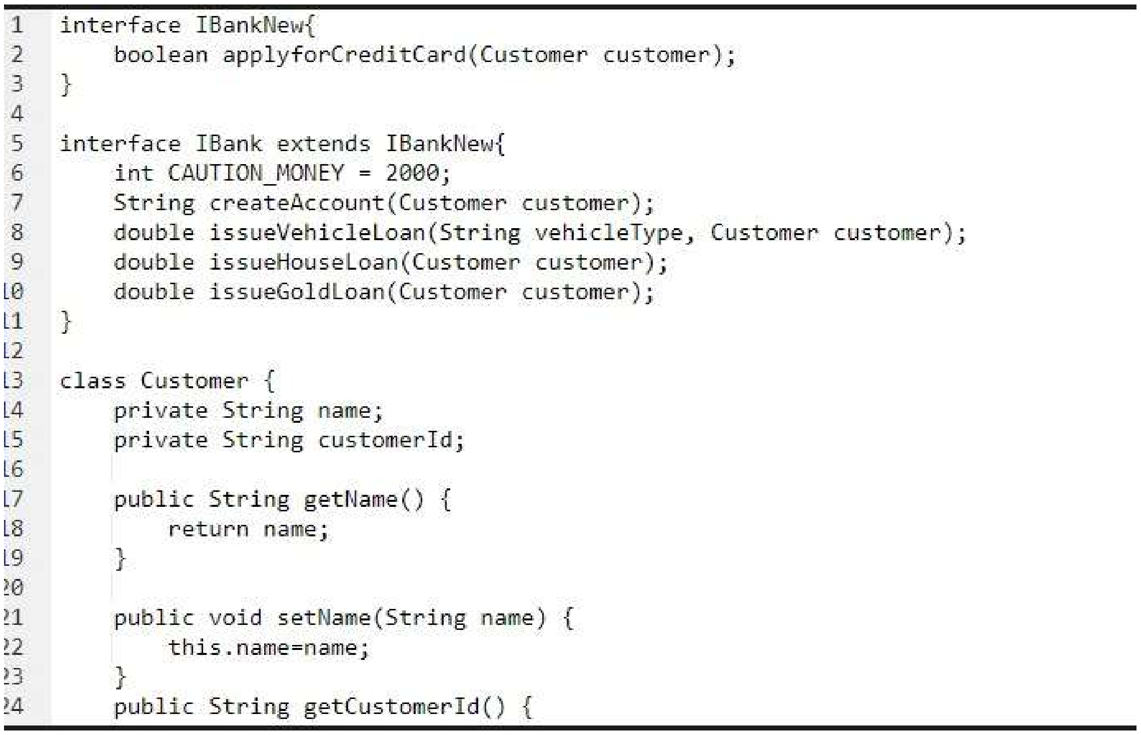
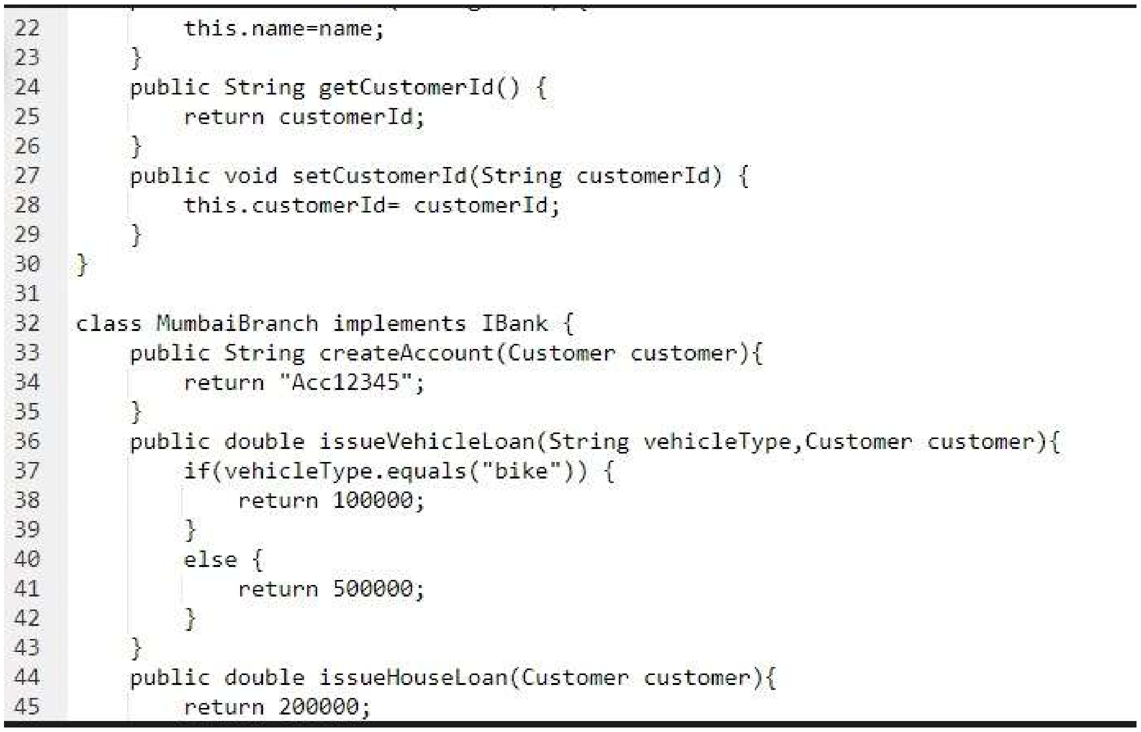
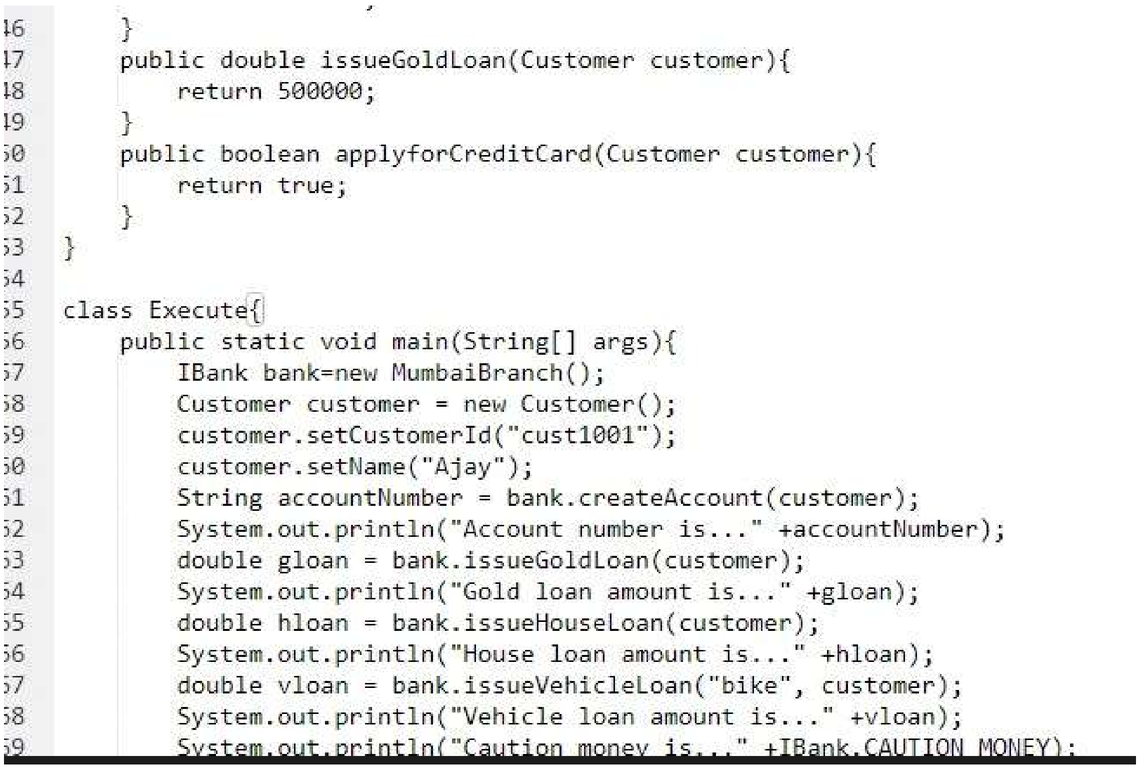


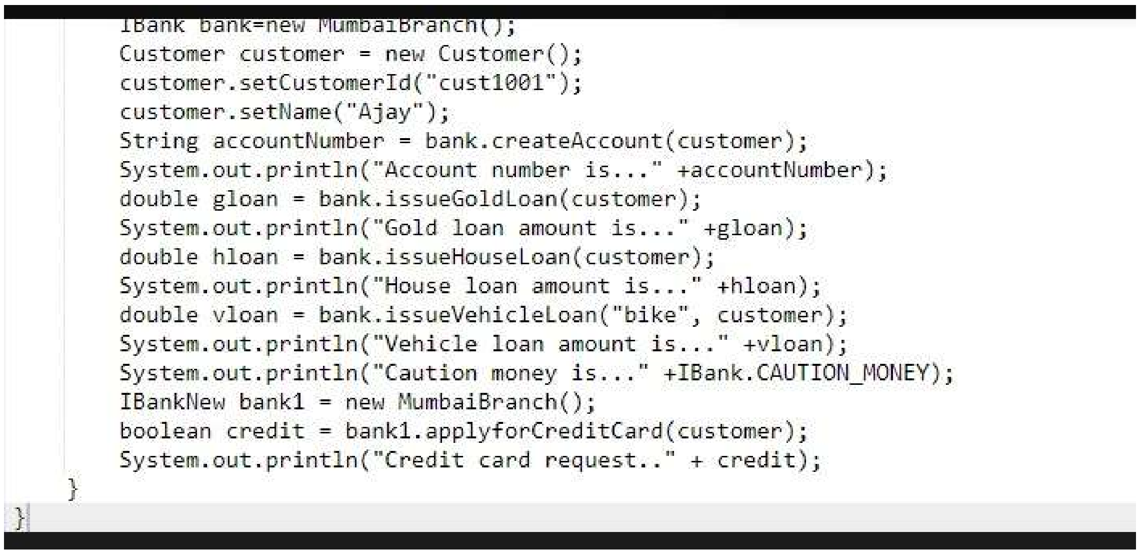
```java
interface IBankNew{
    boolean applyforCreditCard(Customer customer);
}

interface IBank extends IBankNew{
    int CAUTION_MONEY = 2000;
    String createAccount(Customer customer);
    double issueVehicleLoan(String vehicleType, Customer customer);
    double issueHouseLoan(Customer customer);
    double issueGoldLoan(Customer customer);
}

class Customer {
    private String name;
    private String customerId;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name=name;
    }
    public String getCustomerId() {
```



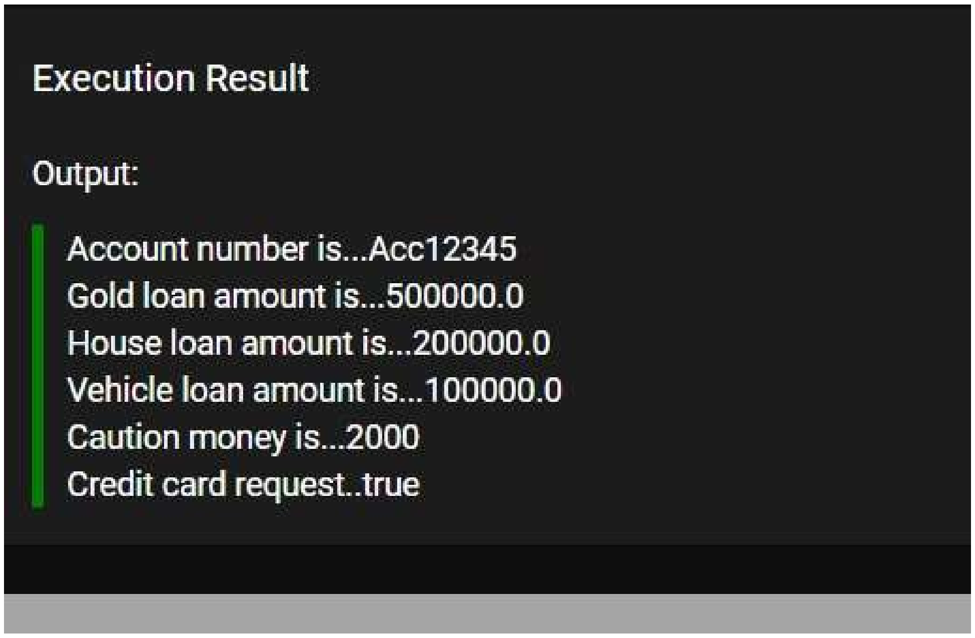
```java
        this.name=name;
    }
    public String getCustomerId() {
