



Concept of encapsulation

- The object is at the core of Java programming.
- Java provides the **concept of class** to build objects.
- A class defines the shape and working of an object.
- The concept of class is the logical construct upon which the entire Java language is built.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHARAGPUR



What is a class?

- A **class** is a group of objects, which have common properties.
- It is a **template** or blueprint from which objects are created.
- It is a logical entity.

A class in Java can contain:

- Fields
- Methods
- Constructors
- Blocks
- Nested class(es) and interface(s)



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE

IIT KHAI



General structure of a class

```
class <class-name>{
    <type> <variable 1>;
    <type> <variable 2>;
    <type> <variable 3>;
    ...
    <type> <variable n>;
}

<type> <method 1>(<parameter-list 1>) {
    // Body of the method 1
}
<type> <method 1>(<parameter-list 2>) {
    // Body of the method 2
}
...
<type> <method 1>(<parameter-list m>) {
    // Body of the method m
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANT

CSE





Adding methods to Circle class

```
class circle {  
    double x,y; // The coordinates of the center  
    double r;   // The radius  
  
    // Method that returns circumference  
    double circumference(){  
        return 2*3.14159*r;  
    }  
    // Method that returns area  
    double area(){  
        return (22/7)*r*r;  
    }  
}
```



Declaring object of type Circle class

```
// A program that uses the circle class
// Call this file circledemo1.java
class Circle {
    double x,y; // The coordinates of the center
    double r;   // The radius

    // Method that returns circumference
    double circumference(){
        return 2*3.14159*r;
    }
    // Method that returns area
    double area(){
        return (22/7)*r*r;
    }
}
```

```
//The following class declare an object of type Circle
class CircleDemo1 {
    public static void main(String args[]){
        Circle c = new Circle();
        c.x = 0.0;
        c.y = 0.0;
        c.r = 5.0;
        System.out.println("Circumference" + c.circumference());
        System.out.println("Area" + c.area());
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHAI



Multiple objects declaration

```
// A program that declares two objects of the Circle class
// Call this file CircleDemo2.java
class Circle {
    double x, y;
    double r;

    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
}
```

```
//The following class declares multiple objects of type Circle
class CircleDemo2 {
    public static void main(String args[]){
        Circle c1 = new Circle();
        Circle c2 = new Circle();
        // Initialize the circles
        c1.x = 3.0;
        c1.y = 4.0;
        c1.r = 5.0;
        c2.x = -4.0;
        c2.y = -8.0;
        c2.r = 10.0;
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
    }
}
```

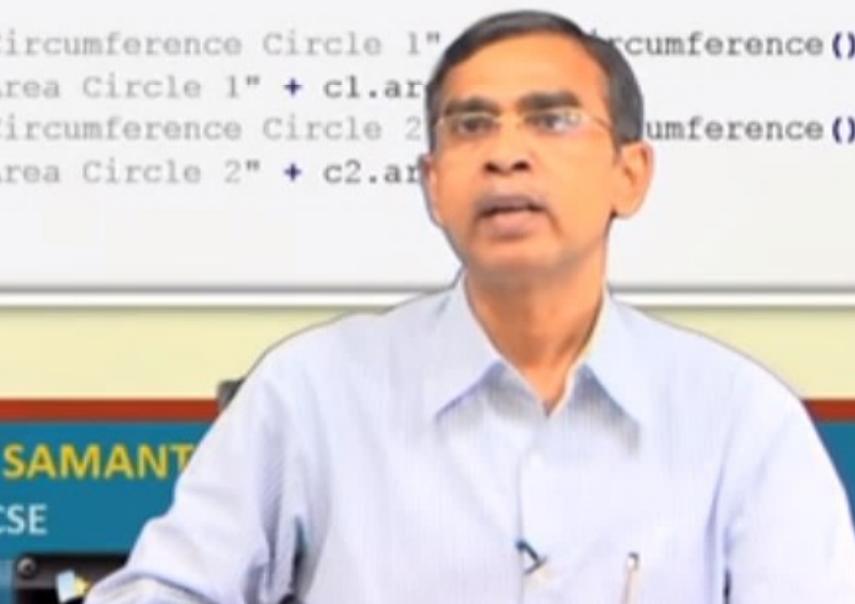


IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANT

CSE





Multiple Classes Declaration



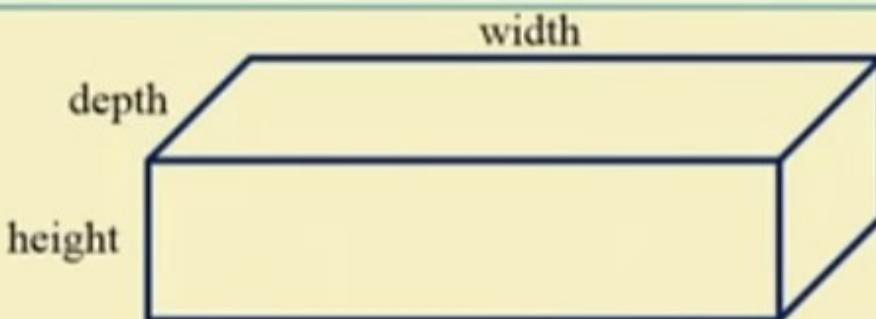
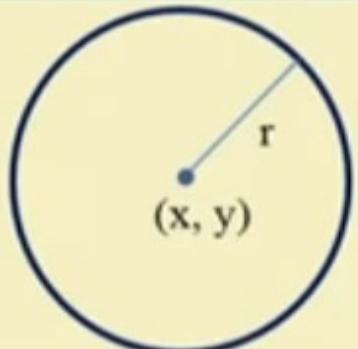
IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANT
CSE
IIT KHARAGPUR

