INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian
Associate Professor
Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee 247 667
balarfcs@iitr.ac.in
https://sites.google.com/site/balaiitr/

Creating Objects

```
int [] arr;
arr = new int[10];
```

```
int [] arr = new int[10];
```

```
Student1 s1;
s1=new Student1();      // A
```

- Creating an object is also referred to as instantiating an object.
- When we declare **s1**, it points to null.
- When we instantiate as given in second line **s1** is a reference to **Student1**.
- The method **Studnet1()** is the default constructor of the class.
- Same as

```
Student1 s1=new Student1();   // B
```

Creating Objects

- We can create any number of objects of Student1. For example
  - **`Student1 s1=new Student1();`**
  - **`Student1 s2=new Student1();`**

- Object References

Accessing Class members

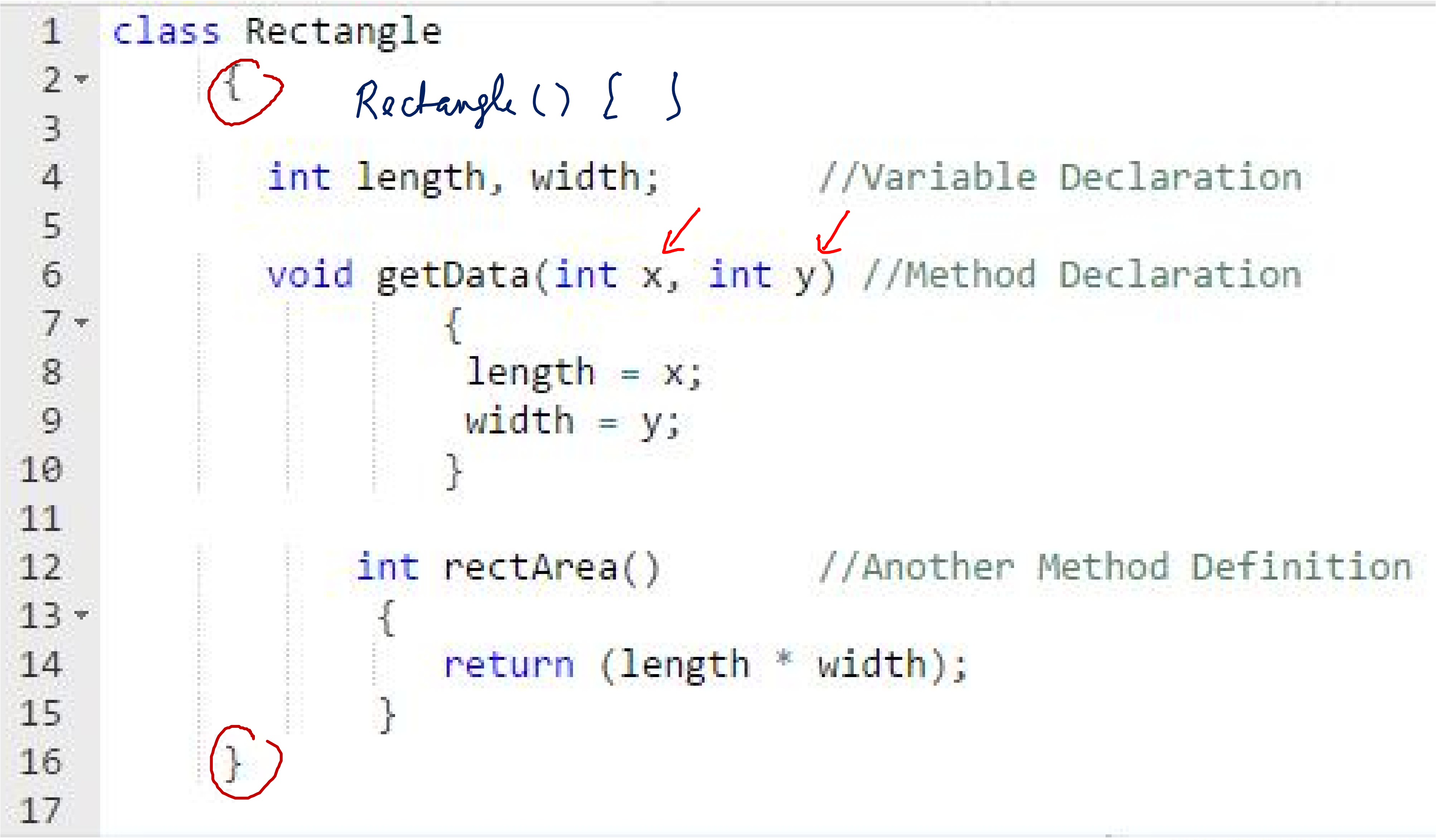
- Object and dot operator

S1. id = 1007;

S1. getdata ( ) ;