        return customerId;
    }
    public void setCustomerId(String customerId) {
        this.customerId= customerId;
    }
}

class MumbaiBranch implements IBank {
    public String createAccount(Customer customer){
        return "Acc12345";
    }
    public double issueVehicleLoan(String vehicleType,Customer customer){
        if(vehicleType.equals("bike")) {
            return 100000;
        }
        else {
            return 500000;
        }
    }
    public double issueHouseLoan(Customer customer){
        return 200000;
```



```
    }
    public double issueGoldLoan(Customer customer){
        return 500000;
    }
    public boolean applyforCreditCard(Customer customer){
        return true;
    }
}

class Execute{
    public static void main(String[] args){
        IBank bank=new MumbaiBranch();
        Customer customer = new Customer();
        customer.setCustomerId("cust1001");
        customer.setName("Ajay");
        String accountNumber = bank.createAccount(customer);
        System.out.println("Account number is..." +accountNumber);
        double gloan = bank.issueGoldLoan(customer);
        System.out.println("Gold loan amount is..." +gloan);
        double hloan = bank.issueHouseLoan(customer);
        System.out.println("House loan amount is..." +hloan);
        double vloan = bank.issueVehicleLoan("bike", customer);
        System.out.println("Vehicle loan amount is..." +vloan);
        System.out.println("Caution money is..." +IBank.CAUTION_MONEY);
```



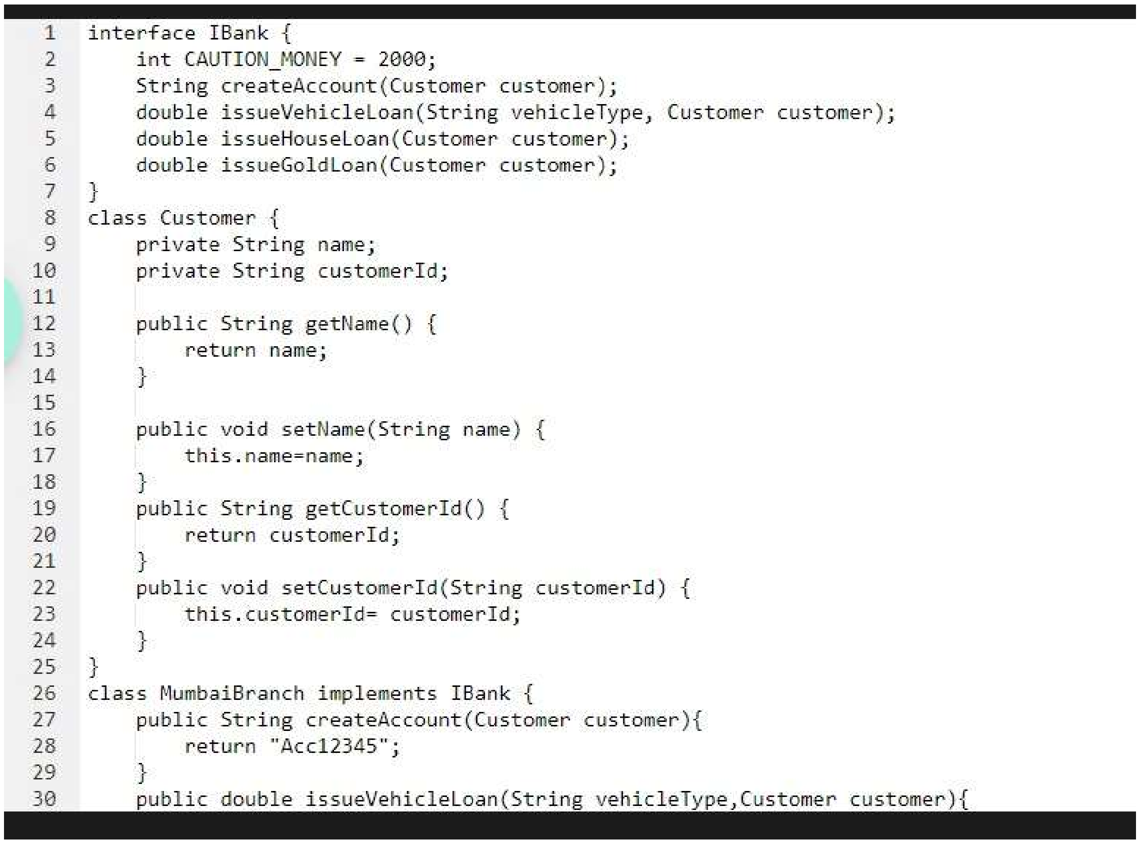
```
        IBank bank=new MumbaiBranch();