Multiple classes in a program



```
class Circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
}
```

```
class Box{
    double width;
    double height;
    double depth;
    double area(){
        double a;
        a = (width*height + height*depth + width*depth) * 2;
        return a;
    }
    double volume(){
        double v;
        v = width*height*depth;
        return v;
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMAN

CSE





Multiple class objects

```
// Declaring objects of type Circle and Box
class MulticlassDemo {
    public static void main(String args[]){
        Circle c = new Circle();
        Box b = new Box();
        // Initialize the circles
        c.x = 3.0; c.y = 4.0; c.r = 5.0;
        b.width = 3.0; b.height = 4.0; b.depth = 5.0;
        System.out.println("Circumference Circle" + c.circumference());
        System.out.println("Area Circle" + c.area());
        System.out.println("Area of Box" + b.area());
        System.out.println("Volume of Box" + b.volume());
    }
}
// Save this file as MulticlassDemo.java
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANT
CSE
IIT KHARAGPUR



Important notes

1. There should be a class which contains a method `main()`. This class is called **main class**.
2. There should be only **one** main class.
3. The name of the program file should be same as the name of the main class followed by `.java` as an extension.
4. If there is no main class, then there should be compilation error.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMAN
CSE
IIT KHA



Java program without main class

```

class Circle {
    double x, y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
}
class Box{
    double width;
    double height;
    double depth;
    double area(){
        double a;
        a = (width*height + height*depth + width*depth) * 2;
        return a;
    }
    double volume(){
        double v;
        v = width*height*depth;
        return v;
    }
}

```

Name the file as [Test.java](#).

This program reports compilation error as follows.

Error: Main method not found in class
Circle, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend
javafx.application.Application

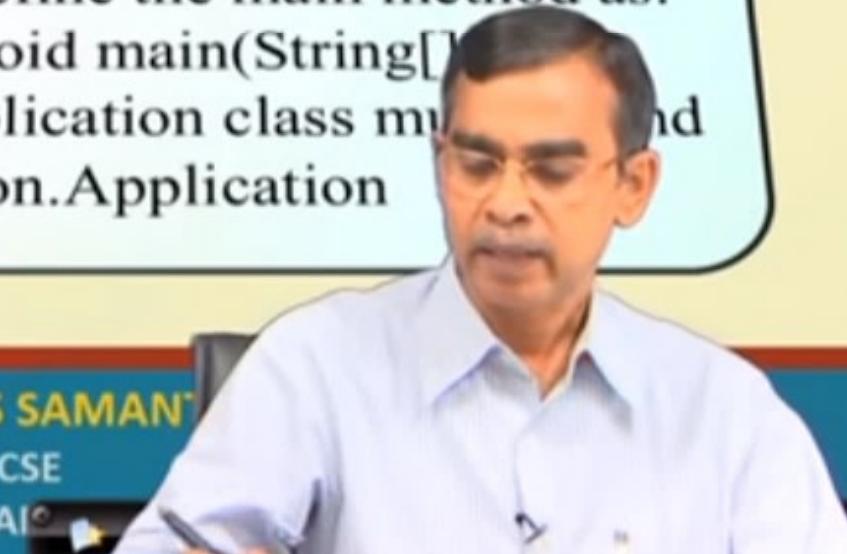


IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANIEGO
CSE
IIT KHARAGPUR





Method with parameters

```
class circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
    void setCircle(double a, double b, double c){
        x = a; // Set center x-coordinate
        y = b; // Set center y-coordinate
        r = c; // Set radius
    }
}
```

```
class CircleDemo3 {
    public static void main(String args[]){
        Circle c1 = new Circle();
        Circle c2 = new Circle();
        // Initialize the circles
        c1.setCircle(3.0,4.0,5.0);
        c2.setCircle(-4.0,8.0,10.0);
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area of circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area of circle 2" + c2.area());
    }
}
```

Name the file as [CircleDemo3.java](#)



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANIE
CSE
IIT KHARAGPUR





Constructors



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA
CSE
IIT KHARAGPUR



Constructor for automatic object initialization

1. It can be tedious to initialize all of the variables in a class each time an object is instantiated.
2. Java allows objects to initialize themselves when they are created/declared.
3. This automatic initialization is performed through the **concept of constructor**.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANIEGO
CSE
IIT KHARAGPUR



Constructors – Some properties

1. A constructor **initializes an object** immediately upon creation.
2. Constructor in Java is a **method**.
3. This method has the **same name** as the class in which it resides.
4. Once defined, the constructor is **automatically called** immediately after object is created.
5. Constructor is a method which has **no return type**.
6. In fact, the implicit return type of a class constructor is the class type itself.
7. Constructor initialize the internal state of an object.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANIE
CSE
IIT KHARAGPUR



Constructor : An example

```
class Circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }

    Circle (double a, double b, double c){
        x = a; // Set center x-coordinate
        y = b; // Set center y-coordinate
        r = c; // Set radius
    }
}
```

```
class CircleDemo4 {
    public static void main(String args[]){
        Circle c1 = new Circle(3.0,4.0,5.0);
        Circle c2 = new Circle(-4.0,8.0,10.0);
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE



The this Keyword



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE

IIT KHAI





this keyword concept

1. `this` is used to reduce name-space collisions.
2. Sometimes a method will need to refer to the object that invoked it.
3. To allow this Java defines `this` keyword.
4. `this` can be used inside any method to refer to the current object.
5. That is, `this` is always a reference to the object on which the method is invoked.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHAI

Constructor : An example

