Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna

Outline

1. Basics of Method
2. Method Declaration
3. Method Naming
4. Method Calling

Basics of Method

- ❑ A method is a block of code or collection of statements or a set of code grouped together to perform a certain task.
- ❑ A method is mainly used to achieve the reusability of code.
- ❑ We write a method once and use it many times. We do not require to write code again and again.
- ❑ A method also supports easy modification and readability of code, just by adding or removing a chunk of code.
- ❑ The method is executed only when we call or invoke it.
- ❑ The most important method in Java is the main() method.
- ❑ In Java, every method must be a part of any class.

Basics of Method (Cont…)

❑ Types of Method

- There are two types of methods in Java:

1. **Predefined method:** These methods are already defined in the Java class libraries, which are also known as the standard library method or built-in method.

   We can directly use these methods just by calling them in the program.

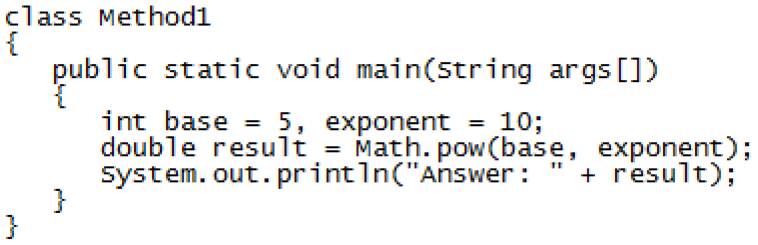
   Each and every predefined method is defined inside a class. Such as print() method is defined in the java.io.PrintStream class.

   Example: length(), equals(), compareTo(), sqrt(), etc.

2. **User-defined method:** The method written by the user or programmer is known as a user-defined method. These methods are modified according to the requirement.

Basics of Method (Cont…)

❑ Types of Method (Cont…)



```
class Method1
{
    public static void main(String args[])
    {
        int base = 5, exponent = 10;
        double result = Math.pow(base, exponent);
        System.out.println("Answer: " + result);
    }
}
```

Basics of Method (Cont…)

❑ Types of Method (Cont…)

❖ Output

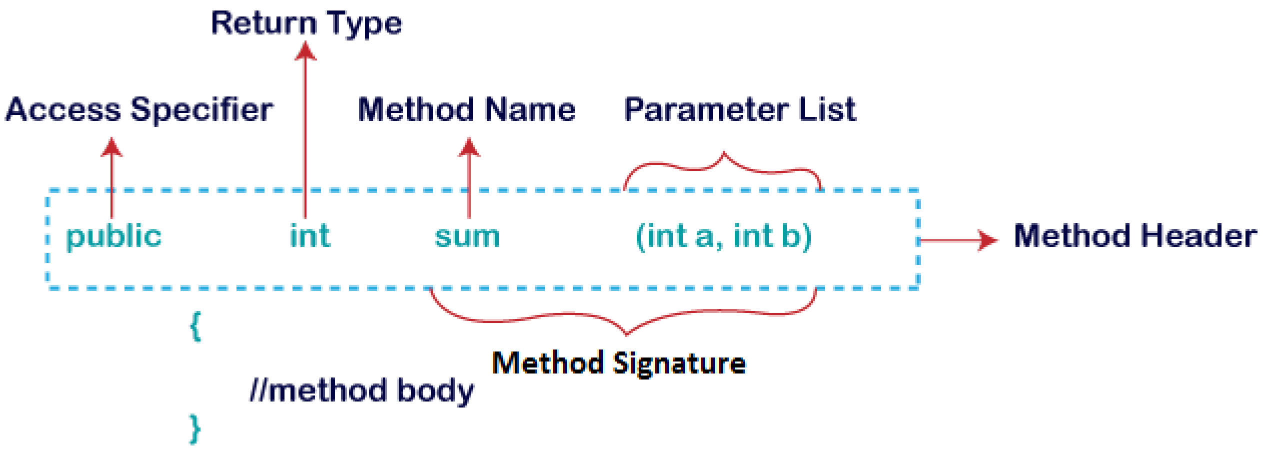
```
9765625.0
```

Method Declaration

- The method declaration provides information about method attributes.
- It has six components that are known as **method header**.