        Customer customer = new Customer();
        customer.setCustomerId("cust1001");
        customer.setName("Ajay");
        String accountNumber = bank.createAccount(customer);
        System.out.println("Account number is..." +accountNumber);
        double gloan = bank.issueGoldLoan(customer);
        System.out.println("Gold loan amount is..." +gloan);
        double hloan = bank.issueHouseLoan(customer);
        System.out.println("House loan amount is..." +hloan);
        double vloan = bank.issueVehicleLoan("bike", customer);
        System.out.println("Vehicle loan amount is..." +vloan);
        System.out.println("Caution money is..." +IBank.CAUTION_MONEY);
        IBankNew bank1 = new MumbaiBranch();
        boolean credit = bank1.applyforCreditCard(customer);
        System.out.println("Credit card request.." + credit);
    }
}
```



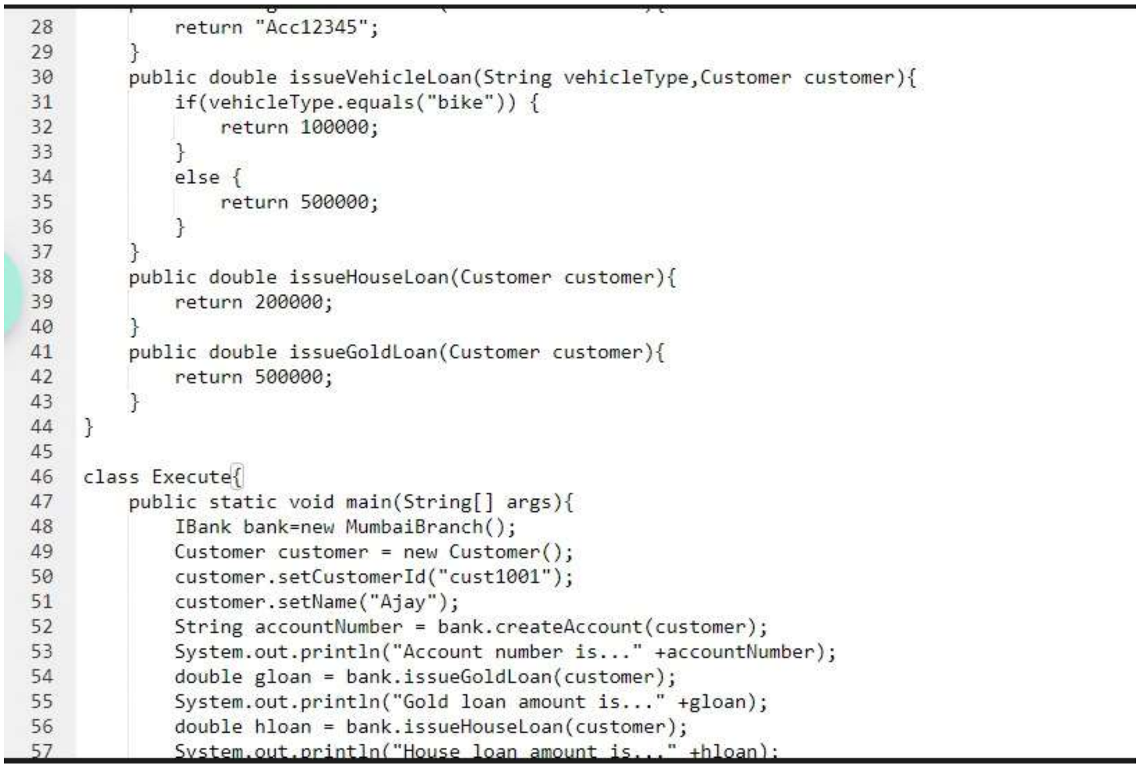
Execution Result

Output:

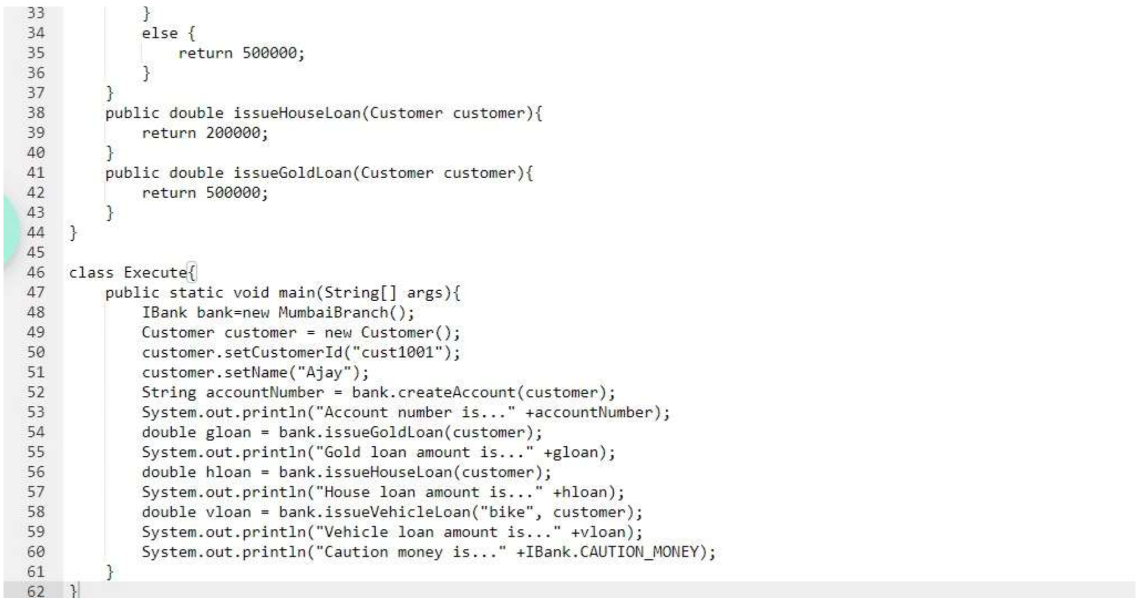
```
Account number is...Acc12345
Gold loan amount is...500000.0
House loan amount is...200000.0
Vehicle loan amount is...100000.0
Caution money is...2000
Credit card request..true
```



```java
interface IBank {
    int CAUTION_MONEY = 2000;
    String createAccount(Customer customer);
    double issueVehicleLoan(String vehicleType, Customer customer);
    double issueHouseLoan(Customer customer);
    double issueGoldLoan(Customer customer);
}
class Customer {
    private String name;
    private String customerId;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name=name;
    }
    public String getCustomerId() {
        return customerId;