```
class Circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
    void setCircle(double a, double b, double c){
        x = a; // Set center x-coordinate
        y = b; // Set center y-coordinate
        r = c; // Set radius
    }
}
```

```
class Circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
    Circle (double x, double y, double r){
        this.x = x; // Set center x-coordinate
        this.y = y; // Set center y-coordinate
        this.r = r; // Set radius
    }
}
```

```
class CircleDemo5 {
    public static void main(String args[]){
        Circle c1 = new Circle();
        c1.setCircle(3.0,4.0,5.0);
        Circle c2 = new Circle (-4.0,8.0,10.0);
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area())
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANT
CSE



Multiple constructors

1. Sometimes, it is necessary to initialize an object in a number of ways.
2. Java allows his using the concept of **constructor overloading**.
3. In other words, Java allows to declare one or more constructor method with **different lists of parameters** and **different method definition**.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE



Constructor overloading : An example

```
class Circle {
    double x,y;
    double r;
    Circle (double x, double y, double r){
        this.x = x; this.y = y; this.r = r;
    }
    Circle (double r){
        x = 0; y=0; this.r = r;
    }
    Circle (Circle c){
        x = c.x; y = c.y; r = c.r;
    }
    Circle (){
        x = 0.0; y = 0.0; r = 1.0;
    }
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return (22/7)*r*r;
    }
}
```

```
class CircleDemo6 {
    public static void main(String args[]){
        Circle c1 = new Circle(3.0,4.0,5.0);
        Circle c2 = new Circle(5.0);
        Circle c3 = new Circle(c1);
        Circle c4 = new Circle();
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
        System.out.println("Circumference Circle 3" + c3.circumference());
        System.out.println("Area Circle 3" + c3.area());
        System.out.println("Circumference Circle 4" + c4.circumference());
        System.out.println("Area Circle 4" + c4.area());
    }
}
```

Name the file as [CircleDemo6.java](#)



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE



this keyword again

1. There is a specialized use of **this** keyword that arises when a class has multiple constructors.
2. In that case, **this** can be used from one constructor to invoke one of the other constructor of the same class.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHARAGPUR

this with multiple constructors : An example

```
class Circle {  
    double x, y;  
    double r;  
    Circle (double x, double y, double r){  
        this.x = x; this.y = y; this.r = r;  
    }  
    Circle (double r){  
        this(0.0, 0.0, r);  
    }  
    Circle (Circle c){  
        this(c.x, c.y, c.r);  
    }  
    Circle (){  
        this(0.0, 0.0, 1.0);  
    }  
    double circumference(){  
        return 2*3.14159*r;  
    }  
    double area(){  
        return (22/7)*r*r;  
    }  
}
```

Note:

1. There is a very important restriction on the `this` syntax.
2. It should appear only as the first statement in a constructor.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE



Questions to think...

- How Java developer makes the language simple compared to other language?
- Are there any specific things in Java, which a programmer should aware about those?



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA
CSE
IIT KHAI





About main () method



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA
CSE
IIT KHAI





Analysis of the program

Let us examine each statement step-by-step.

Import Statements



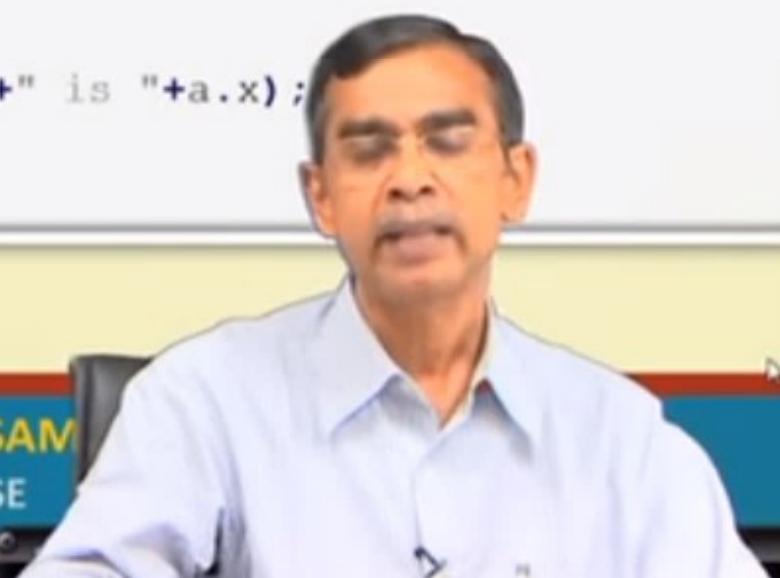
Declaration of class



Declaration of main class



```
import java.lang.*;  
  
class Calculator{  
    double i;  
    double x = Math.sqrt(i);  
}  
  
class Example{  
    public static void main(String args[]){  
        Calculator a = new Calculator();  
        a.i = 20;  
        System.out.println("Square root of "+a.i+" is "+a.x);  
    }  
}
```



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA
CSE
IIT KHARAGPUR



Significance of main class

- Java program starts its execution from a method belongs to a class only.
- The **main()** method is the starting point of the execution of the main thread.
- If there are multiple classes, then ambiguity is resolved by incorporating a main() method into only one special class called main class.
- The name of the Java program should be named after this class so that Java interpreter unanimously choose that class to start its execution.

