(Q1.1) Create a Book class with a parameterized constructor:

- **Define a Book class with attributes title, author, and price.**
- **Implement a parameterized constructor to initialize these attributes.**
- **Write a display method to print the book details.**

Code

```
class Book {
   String title;
   String author;
   float price;
   // parameterized constructor
   Book(String t,String a,float p)
   {
      title=t;
      author=a;
      price=p;

   }
   void display()
   {
      System.out.println("The name of the book is " +title);
      System.out.println("The author is "+author);
      System.out.println("The price is "+price);
   }

}
public class Mainclass{
    public static void main(String[] args)
        {
           // creating an object of the class
            Book book1=new Book("A girl in room 105","Chetan Bhagat",160);
           //displaying the function
            book1.display();
         }
 }
```

Output

```
PS D:\JAVA from beginning> cd "d:\JAVA from beginning\" ; if ($?) { javac Mainclass.java } ; if ($?) { java Mainclass }
The name of the book is A girl in room 105
The author is Chetan Bhagat
The price is 160.0
```

(Q1.2) Create an overloaded constructor for a Student class:

- **Define a Student class with attributes name, rollNumber, and grade.**
- **Implement a parameterized constructor to initialize all attributes.**
- **Implement another constructor that only initializes name and rollNumber with a default grade of 'A'.**
- **Write a display method to print the student details.**

Code

```
class Student{
   String name;
   int age;
   public void printInfo()
   {
      System.out.println("The name of the student is"+this.name);
      System.out.println("The age of the student is"+this.age);
}
}

public class Oops2 {
 public static void main(String args[])
 {
 Student s1=new Student();
 s1.name="Nandini";
 s1.age=20;
 Student s2=new Student();
 s2.name="Rima";
 s2.age=21;
 s1.printInfo();
 s2.printInfo();
}
}
```

Output

```
PS D:\JAVA from beginning> cd "d:\JAVA from beginning\" ; if ($?) { javac Oops2.java } ; if ($?) { java Oops2 }
The name of the student isNandini
The age of the student is20
The name of the student isRima
The age of the student is21
```

(Q1.3) Create a Rectangle class with default and parameterized constructors:

- ● Define a Rectangle class with attributes length and width.

● Implement a default constructor that initializes length and width to 1.

● Implement a parameterized constructor to initialize length and width with given values.

● Write methods to calculate and return the area and perimeter of the rectangle.

● Write a display method to print the rectangle details.

Code

```
import java.util.*;
public class Rectangle {
   double height;
   double width;

   Rectangle() {
      height = 1;
      width = 1;
   }
```

```
Rectangle(double h, double w) {
this.height = h;
this.width = w;
}

double calculatePerimeter() {
return 2 * (height + width);
}

double calculateArea() {
return (height * width);
}
void display() {

System.out.println("The perimeter of the rectangle is: " + calculatePerimeter());

System.out.println("The area of the reactangle is : " + calculateArea());
}

public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the height of the rectangle");
double height = sc.nextDouble();
System.out.println("Enter the width of the rectangle ");
double width = sc.nextDouble();
Rectangle recObj = new Rectangle(height, width);
recObj.display();
sc.close();
}
```

Output

```
PSjava -cp /tmp/Tgss6xoxuN/Rectangle
Enter the height of the rectangle
34
Enter the width of the rectangle
23
The perimeter of the rectangle is: 52.0
The area of the reactangle is : 69.0
```

(Q1.4) Create a Person class with a constructor chaining:

- **Define a Person class with attributes name, age, and address.**
- **Implement a parameterized constructor to initialize all attributes**
- **Implement another constructor that only initializes name and age, and sets address to "Unknown".**
- **Use constructor chaining to avoid code duplication.**
- **Write a display method to print the person details.**

Code

```
public class Person {
    String name;
    int age;
    String address;

    // Constructor to initialize all attributes
    public Person(String name, int age, String address) {
        this.name = name;
        this.age = age;
        this.address = address;
    }

    // Constructor to initialize name and age, set address to "Unknown"
    public Person(String name, int age) {
        this(name, age, "Unknown"); // Constructor chaining
    }

    // Method to display person details
    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
    }

public static void main(String[] args) {
 // Creating Person object with all attributes
 Person person1 = new Person("Alice", 30, "123 Main St");
 person1.display();

 System.out.println();

 // Creating Person object with name and age, address set to "Unknown"
 Person person2 = new Person("Bob", 25);
 person2.display();
 }
}
```

Output

```
PS D:\JAVA from beginning> cd "d:\JAVA from beginning\" ; if ($?) { javac Person.java } ; if ($?) { java Person }
Name: Alice
Age: 30
Address: 123 Main St

Name: Bob
Age: 25
Address: Unknown
```