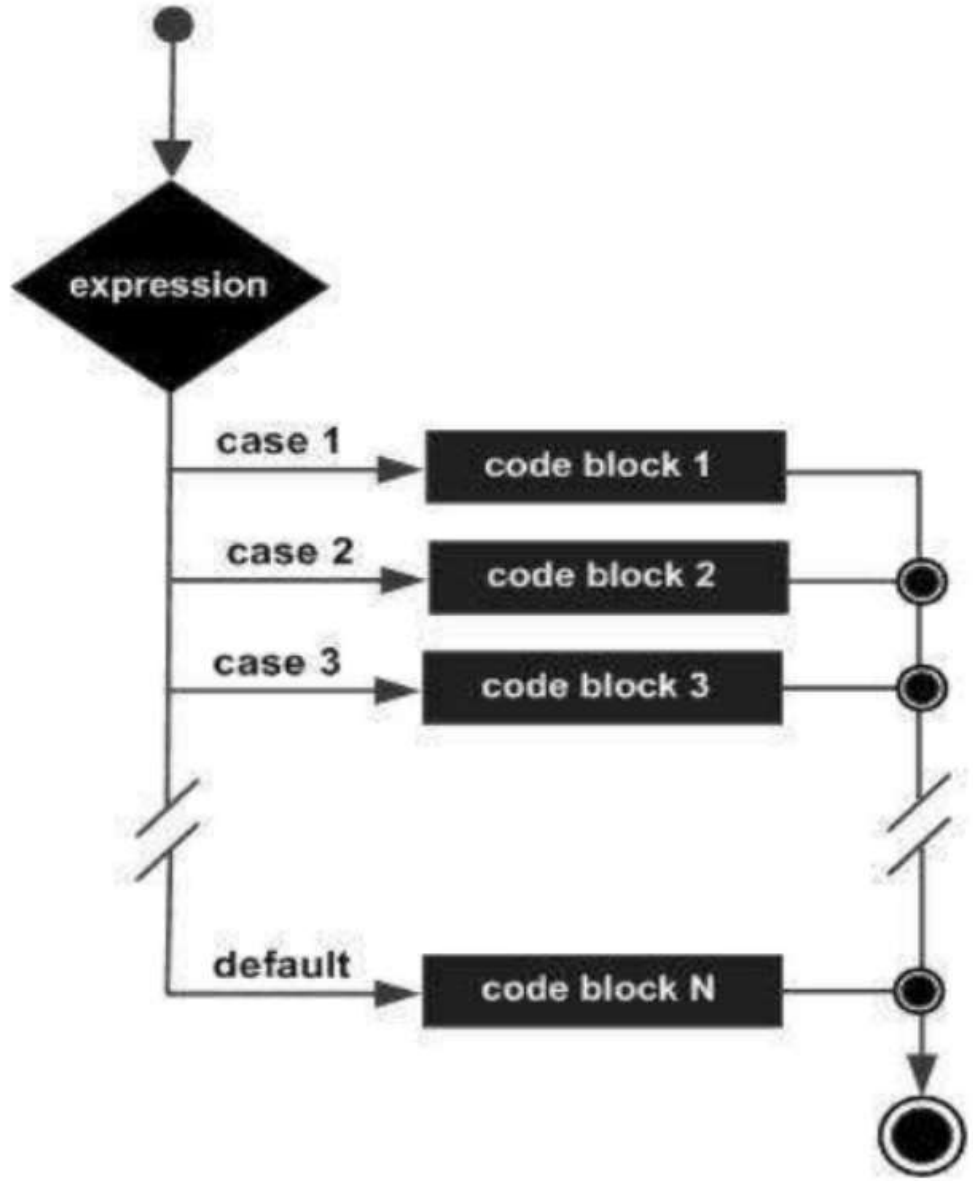
The following Rules Apply to a Switch Statement

- The variable used in a switch statement can only be integers, convertable integers (byte, short, char), strings and enums.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The value for a case must be the same data type as the variable in the switch and it must be a constant or a literal.

The following Rules Apply to a Switch Statement

- When the variable being switched on is equal to a case, the statements following that case will execute until a *break* statement is reached.
- When a ***break* statement** is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will *fall through* to subsequent cases until a break is reached.
- A *switch* statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Flowchart of Switch Statement



Program Example of Switch Statement

```
12  public class MyFirstProgram {
13
14      public static void main(String[] args) {
15
16      int number=20;
17      switch(number){
18      case 10:{   System.out.println("10");break;}
19      case 20:{   System.out.println("20");break; }
20      case 30:{ System.out.println("30");break; }
21      default:{   System.out.println("Not in 10, 20 or 30"); }
22          }
23      }
24  }
```