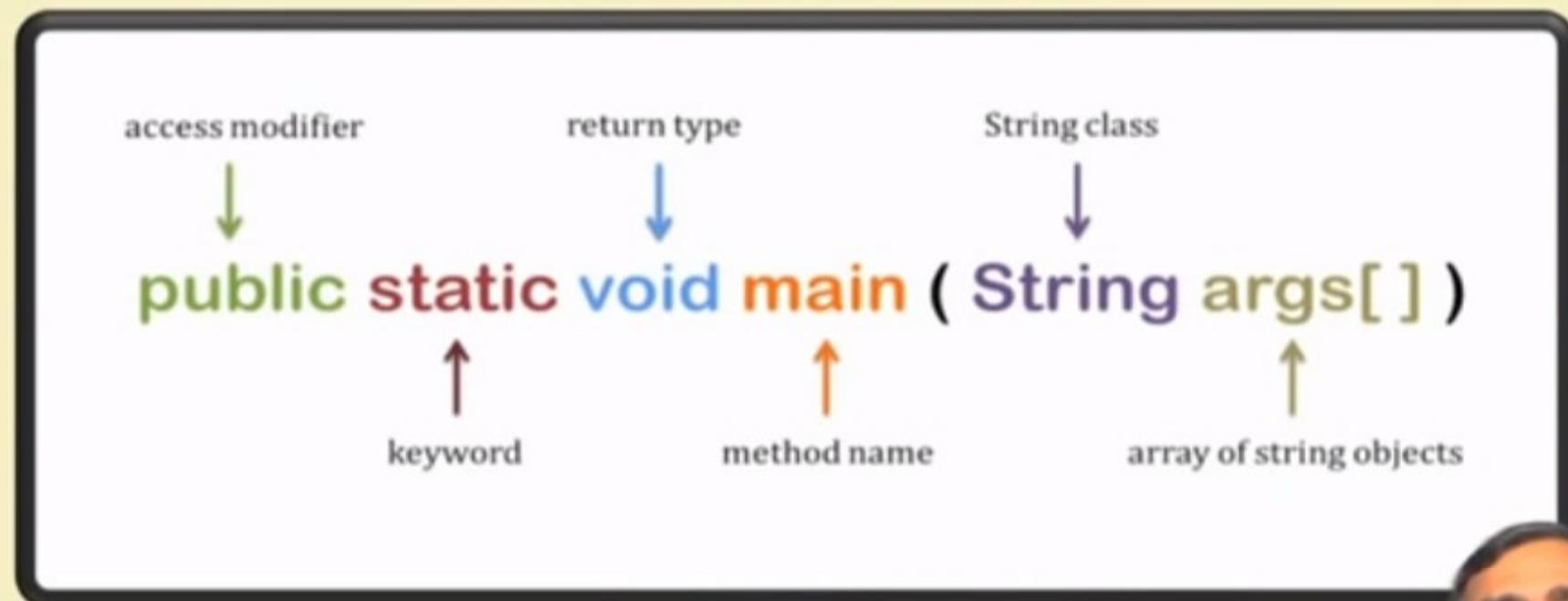


IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHAI



Understanding basic Java syntax



Java `main()` method



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHARAGPUR



static keyword

```
access modifier      return type      String class  
↓                ↓                  ↓  
public static void main ( String args[] )  
↑                ↑                  ↑  
keyword          method name    array of string objects
```

Java main() method

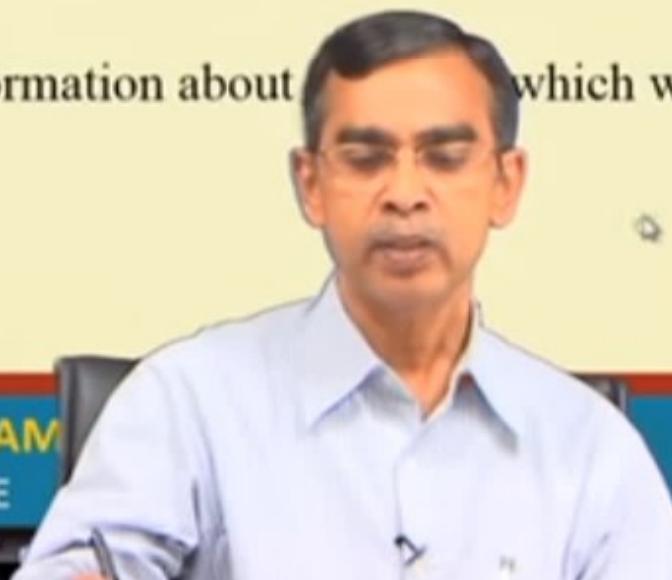
➤ static

- The keyword **static** allows **main()** to be called without having to instantiate a particular instance of the class.
- This is necessary since **main()** is called by Java interpreter before any objects are made.

Note: There are more information about **static** which will be discussed shortly.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAM
CSE
IIT KHA



void keyword

```
access modifier      return type      String class
↓                  ↓                  ↓
public static void main ( String args[])
↑                  ↑                  ↑
keyword           method name       array of string objects
```

➤ void

- As per the Java programming language paradigm, each method should return a value; if it does not return anything, then the return type should be **void**.
- The keyword **void** simply tells the compiler that **main()** does not return any value after its execution.