1. **Method signature:** It includes the method name and parameter list.
2. **Access specifier:** It specifies the visibility of the method. There are four types of access specifier: public, private, protected and default.
3. **Return type:** A method can return a primitive data type, object, collection, etc. If it does not return anything, we use **void** keyword.
4. **Method name**
5. **Parameter list:** It contains the data types and variable names separated by comma.
6. **Method body:** A set of statements.

Method Declaration (Cont…)

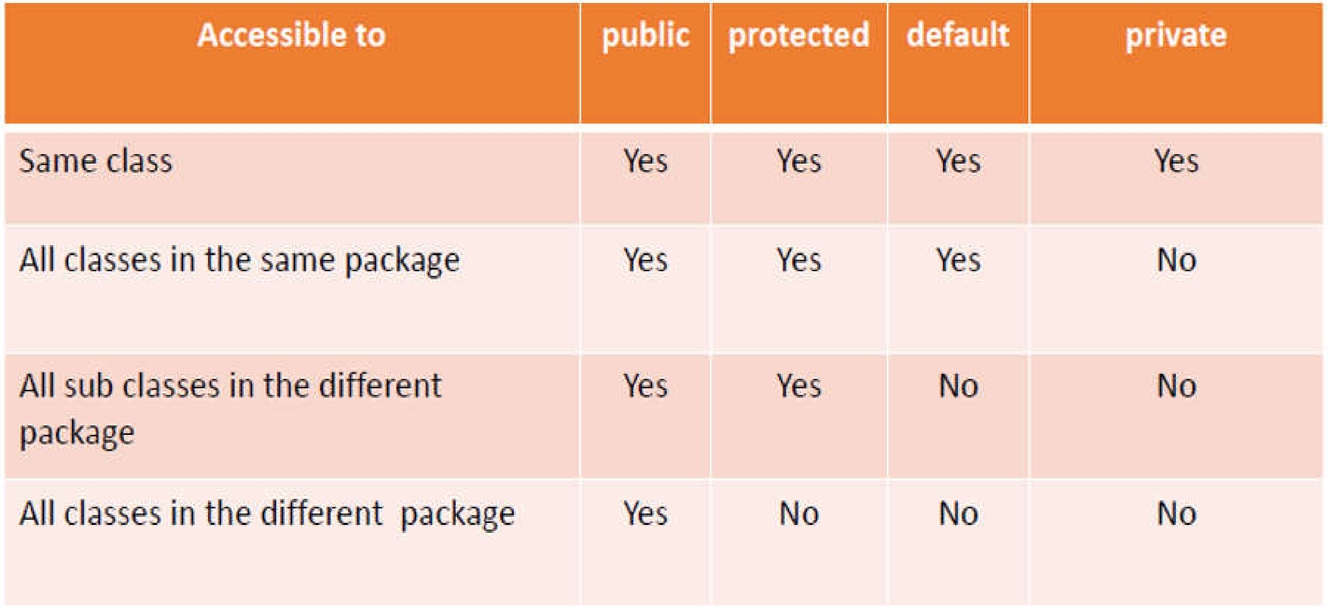


```
public          int        sum          (int a, int b)
          {
                 //method body
          }
```

Fig. 1: Method declaration

Method Declaration (Cont…)

- Access Specifier



Accessible to	public	protected	default	private
Same class	Yes	Yes	Yes	Yes
All classes in the same package	Yes	Yes	Yes	No
All sub classes in the different package	Yes	Yes	No	No
All classes in the different package	Yes	No	No	No

Method Naming

- The method name must be a verb and start with a lowercase letter.
- If the method name has more than two words, the first name must be a verb followed by adjective or noun.
- In the multi-word method name, the first letter of each word must be in uppercase except the first word. For example:

  Single-word method name: show(), sum()

  Multi-word method name: areaOfSquare(), stringComparision()
- It is also possible that a method has the same name as another method name in the same class, it is known as **method overloading**.

Method Calling

- Once a method is defined, it must be called.