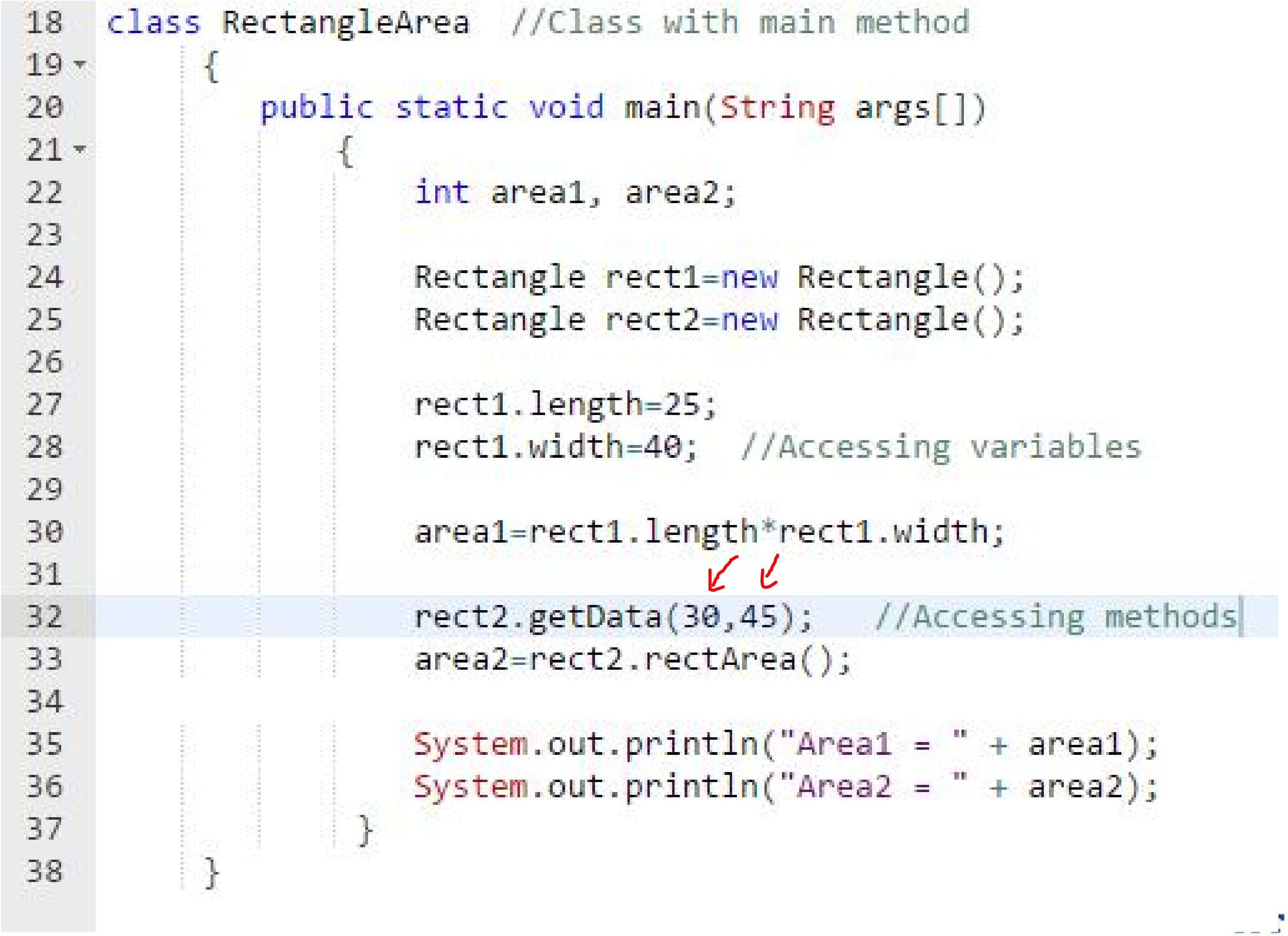
```
Objectname.variablename=value;
Objectname.methodname(parameterList);
```