Java main() method



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHARAGPUR



main() method

```
access modifier      return type      String class  
↓                ↓                  ↓  
public static void main ( String args[] )  
↑                ↑                  ↑  
keyword          method name    array of string objects
```

➤ main

- **main** is the name of a method in a class.
- This method is searched by **JVM** as a starting point for an application with a particular signature only.

Java main() method



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAMANTA
CSE
IIT KHARAGPUR



public Keyword

```

access modifier      return type      String class
↓                  ↓                  ↓
public static void main ( String args[] )
↑                  ↑                  ↑
keyword            method name     array of string objects
  
```

Java main() method

➤ public

- It is an access specifier, which allows the programmer to control the visibility of class members.
- **public** member may be accessed by code outside the class in which it is declared.
- **main()** must be declared as public, since it must be called by code outside of its class when the program is started.

Note: By default a member is **public**.

Other access specifiers will be discussed later.

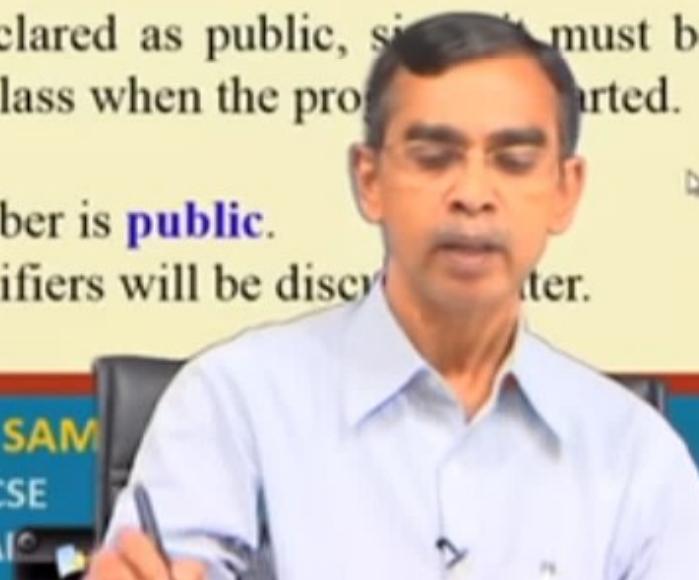


IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHARAGPUR



Arguments in main()

```

access modifier      return type      String class
↓                  ↓                  ↓
public static void main ( String args[])
↑                  ↑                  ↑
keyword            method name     array of string objects
  
```

Java main() method

➤ String args[]

- Here, **String** is a class defined in **java.lang API**.
- **args []** is an **array** to store objects of class **String**.
- Here, you could write anything, say **String x[]** instead of **String args[]**. **args[]** is a common practice that every programmer uses. It is a customary.
- Java sees everything as **String** objects.
- It will help to read an input and then store it into the array **args[]** as **String** objects.



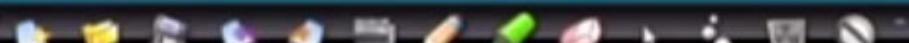
IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE

IIT KHAI



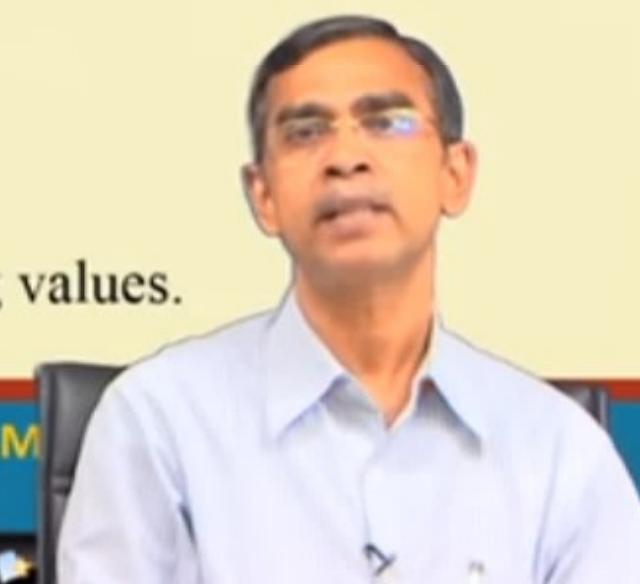


Output from Java program

Statement 12 includes the following code

```
System.out.println("Square root of "+a.i+" is "+a.x);
```

- **System** is a final class from the **java.lang** package.
- **out** is a class variable of type **PrintStream** declared in the **System** class.
- **println** is a method of the **PrintStream** class.
- **a.i** and **a.x** represents the names of variables to be printed.
- **+** is a concatenation operator, it is used to concatenate the string values.



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHA





print versus println methods

Consider the following lines to be printed as output

Debasis

Samanta

This can be done using both **println()** and **print()** functions

```
System.out.println("Debasis");
System.out.println("Samanta");
```

```
System.out.print("Debasis");
System.out.print("\n");
System.out.print("Samanta");
```

- The **println("...")** method prints the string "..." and moves the cursor to a new line.
- The **print("...")** method instead prints just the string "...", but does not move the cursor to a new line. Hence, subsequent printing instructions will print on the same line.

Note: The **println()** method can also be used without parameters, to position the cursor on the next line.