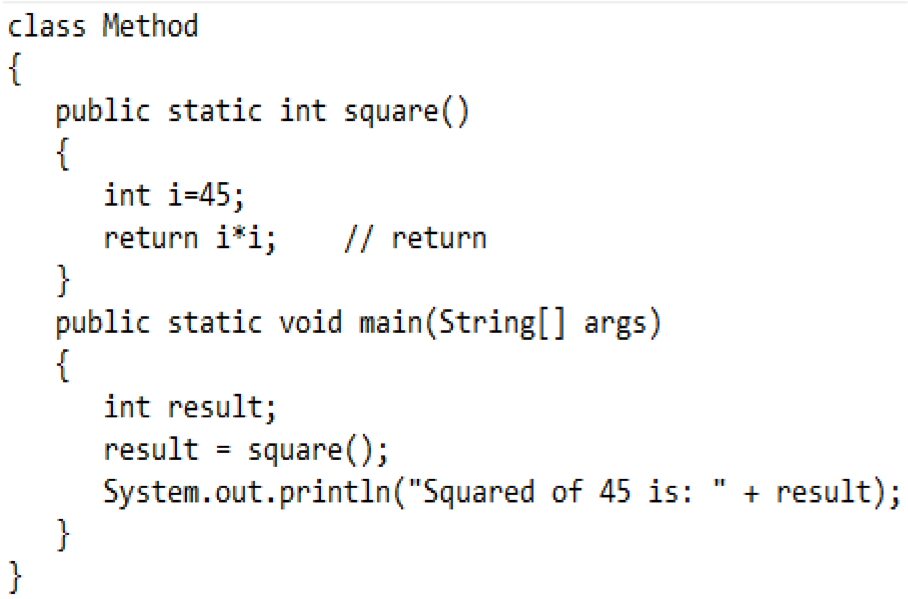
- The calling of a method in a program is simple.

- When we call or invoke a method, the program control transfer to the called method.

- We can call a method for any number of time in a program.

Method Calling (Cont…)

- ❑ **Return from Method without Argument**



```
class Method
{
   public static int square()
   {
      int i=45;
      return i*i;    // return
   }
   public static void main(String[] args)
   {
      int result;
      result = square();
      System.out.println("Squared of 45 is: " + result);
   }
}
```

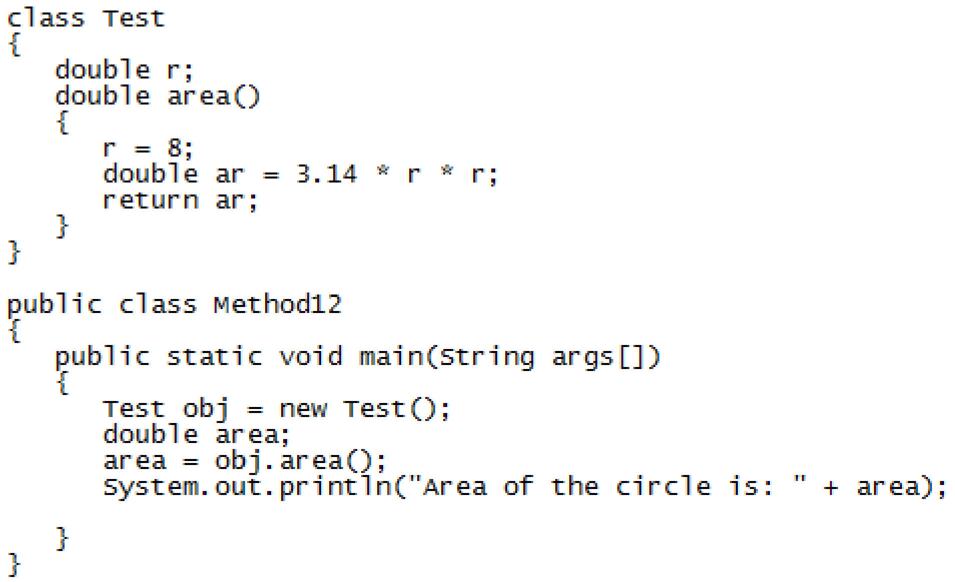
Method Calling (Cont…)

- **Return from Method without Argument (Cont…)**
  - Output

```
Squared of 45 is: 2025
```

Method Calling (Cont…)