Example

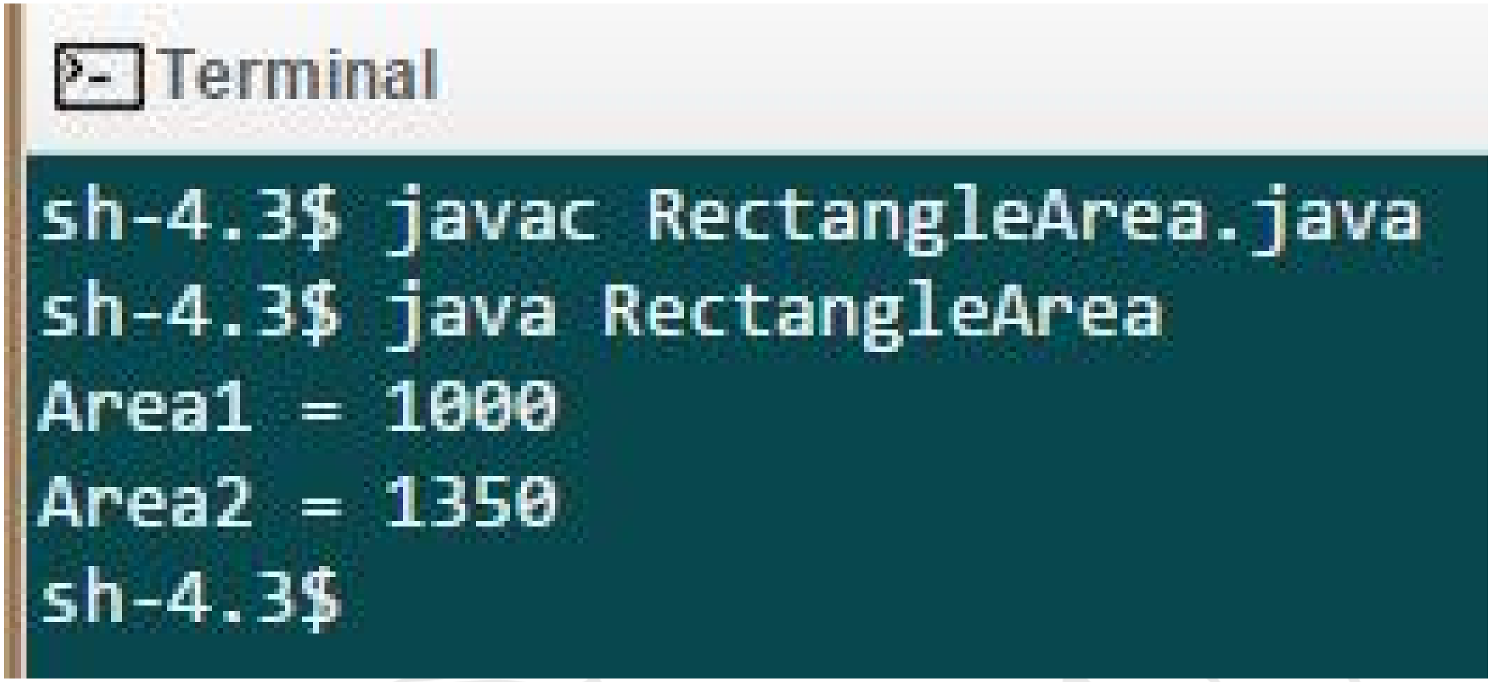


```
class Rectangle
     {

       int length, width;       //Variable Declaration

       void getData(int x, int y) //Method Declaration
               {
                length = x;
                width = y;
               }

           int rectArea()       //Another Method Definition
            {
               return (length * width);
            }
     }

```

Rectangle ( ) { }



```java
class RectangleArea  //Class with main method
     {
        public static void main(String args[])
            {
                int area1, area2;

                Rectangle rect1=new Rectangle();
                Rectangle rect2=new Rectangle();

                rect1.length=25;
                rect1.width=40;  //Accessing variables

                area1=rect1.length*rect1.width;

                rect2.getData(30,45);   //Accessing methods
                area2=rect2.rectArea();

                System.out.println("Area1 = " + area1);
                System.out.println("Area2 = " + area2);
             }
     }
```



```
sh-4.3$ javac RectangleArea.java
sh-4.3$ java RectangleArea
Area1 = 1000
Area2 = 1350
sh-4.3$
```

Constructors

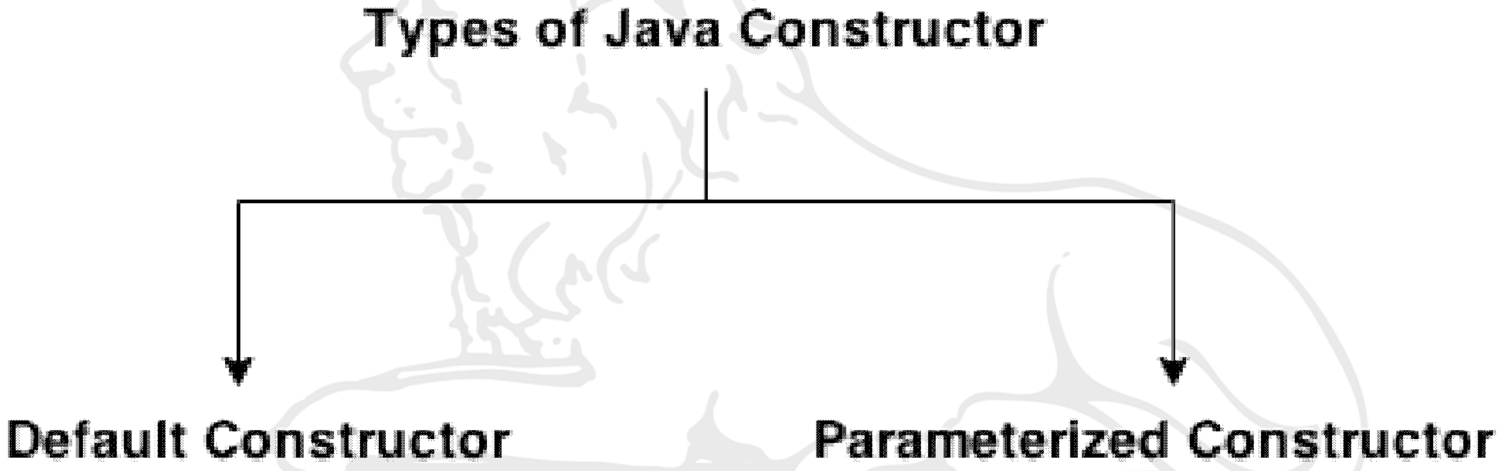
- **Constructor in java** is a *special type of method* that is used to initialize the object.
- Java constructor is *invoked at the time of object creation*. It constructs the values i.e. provides data for the object that is why it is known as constructor.