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE





Command line input in Java

Let us run this Java program

```
public class Echo{  
    public static void main(String args[]){  
        for(int i=0;i<args.length;i++){  
            System.out.print(args[i]+" ");  
            System.out.print("\n");  
        }  
        System.exit(0);  
    }  
}
```

```
C:\Users\Desktop\Java\Echo>Hi Debasis Samanta  
Hi  
Debasis  
Samanta
```



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHARAGPUR





Command line input in Java

Let us run the same Java program with different input:

```
public class Echo{  
    public static void main(String args[]){  
        for(int i=0;i<args.length;i++){  
            System.out.print(args[i]+" ");  
            System.out.print("\n");  
        }  
        System.exit(0);  
    }  
}
```

C:\Users\Desktop\Java\Echo>1 2 3 4 5 6 7

1
2
3
4
5
6



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM

CSB





Practice another Java program

Let us run this Java program:

```
public class UserArgument{  
    public static void main(String args[]){  
        System.out.print("Hi ");  
        System.out.print(args[0]);  
        System.out.print(", How are you?");  
    }  
}
```

C:\Users\Desktop\Java\UserArguement>Debasis
Hi Debasis, How are you?



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHAR





Numeric input to program

Let us run this Java program:

```
import java.lang.*;  
  
class Calculator{  
    double i;  
    double x = Math.sqrt(i);  
}  
class Example{  
    public static void main(String args[]){  
        Calculator a = new Calculator();  
        a.i = Integer.parseInt(args[0]);  
        System.out.println("Square root of "+a.i+" is "+a.x);  
    }  
}
```

C:\Users\Desktop\Java\Calculator>56
Square root of 56 is 7.483319234678



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM
CSE
IIT KHAR





Input to Java program with Scanner Class

```

import java.util.Scanner ← 1. Import Scanner Class

public class ScannerDemo
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in); ← 2. Construct Scanner class Object
        System.out.println("Enter first no= ");

        int num1, num2; ← 3. Define Variable to Receive Input

        num1=s.nextInt(); ←
        System.out.println("Enter 2nd no "); ← 4. Read Input from Keyboard
        num2=s.nextInt(); ←

        System.out.println("Sum of no is= "+(num1+num2));
    }
}

```

Scanner is one of the predefined class which is used for reading the data dynamically from the keyboard.



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAM
CSE
IIT KHAR

Example program for Scanner : Maximum

```
import java.util.Scanner;

public class MaximumCalculator {
    public static void main(String args[]) {
        Scanner scnr = new Scanner(System.in);
        // Calculating the maximum two numbers in Java
        System.out.println("Please enter two numbers to find maximum of two");
        int a = scnr.nextInt();
        int b = scnr.nextInt();
        if (a > b) {
            System.out.printf("Between %d and %d, maximum is %d \n", a, b, a);
        }
        else {
            System.out.printf("Between %d and %d, maximum number is %d \n", a, b, b);
        }
    }
}
```



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE

IIT KHARAGPUR





Example program with Scanner and array

```
import java.util.*;
class SimpleArrayList{
public static void main(String args[]){
    int sum = 0;
    float avg = 0;
    ArrayList <Integer> l = new ArrayList<Integer>();
    System.out.println("Enter the input ");
    Scanner input = new Scanner(System.in);
    while (input.hasNextInt()) {
        l.add(input.nextInt());
    }
    for (int i = 0; i < l.size(); i++) {
        sum = sum+l.get(i);
    }
    avg = sum/(l.size());
    System.out.println("Average : " + avg);
}
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM

CSE

IIT KHARAGPUR



Example program with Scanner and array

```
import java.util.*;
class SimpleArrayList{
public static void main(String args[]){
    int sum = 0;
    float avg = 0;
    ArrayList <Integer> l = new ArrayList<Integer>();
    System.out.println("Enter the input ");
    Scanner input = new Scanner(System.in);
    while (input.hasNextInt()) {
        l.add(input.nextInt());
    }
    for (int i = 0; i < l.size(); i++) {
        sum = sum+l.get(i);
    }
    avg = sum/(l.size());
    System.out.println("Average : " + avg);
}
}
```

```
C:\Users\Desktop\Java\SimpleArrayList>Ent
er the input
5
6
4^Z
Average : 5.0
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM

CSE





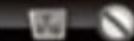
Input with DataInputStream : Calculator Program

```
import java.io.*;

class InterestCalculator{
    public static void main(String args[ ] ) {
        Float principalAmount = new Float(0);
        Float rateOfInterest = new Float(0);
        int numberOfYears = 0;
        DataInputStream in = new DataInputStream(System.in);
        String tempString;
        System.out.println("Enter Principal Amount: ");
        System.out.flush();
        tempString = in.readLine();
        principalAmount = Float.valueOf(tempString);
        System.out.println("Enter Rate of Interest: ");
        System.out.flush();
        tempString = in.readLine();
        rateOfInterest = Float.valueOf(tempString);
        System.out.println("Enter Number of Years: ");
        System.out.flush();
        tempString = in.readLine();
        numberOfYears = Integer.parseInt(tempString);
        // Input is over: calculate the interest
        float interestTotal = principalAmount*rateOfInterest*numberOfYears;
        System.out.println("Total Interest = " + interestTotal);
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSESDEBASIS SAM
CSE
IIT KHAR

Input with DataInputStream : Calculator Program