❑ Return from Method without Argument (Cont…)



```
class Test
{
    double r;
    double area()
    {
        r = 8;
        double ar = 3.14 * r * r;
        return ar;
    }
}

public class Method12
{
    public static void main(String args[])
    {
        Test obj = new Test();
        double area;
        area = obj.area();
        System.out.println("Area of the circle is: " + area);

    }
}
```

Method Calling (Cont…)

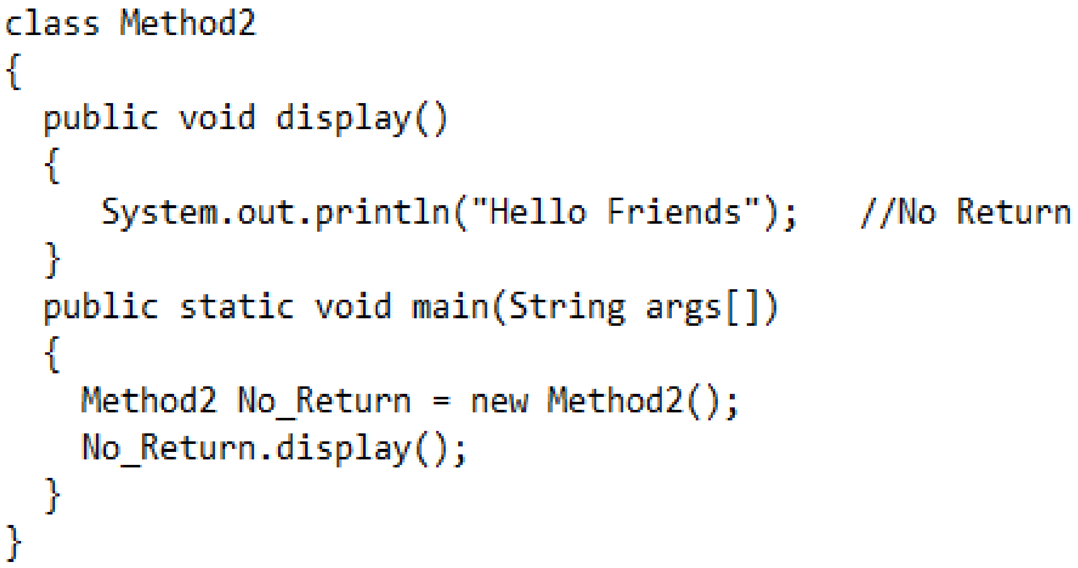
❑ Return from Method without Argument (Cont…)

- Output

```
Area of the circle is: 200.96
```

Method Calling (Cont…)

❑ No Return from Method without Argument



```
class Method2
{
  public void display()
  {
     System.out.println("Hello Friends");   //No Return
  }
  public static void main(String args[])
  {
    Method2 No_Return = new Method2();
    No_Return.display();
  }
}
```

Method Calling (Cont…)

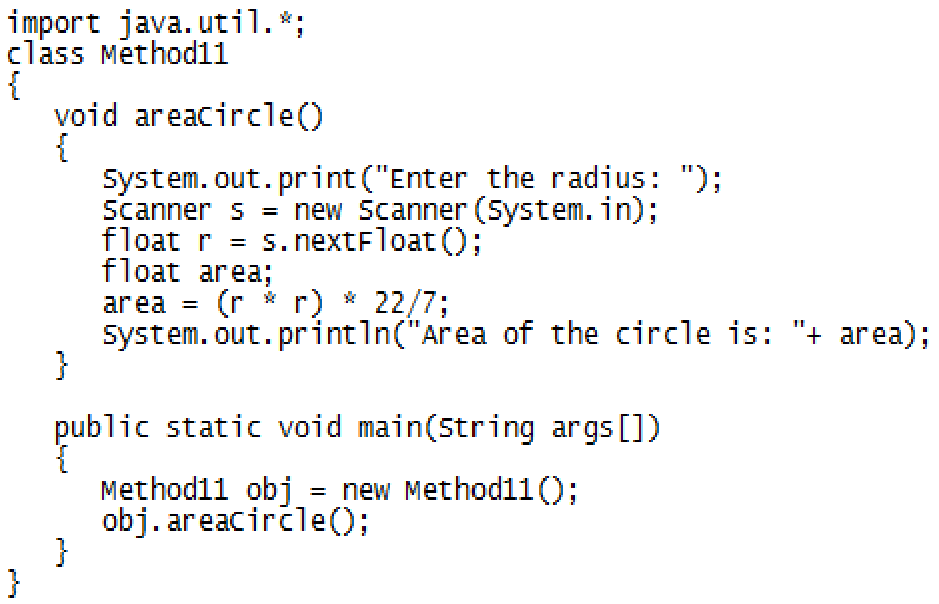
❑ No Return from Method without Argument (Cont…)