Rules for creating java constructor

- There are basically two rules defined for the constructor.
  - Constructor name must be same as its class name
  - Constructor must have no explicit return type

Types of java constructors

- There are two types of constructors:
  - – Default constructor (no-arg constructor)
  - – Parameterized constructor



Types of Java Constructor

Default Constructor

Parameterized Constructor

Java Default Constructor

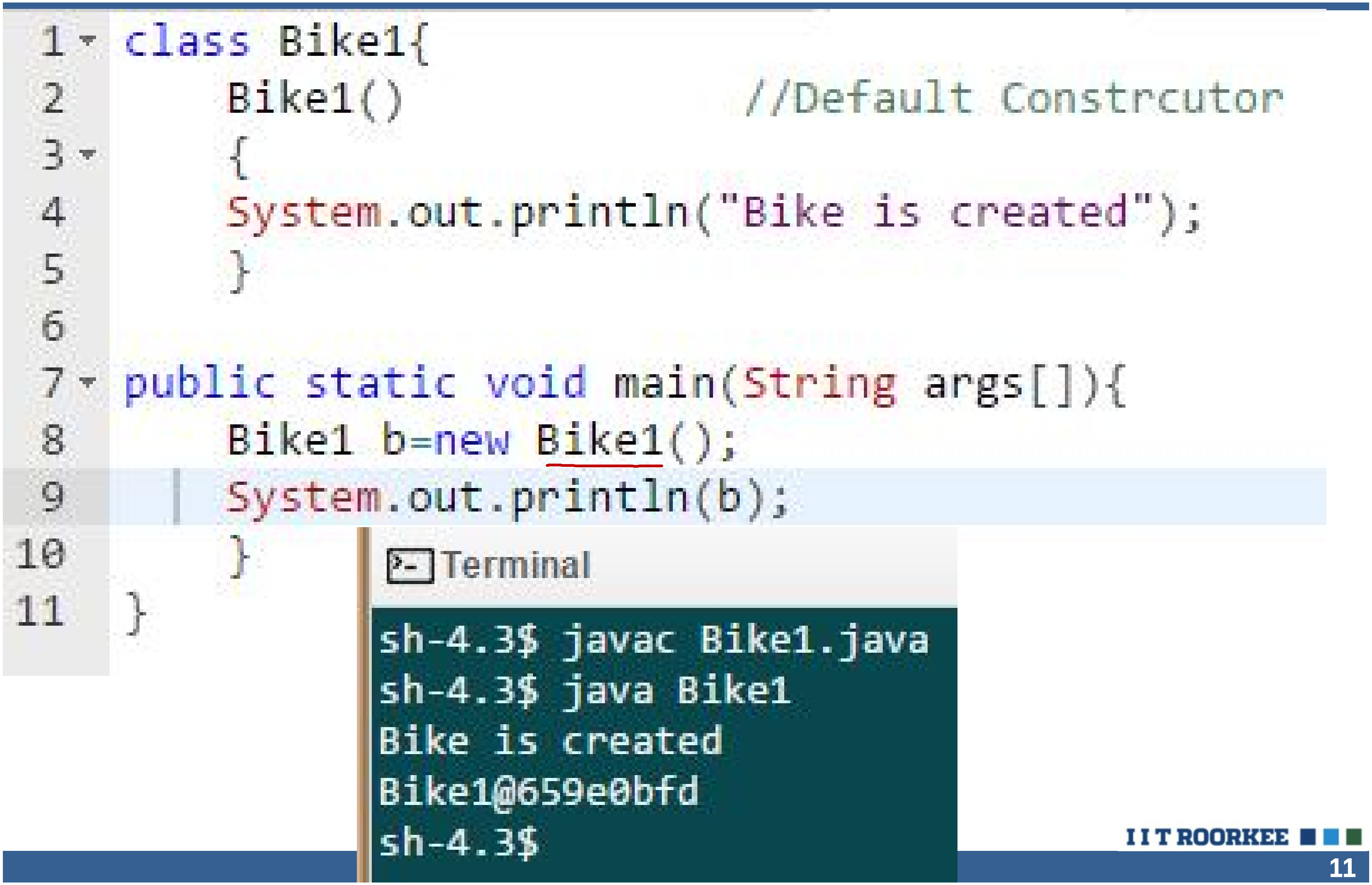
- Java Default Constructor

- A constructor that have no parameter is known as default constructor.