    }
    public void setCustomerId(String customerId) {
        this.customerId= customerId;
    }
}
class MumbaiBranch implements IBank {
    public String createAccount(Customer customer){
        return "Acc12345";
    }
    public double issueVehicleLoan(String vehicleType,Customer customer){
```

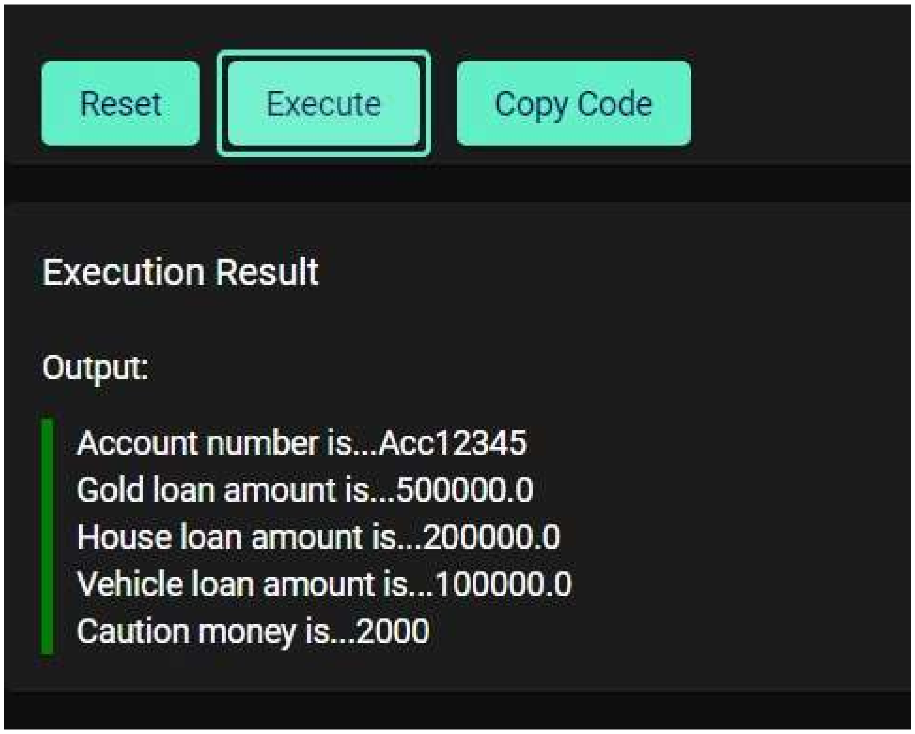


```
        return "Acc12345";
    }
    public double issueVehicleLoan(String vehicleType,Customer customer){
        if(vehicleType.equals("bike")) {
            return 100000;
        }
        else {
            return 500000;
        }
    }
    public double issueHouseLoan(Customer customer){
        return 200000;
    }
    public double issueGoldLoan(Customer customer){
        return 500000;
    }
}

class Execute{
    public static void main(String[] args){