- Output

```
Hello Friends
```

Method Calling (Cont…)

❑ No Return from Method without Argument (Cont…)



```
import java.util.*;
class Method11
{
    void areaCircle()
    {
        System.out.print("Enter the radius: ");
        Scanner s = new Scanner(System.in);
        float r = s.nextFloat();
        float area;
        area = (r * r) * 22/7;
        System.out.println("Area of the circle is: "+ area);
    }

    public static void main(String args[])
    {
        Method11 obj = new Method11();
        obj.areaCircle();
    }
}
```

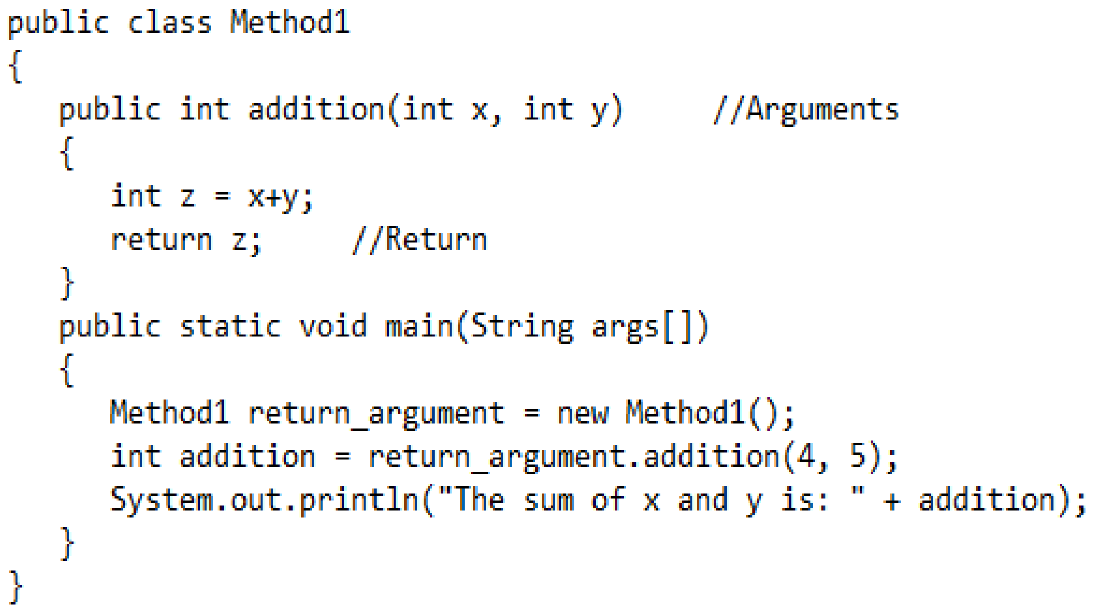
Method Calling (Cont…)

- **No Return from Method without Argument (Cont…)**
  - Output

```
Enter the radius: 5
Area of the circle is: 78.57143
```

Method Calling (Cont…)

❑ Return with Arguments



```
public class Method1
{
    public int addition(int x, int y)     //Arguments
    {
        int z = x+y;
        return z;     //Return
    }
    public static void main(String args[])
    {
        Method1 return_argument = new Method1();
        int addition = return_argument.addition(4, 5);
        System.out.println("The sum of x and y is: " + addition);
    }
}
```