- **Syntax of default constructor:**

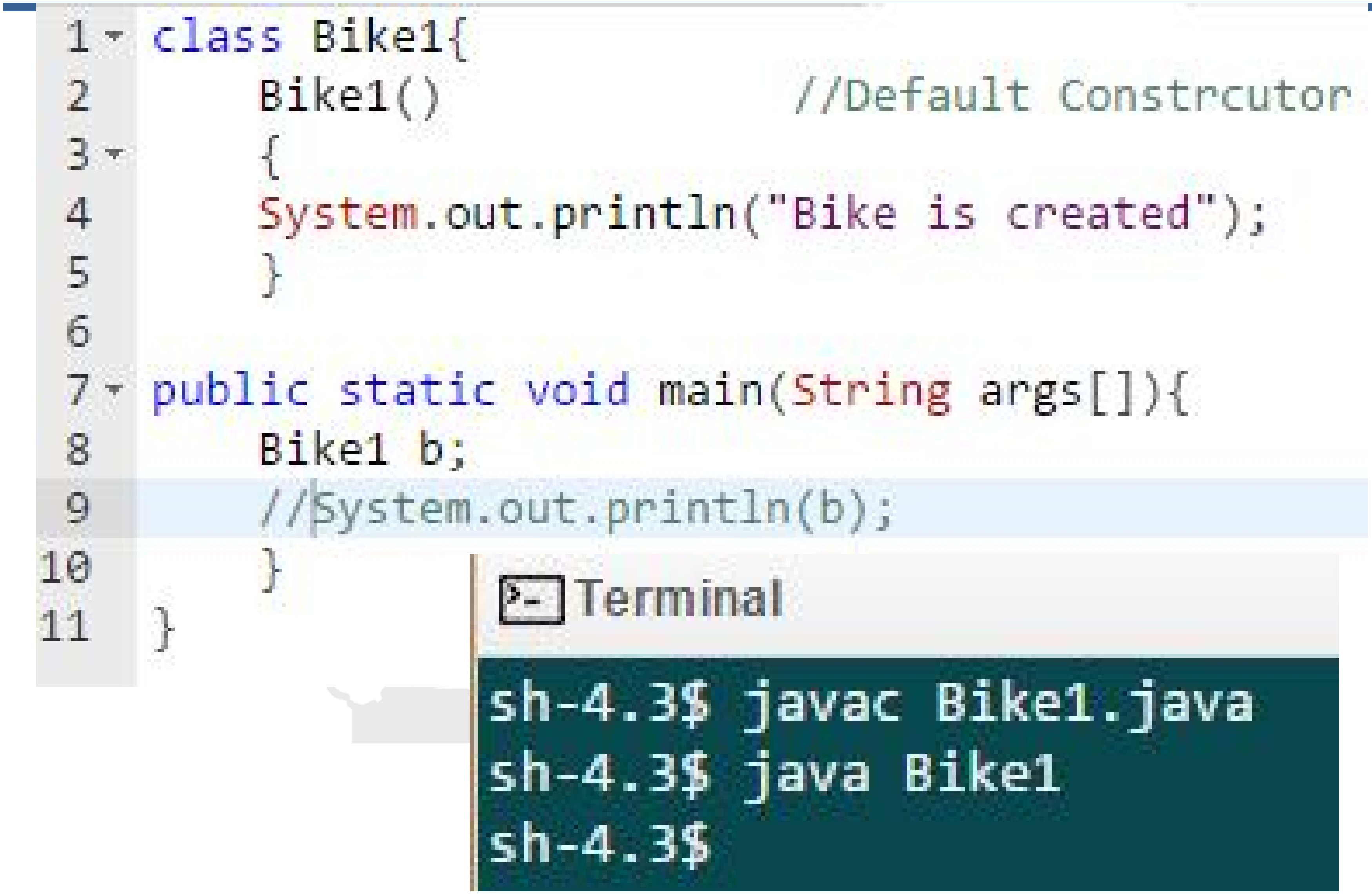
```
<class_name>(){}
```

Example



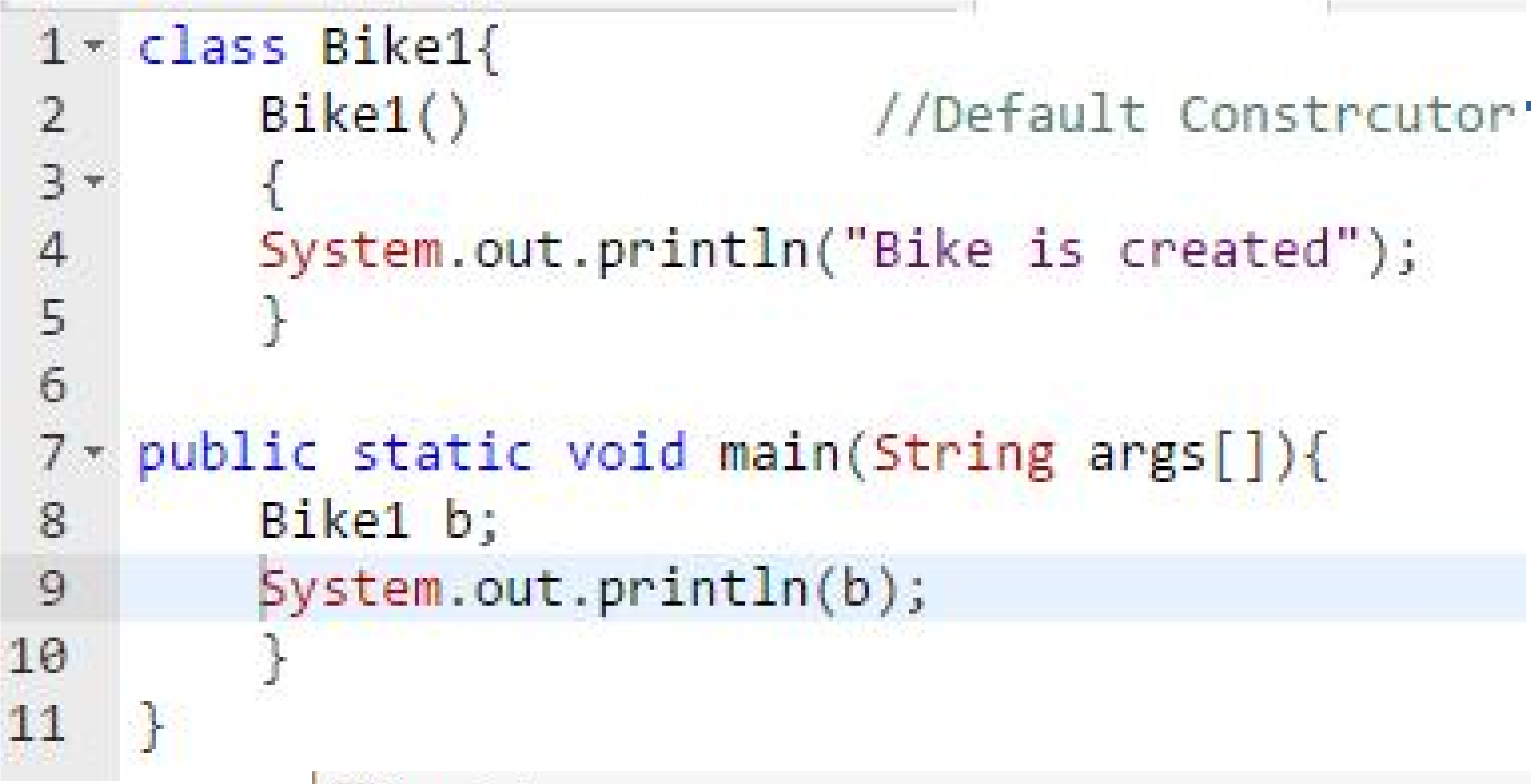
```java
class Bike1{
    Bike1()                    //Default Constrcutor
    {
    System.out.println("Bike is created");
    }

public static void main(String args[]){
    Bike1 b=new Bike1();
    System.out.println(b);
    }
}