```
import java.io.*;  
  
class InterestCalculator{  
    public static void main(String args[ ] ) {  
        Float principalAmount = new Float(0);  
        Float rateOfInterest = new Float(0);  
        int numberOfYears = 0;  
        DataInputStream in = new DataInputStream(System.in);  
        String tempString;  
        System.out.println("Enter Principal Amount: ");  
        System.out.flush();  
        tempString = in.readLine();  
        principalAmount = Float.valueOf(tempString);  
        System.out.println("Enter Rate of Interest: ");  
        System.out.flush();  
        tempString = in.readLine();  
        rateOfInterest = Float.valueOf(tempString);  
        System.out.println("Enter Number of Years: ");  
        System.out.flush();  
        tempString = in.readLine();  
        numberOfYears = Integer.parseInt(tempString);  
        // Input is over: calculate the interest  
        float interestTotal = principalAmount*rateOfInterest*numberOfYears;  
        System.out.println("Total Interest = " + interestTotal);  
    }  
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAM

CSE





Input with DataInputStream : Calculator Program

```
import java.io.*;  
  
class InterestCalculator{  
    public static void main(String args[ ] ) {  
        Float principalAmount = new Float(0);  
        Float rateOfInterest = new Float(0);  
        int numberOfYears = 0;  
        DataInputStream in = new DataInputStream(System.in);  
        String tempString;  
        System.out.println("Enter Principal Amount: ");  
        System.out.flush();  
        tempString = in.readLine();  
        principalAmount = Float.valueOf(tempString);  
        System.out.println("Enter Rate of Interest: ");  
        System.out.flush();  
        tempString = in.readLine();  
        rateOfInterest = Float.valueOf(tempString);  
        System.out.println("Enter Number of Years: ");  
        System.out.flush();  
        tempString = in.readLine();  
        numberOfYears = Integer.parseInt(tempString);  
        // Input is over: calculate the interest  
        float interestTotal = principalAmount*rateOfInterest*numberOfYears;  
        System.out.println("Total Interest = " + interestTotal);  
    }  
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTHA

CSE

IIT KHARAGPUR



Input with DataInputStream : Calculator Program

```
import java.io.*;

class InterestCalculator{
    public static void main(String args[ ] ) {
        Float principalAmount = new Float(0);
        Float rateOfInterest = new Float(0);
        int numberOfYears = 0;
        DataInputStream in = new DataInputStream(System.in);
        String tempString;
        System.out.println("Enter Principal Amount: ");
        System.out.flush();
        tempString = in.readLine();
        principalAmount = Float.valueOf(tempString);
        System.out.println("Enter Rate of Interest: ");
        System.out.flush();
        tempString = in.readLine();
        rateOfInterest = Float.valueOf(tempString);
        System.out.println("Enter Number of Years: ");
        System.out.flush();
        tempString = in.readLine();
        numberOfYears = Integer.parseInt(tempString);
        // Input is over: calculate the interest
        float interestTotal = principalAmount*rateOfInterest*numberOfYears;
        System.out.println("Total Interest = " + interestTotal);
    }
}
```



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTHA

CSE

IIT KHARAGPUR





Input with DataInputStream : Calculator Program

```

import java.io.*;

class InterestCalculator{
    public static void main(String args[ ] ) {
        Float principalAmount = new Float(0);
        Float rateOfInterest = new Float(0);
        int numberOfYears = 0;
        DataInputStream in = new DataInputStream(System.in);
        String tempString;
        System.out.println("Enter Principal Amount: ");
        System.out.flush();
        tempString = in.readLine();
        principalAmount = Float.valueOf(tempString);
        System.out.println("Enter Rate of Interest: ");
        System.out.flush();
        tempString = in.readLine();
        rateOfInterest = Float.valueOf(tempString);
        System.out.println("Enter Number of Years: ");
        System.out.flush();
        tempString = in.readLine();
        numberOfYears = Integer.parseInt(tempString);
        // Input is over: calculate the interest
        float interestTotal = principalAmount*rateOfInterest*numberOfYears;
        System.out.println("Total Interest = " + interestTotal);
    }
}

```

C:\Users\Desktop\Java\InterestCalculator>
 Enter Principal Amount:
 100.0
 Enter Rate of Interest:
 12.5
 Enter Number of Years:
 2
 Total Interest = 25.0



IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA

CSE

IIT KHAR





In today's demonstration

1. Java programs to demonstrate use of *print()*, *println()* and *printf()*.
2. Java program to demonstrate *Command line input* in Java
3. Programs demonstrate how to take Input using scanner class
4. Program to show how to read input with *DataInputStream*

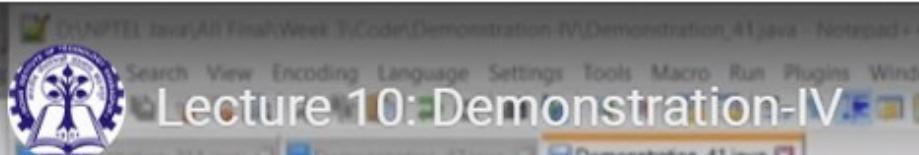


IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

DEBASIS SAMANTA
CSE
IIT KHARAGPUR



Lecture 10: Demonstration-IV