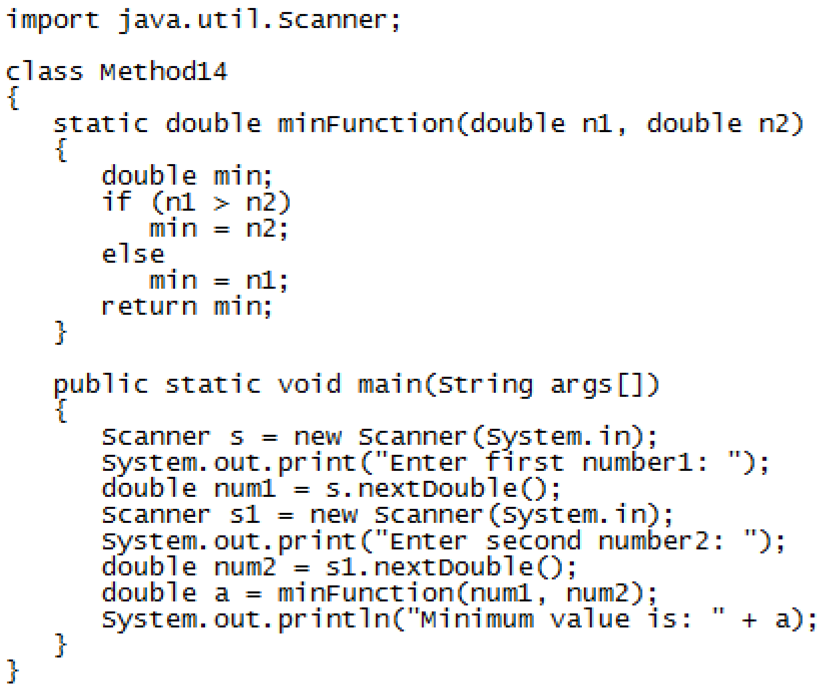
Method Calling (Cont…)

- **Return with Arguments (Cont…)**
  - Output

```
The sum of x and y is: 9
```

Method Calling (Cont…)

❑ Return with Arguments (Cont…)

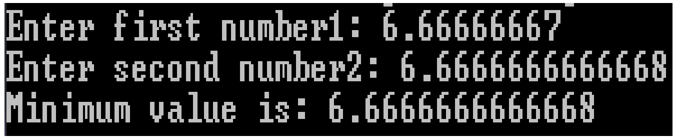


```
import java.util.Scanner;

class Method14
{
   static double minFunction(double n1, double n2)
   {
      double min;
      if (n1 > n2)
         min = n2;
      else
         min = n1;
      return min;
   }

   public static void main(String args[])
   {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter first number1: ");
      double num1 = s.nextDouble();
      Scanner s1 = new Scanner(System.in);
      System.out.print("Enter second number2: ");
      double num2 = s1.nextDouble();
      double a = minFunction(num1, num2);
      System.out.println("Minimum value is: " + a);
   }
}
```

Method Calling (Cont…)

❑ Return with Arguments (Cont…)

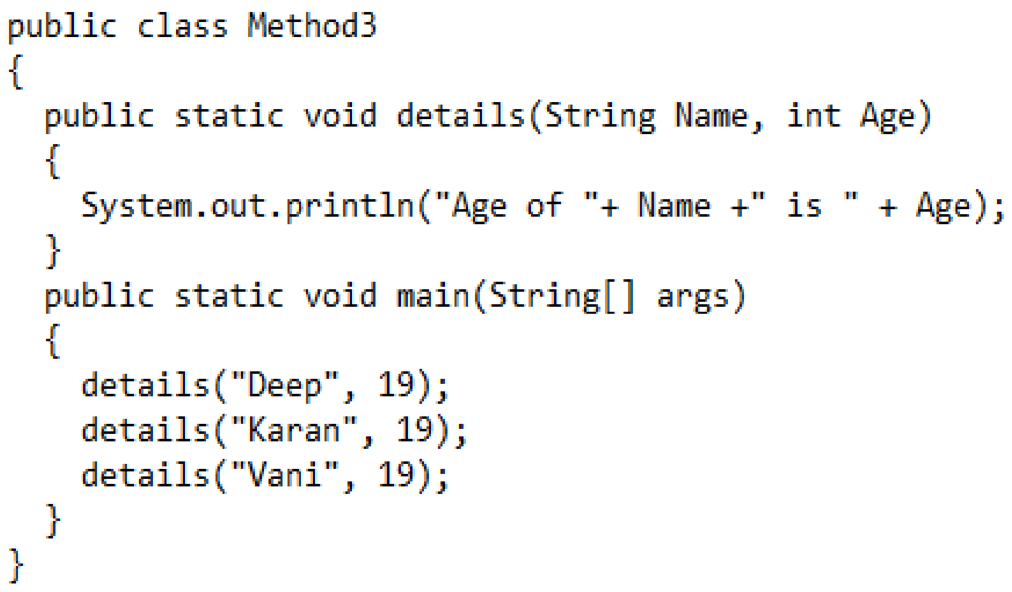
- ❖Output



```
Enter first number1: 6.66666667
Enter second number2: 6.6666666666668
Minimum value is: 6.6666666666668
```