```

```
sh-4.3$ javac Bike1.java
sh-4.3$ java Bike1
Bike is created
Bike1@659e0bfd
sh-4.3$
```

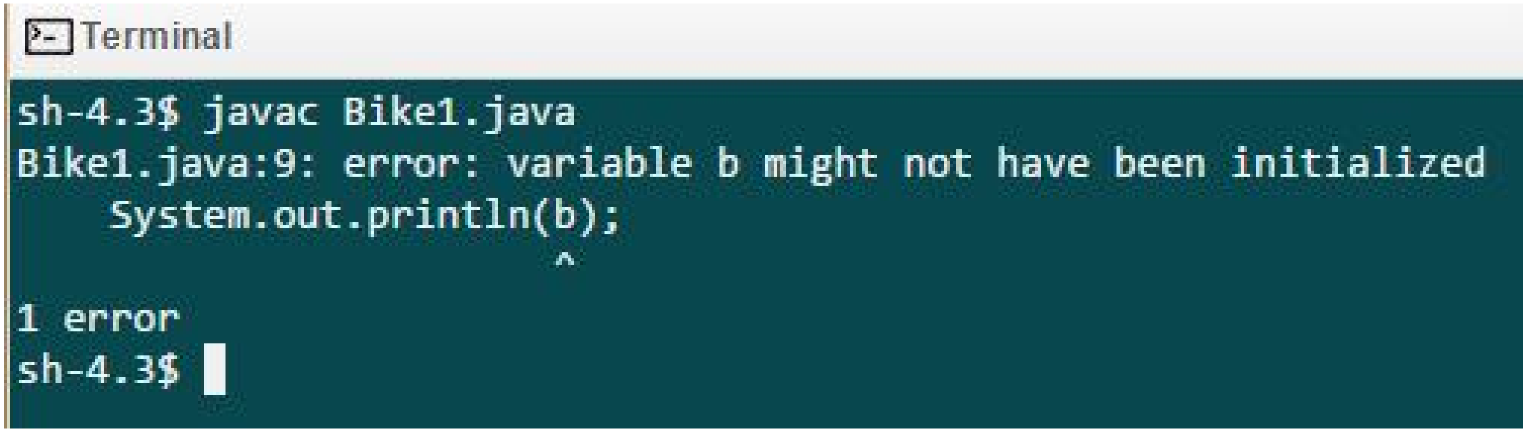


```java
class Bike1{
	Bike1()				//Default Constrcutor
	{
	System.out.println("Bike is created");
	}

public static void main(String args[]){
	Bike1 b;
	//System.out.println(b);
	}
}
```

```
sh-4.3$ javac Bike1.java
sh-4.3$ java Bike1
sh-4.3$
```