Simple Calculator Using Switch Case Statement In Java.....

```
import java.util.Scanner;
public class MyFirstProgram {


   public static void main(String[] args) {
      Scanner s=new Scanner(System.in);
      System.out.println("You Have Following Choices : ");
      System.out.println("1. Addition");
      System.out.println("2. Subtraction ");
      System.out.println("3. Multiplication ");
      System.out.println("4. Division");
      System.out.println("Enter Your Choice : ");
      int i=s.nextInt();
```

Simple Calculator Using Switch Case Statement In Java.....

```
System.out.println("Enter First Number ");
int a=s.nextInt();
System.out.println("Enter Second Number ");
int b=s.nextInt();
//result' will store the result of operation
double result=0;
```

Simple Calculator Using Switch Case Statement In Java.....

```
switch(i)
      {
         case 1:
            result=a+b;
            break;
         case 2:
            result=a-b;
            break;
         case 3:
            result=a*b;
            break;
```

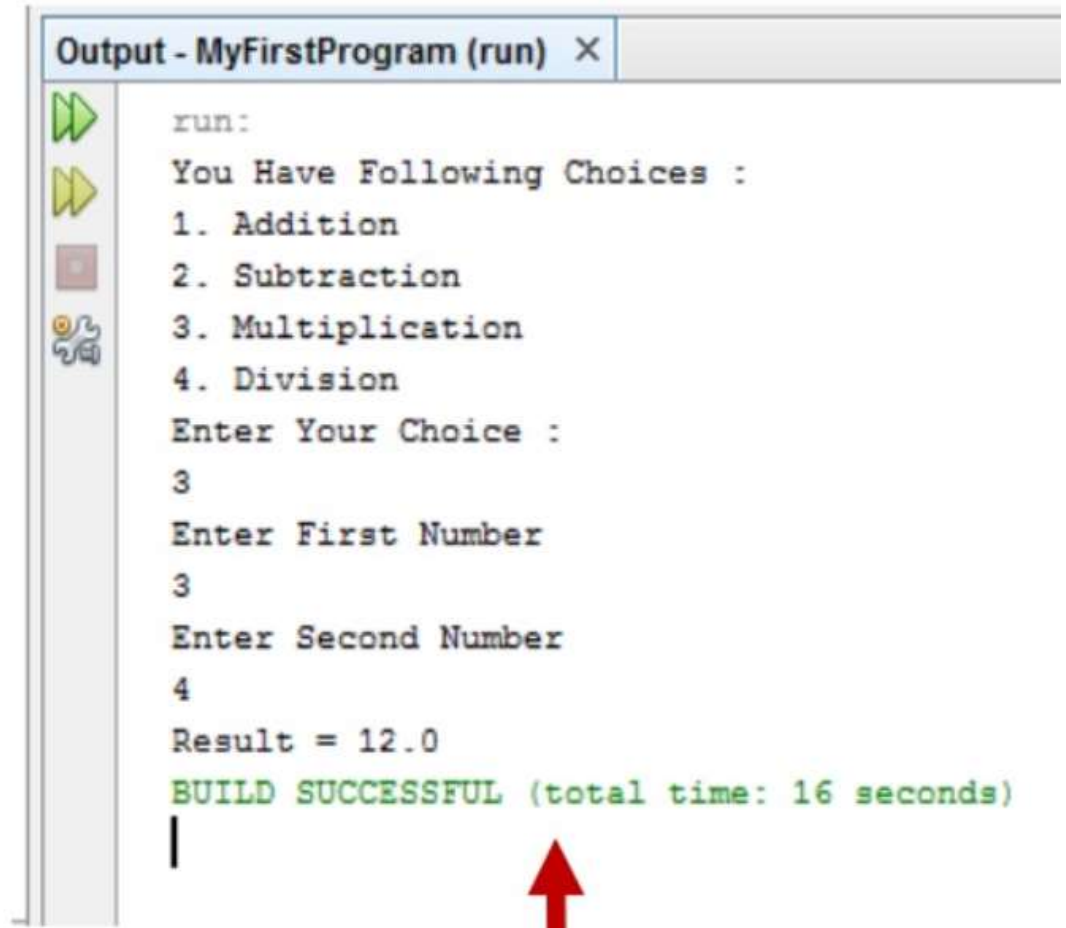
Simple Calculator Using Switch Case Statement In Java.....

```
case 4:
            if(b==0) {    //when denominator becomes zero
               System.out.println("Division Not Possible");
               break;
            }
            else
               result=a/b;
          default:
            System.out.println("You Have Entered A Wrong Choice");
       }
       System.out.println("Result = "+result);

    }
}
```

Outputs of Previous Program is :

You Have Following Choices :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter Your Choice :
3
Enter First Number
3
Enter Second Number
4
Result = 12.0



Compiler View Output