Method Calling (Cont…)

- **No Return with Arguments**



```
public class Method3
{
  public static void details(String Name, int Age)
  {
    System.out.println("Age of "+ Name +" is " + Age);
  }
  public static void main(String[] args)
  {
    details("Deep", 19);
    details("Karan", 19);
    details("Vani", 19);
  }
}
```

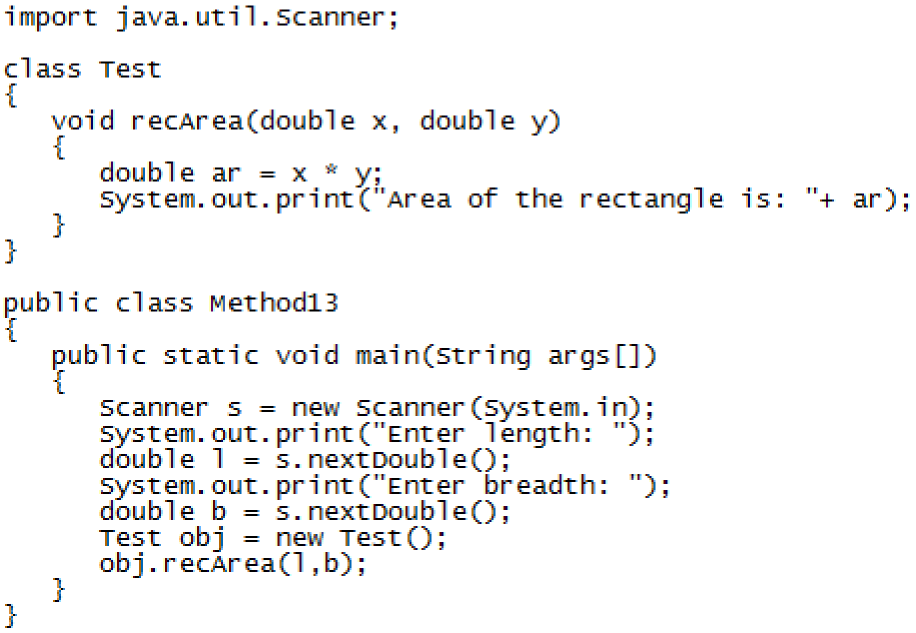
Method Calling (Cont…)

- **No Return with Arguments (Cont…)**
  - Output

```
Age of Deep is 19
Age of Karan is 19
Age of Vani is 19
```

Method Calling (Cont…)

❑ No Return with Arguments (Cont…)



```java
import java.util.Scanner;

class Test
{
   void recArea(double x, double y)
   {
      double ar = x * y;
      System.out.print("Area of the rectangle is: "+ ar);
   }
}

public class Method13
{
   public static void main(String args[])
   {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter length: ");
      double l = s.nextDouble();
      System.out.print("Enter breadth: ");
      double b = s.nextDouble();
      Test obj = new Test();
      obj.recArea(l,b);
   }
}
```

Method Calling (Cont…)

❑ No Return with Arguments (Cont…)

- ❖Output

```
Enter length: 4                    //User enters 4
Enter breadth: 5.6                 //User enters 5.6
Area of the rectangle is: 22.6
```

thank
you . . .

Slides are prepared from various sources, such as Book, Internet Links and many more.