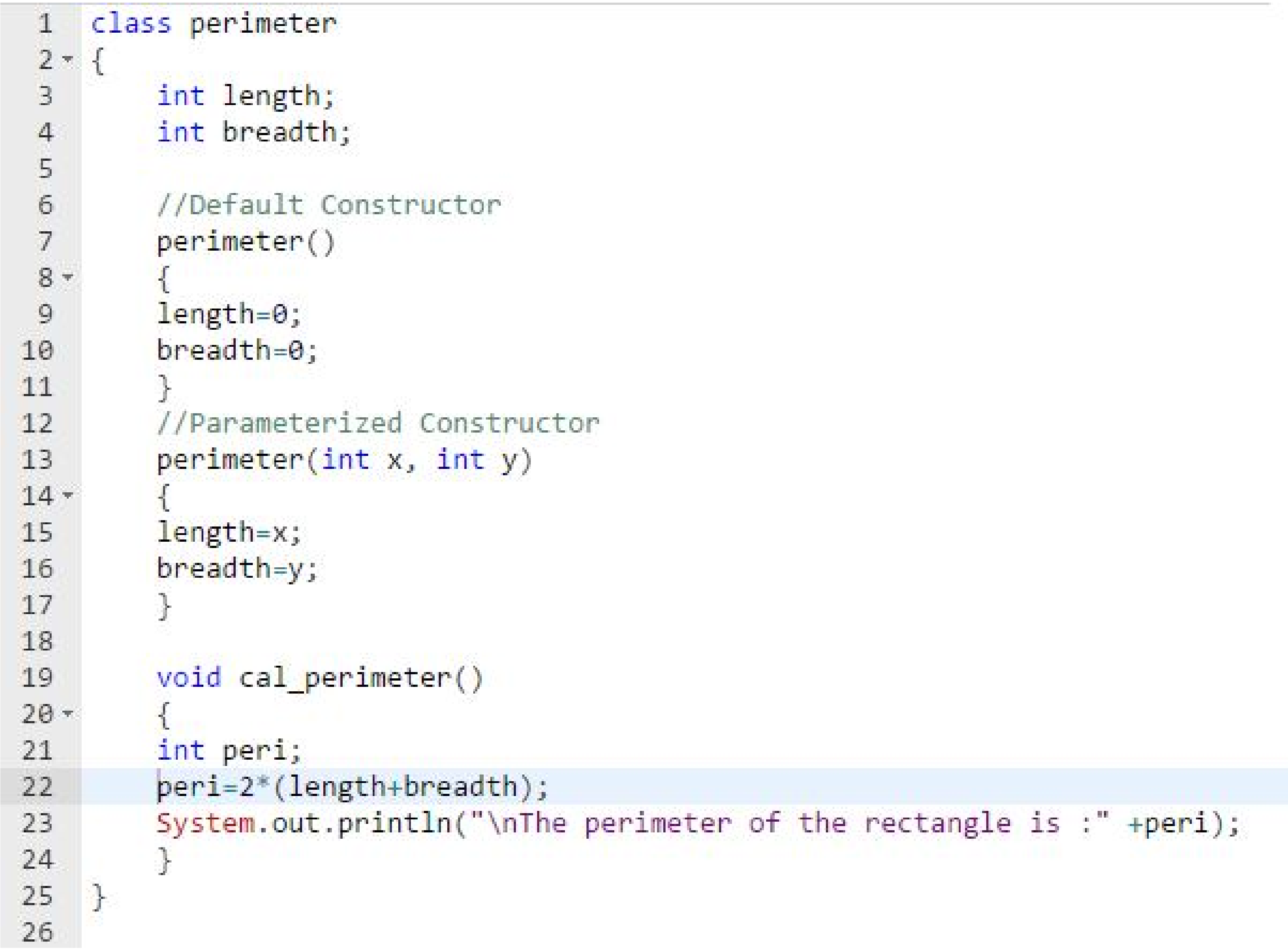


```java
class Bike1{
    Bike1()              //Default Constrcutor
    {
    System.out.println("Bike is created");
    }

public static void main(String args[]){
    Bike1 b;
    System.out.println(b);
    }
}
```

