//Program to read the name from the keyboard by using Scanner class :

---

* java.lang is a **default package** so we need not to import this package that means all the predefined classes which are available in java.lang package we can use every where in java.

* Scanner class is available in **java.util** package so, At the time of using Scanner class we need to import the Scanner class with the help of import keyword.

ReadName.java

```
import java.util.Scanner;

public class ReadName
{
   public static void main(String[] args)
   {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your Name :");
        String name = sc.nextLine();
        System.out.println("Your Name is :"+name);
   }
}
```

//WAP to read employee data from Scanner class

```
package com.ravi.basic;

import java.util.*;

public class ReadEmployeeData
{
   public static void main(String[] args)
   {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the Employee Id :");
        int id = Integer.parseInt(sc.nextLine()); //Buffer Problem

        System.out.print("Enter the Employee Name :");
        String name = sc.nextLine();


        System.out.print("Enter the Employee Salary :");
        double salary = sc.nextDouble();


        System.out.println("Employee Id is :"+id);
        System.out.println("Employee Name is :"+name);
        System.out.println("Employee Salary is :"+salary);

   }

}
```

WAP to read a character from Scanner class :

```
public class ReadChar
{
   public static void main(String[] args)
   {
      java.util.Scanner sc = new java.util.Scanner(System.in);
      System.out.print("Enter your Gender [M/F] :");
      char gender = sc.next().charAt(0);
      System.out.println("Your Gender is :"+gender);
      sc.close();
   }
}
```

---

5) Opeartors :

It is a symbol which describes that how a calculation will be performed on operands.

Types Of Operators :

1) Arithmetic Operator (Binary Operator)

2) Unary Operators

3) Assignment Operator

4) Relational Operator

5) Logical Operators (&& || !)

6) Boolean Operators (& |)

7) Bitwise Operators (^ ~)

8) Ternary Operator

*9) Member Operator(Dot . Operator)

*10) new Operator

*11) instanceof Operator [It is also relational operator]

Basic Concepts of Operators :

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 15;
        int y = x++;
        System.out.println(x + " : "+ y);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 15;
        int y = --x;
        System.out.println(x + " : "+ y);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 20;
        int y = ++20;
        System.out.println(x + " : "+ y);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 20;
        int y = ++(++x);
        System.out.println(x + " : "+ y);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     char ch = 'A';
        ch++;
        System.out.println(ch);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     double d1 = 12.89;
        d1++;
        System.out.println(d1);  //13.89
   }
}
```

Note : Increment and decrement operator we can apply with any data type except boolean

---

```
public class Test
{
   public static void main(String[] args)
   {
     while(false)
        {
            System.out.println("Hello");
        }
        System.out.println("World");
   }
}
```

Note : CE i.e unrechable statement

```
public class Test
{
   public static void main(String[] args)
   {
     boolean b = false;
     while(b)
        {
            System.out.println("Hello");
        }
        System.out.println("World");
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     final int x = 10;
     final int y = 20;

        while(x > y)
        {
            System.out.println("x is Greater than y");
        }
        System.out.println("Hello World");
   }
}
```

Note : CE

```
public class Test
{
   public static void main(String[] args)
   {
     do
        {
            int x = 1;
            System.out.println(x);
            x++;
        }
        while (x<=10);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 1;
        do
        {
            System.out.println("Hello");
            x++;
            if(x==5)
            {
                break;
            }

        }
        while (x<=10);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 1;
        do
        {
            System.out.println("Hello");
            x++;

            boolean b = false;
            if(b = true)
            {
                break;
            }
        }
        while (x<=10);
   }
}
```

Numeric Promotion :

* While working with Arithmetic Opeartor (+, -, *, /, %) OR Unary minus opertaor, after expression execution the minimum data type required is **int** (32 bits) OR in other other the result will be promoted to int type so, to store the result minimum int data type is required.

* While calculation byte, short and char are promoted to int type.

* The main purpose of this numberc promotion to prevent the data from data loss so to represent the result minimum 32 bits are required.

//Program on Unary Minus Operator :

```
public class Test
{
   public static void main(String[] args)
   {
     int x = 15;
        System.out.println(-x);

   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
        byte b = 100;
        byte c = 100;
        short d = b + c; //error
        System.out.println(d);

   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
        short b = 1;
        short c = 5;
        short d = b * c; //error
        System.out.println(d);
   }
}
```

```
public class Test
{
   public static void main(String[] args)
   {
        byte b = 1;
        byte c = -b; //error
        System.out.println(c);

   }
}
```

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


        /*
        byte b = 1;
        b = b + 2;
        System.out.println(b);
        */

        byte b = 2;
        b += 2;   //It is valid because short hand operator
        System.out.println(b);

   }
}
```