```
1 // A Java program to demonstrate working of print  
2 // Edit Demonstration_41.java  
3  
4 class Demonstration_41  
5 {  
6     public static void main(String[] args) {  
7         System.out.println("1. println ");  
8         System.out.println("2. println ");  
9  
10        System.out.print("1. print ");  
11        System.out.print("\n");  
12        System.out.print("2. print");  
13    }  
14}  
15  
16  
17
```

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.17134.376]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\NPTEL Java\All Final\Week 3\Code\Demonstration-IV>javac Demonstra

D:\NPTEL Java\All Final\Week 3\Code\Demonstration-IV>java Demonstra
1. println
2. println

1. print 2. print

D:\NPTEL Java\All Final\Week 3\Code\Demonstration-IV>javac Demonstra

D:\NPTEL Java\All Final\Week 3\Code\Demonstration-IV>java Demonstra
1. println
2. println

1. print
2. print

D:\NPTEL Java\All Final\Week 3\Code\Demonstration-IV>



Java source file

length : 392 lines : 17

Ln : 12 Col : 28 Sel : 0 | 0

14 items

5:31 / 38:55



YouTube





Demonstration_314.java Demonstration_42.java Demonstration_41.java Demonstration_42.java

```
4
5 class Demonstration_42
6
7 {
8     public static void main(String args[])
9     {
10         int x = 100;
11         System.out.printf("Printing simple integer: x = %d\n", x);
12
13         // this will print it upto 2 decimal places
14         System.out.printf("Formatted with precision: PI = %.2f\n", Math.PI);
15
16         float n = 5.2f;
17
18         // automatically appends zero to the rightmost part of decimal
19         System.out.printf("Formatted to specific width: n = %.4f\n", n);
20
21         n = 2324435.3f;
22
23         // here number is formatted from right margin and occupies a
24         // width of 20 characters
25         System.out.printf("Formatted to right margin: n = %20.4f\n", n);
26     }
27 }
28
29 }
```





```
4
5 class Demonstration_42
6
7 {
8     public static void main(String args[])
9     {
10         int x = 100;
11         System.out.printf("Printing simple integer: x = %d\n", x);
12
13         // this will print it upto 2 decimal places
14         System.out.printf("Formatted with precision: PI = %.2f\n", Math.PI);
15
16         float n = 5.2f;
17
18         // automatically appends zero to the rightmost part of decimal
19         System.out.printf("Formatted to specific width: n = %2.1f\n", n);
20
21         n = 2324435.3f;
22
23         // here number is formatted from right margin and occupies a
24         // width of 20 characters
25         System.out.printf("Formatted to right margin: n = %6.2f\n", n);
26     }
27 }
28
29
```





Demonstration_314.java Demonstration_47.java Demonstration_41.java Demonstration_44.java Demonstration_40.java Demonstration_45.java Demonstration_46.java

```
4 import java.io.*;
5 class Demonstration_47{
6     public static void main(String args[ ] ) { // throws Exception
7         Float principalAmount = new Float(0); //
8         Float rateOfInterest = new Float(0);
9         int numberOfYears = 0;
10        try{
11            DataInputStream in = new DataInputStream(System.in);
12
13            String tempString;
14            System.out.print("Enter Principal Amount: ");
15            System.out.flush();
16            tempString = in.readLine();
17
18            principalAmount = Float.valueOf(tempString);
19            System.out.print("Enter Rate of Interest: ");
20            System.out.flush();
21            tempString = in.readLine();
22            rateOfInterest = Float.valueOf(tempString);
23            System.out.print("Enter Number of Years: ");
24            System.out.flush();
25            tempString = in.readLine();
26            numberOfYears = Integer.parseInt(tempString);
27            // Input is over: calculate the interest
28            float interestTotal =
29            principalAmount*rateOfInterest*numberOfYears;
```





Demonstration_311.java Demonstration_47.java Demonstration_41.java Demonstration_44.java Demonstration_43.java Demonstration_45.java Demonstration_46.java

```
10 try{  
11     DataInputStream in = new DataInputStream(System.in);  
12  
13     String tempString;  
14     System.out.print("Enter Principal Amount: ");  
15     System.out.flush();  
16     tempString = in.readLine();  
17  
18     principalAmount = Float.valueOf(tempString);  
19     System.out.print("Enter Rate of Interest: ");  
20     System.out.flush();  
21     tempString = in.readLine();  
22     rateOfInterest = Float.valueOf(tempString);  
23     System.out.print("Enter Number of Years: ");  
24     System.out.flush();  
25     tempString = in.readLine();  
26     numberOfYears = Integer.parseInt(tempString);  
27     // Input is over: calculate the interest  
28     float interestTotal =  
29         principalAmount*rateOfInterest*numberOfYears;  
30     System.out.println("Total Interest = " + interestTotal);  
31 }  
32 catch (Exception ex)  
33 {}  
34 }
```



Demonstration_314.java Demonstration_47.java Demonstration_41.java Demonstration_44.java Demonstration_43.java Demonstration_45.java Demonstration_46.java

```
1  /*
2   * Input with DataInputStream InterestCalculator Program */
3   //Edit Demonstration_47.java
4   import java.io.*;
5   class Demonstration_47{
6       public static void main(String args[ ] ) throws Exception { // throws Exception
7           Float principalAmount = new Float(0);
8           Float rateOfInterest = new Float(0);
9           int numberOfYears = 0;
10          //try{
11          DataInputStream in = new DataInputStream(System.in);
12
13          String tempString;
14          System.out.print("Enter Principal Amount: ");
15          System.out.flush();
16          tempString = in.readLine();
17
18          principalAmount = Float.valueOf(tempString);
19          System.out.print("Enter Rate of Interest: ");
20          System.out.flush();
21          tempString = in.readLine();
22          rateOfInterest = Float.valueOf(tempString);
23          System.out.print("Enter Number of Years: ");
24          System.out.flush();
25          tempString = in.readLine();
26          numberOfYears = Integer.parseInt(tempString);
```