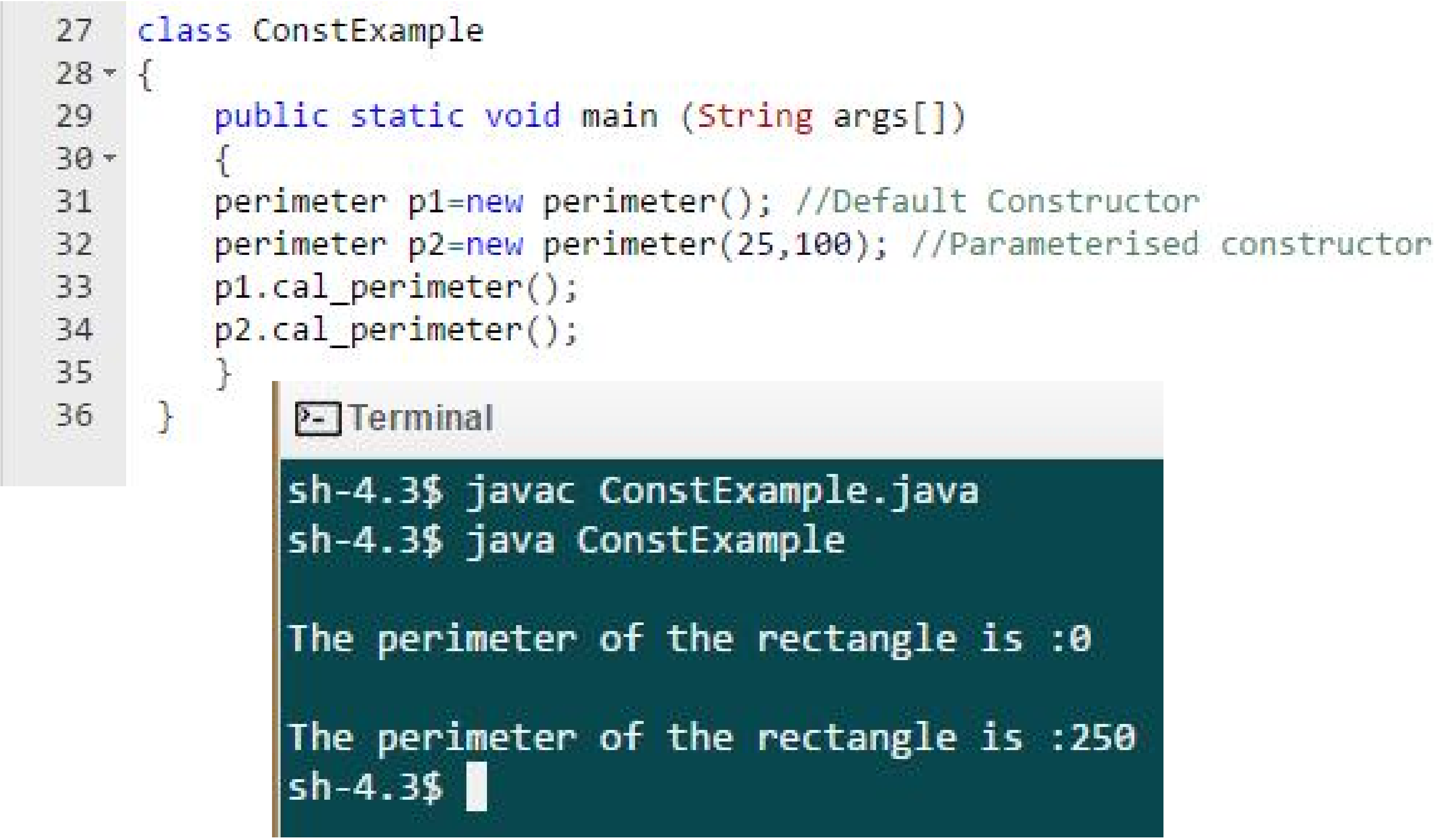


```
sh-4.3$ javac Bike1.java
Bike1.java:9: error: variable b might not have been initialized
    System.out.println(b);
                       ^
1 error
sh-4.3$ 
```

Default and Parameterized Constructor



```
class perimeter
{
    int length;
    int breadth;

    //Default Constructor
    perimeter()
    {
    length=0;
    breadth=0;
    }
    //Parameterized Constructor
    perimeter(int x, int y)
    {
    length=x;
    breadth=y;
    }

    void cal_perimeter()
    {
    int peri;
    peri=2*(length+breadth);
    System.out.println("\nThe perimeter of the rectangle is :" +peri);
    }
}

```



```java
class ConstExample
{
    public static void main (String args[])
    {
    perimeter p1=new perimeter(); //Default Constructor
    perimeter p2=new perimeter(25,100); //Parameterised constructor
    p1.cal_perimeter();
    p2.cal_perimeter();
    }
}
```

```
sh-4.3$ javac ConstExample.java
sh-4.3$ java ConstExample

The perimeter of the rectangle is :0

The perimeter of the rectangle is :250
sh-4.3$
```