        IBank bank=new MumbaiBranch();
        Customer customer = new Customer();
        customer.setCustomerId("cust1001");
        customer.setName("Ajay");
        String accountNumber = bank.createAccount(customer);
        System.out.println("Account number is..." +accountNumber);
        double gloan = bank.issueGoldLoan(customer);
        System.out.println("Gold loan amount is..." +gloan);
        double hloan = bank.issueHouseLoan(customer);
        System.out.println("House loan amount is..." +hloan);
```



```
        }
        else {
            return 500000;
        }
    }
    public double issueHouseLoan(Customer customer){
        return 200000;
    }
    public double issueGoldLoan(Customer customer){
        return 500000;
    }
}

class Execute{
    public static void main(String[] args){
        IBank bank=new MumbaiBranch();
        Customer customer = new Customer();
        customer.setCustomerId("cust1001");
        customer.setName("Ajay");
        String accountNumber = bank.createAccount(customer);
        System.out.println("Account number is..." +accountNumber);
        double gloan = bank.issueGoldLoan(customer);
        System.out.println("Gold loan amount is..." +gloan);
        double hloan = bank.issueHouseLoan(customer);
        System.out.println("House loan amount is..." +hloan);
```

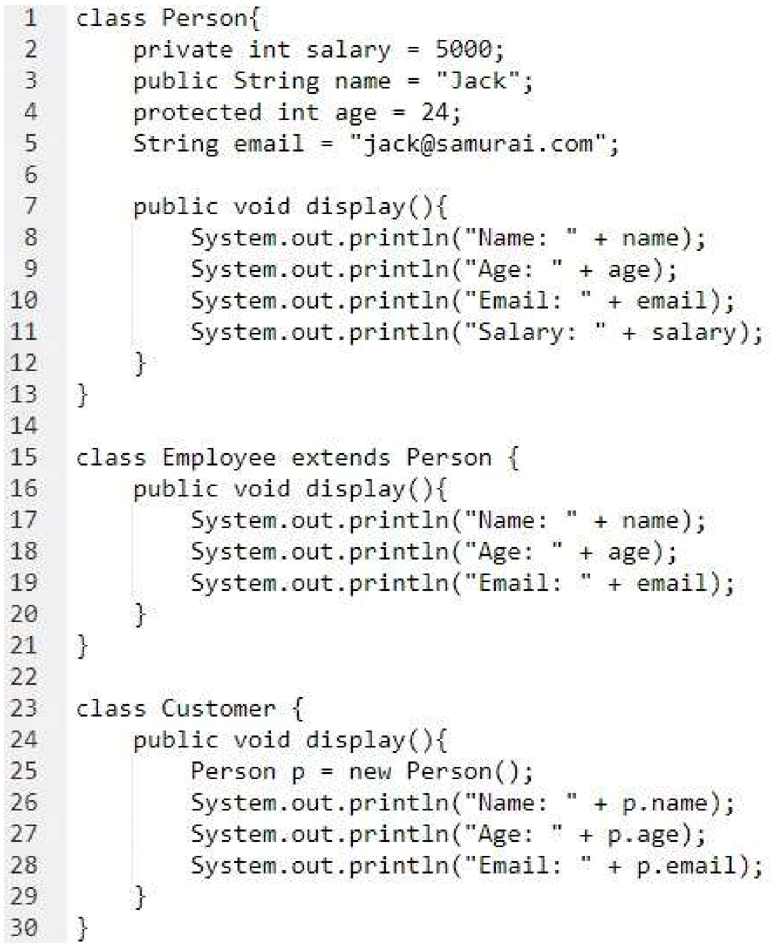
```
        double vloan = bank.issueVehicleLoan("bike", customer);
        System.out.println("Vehicle loan amount is..." +vloan);
        System.out.println("Caution money is..." +IBank.CAUTION_MONEY);
    }
}
```



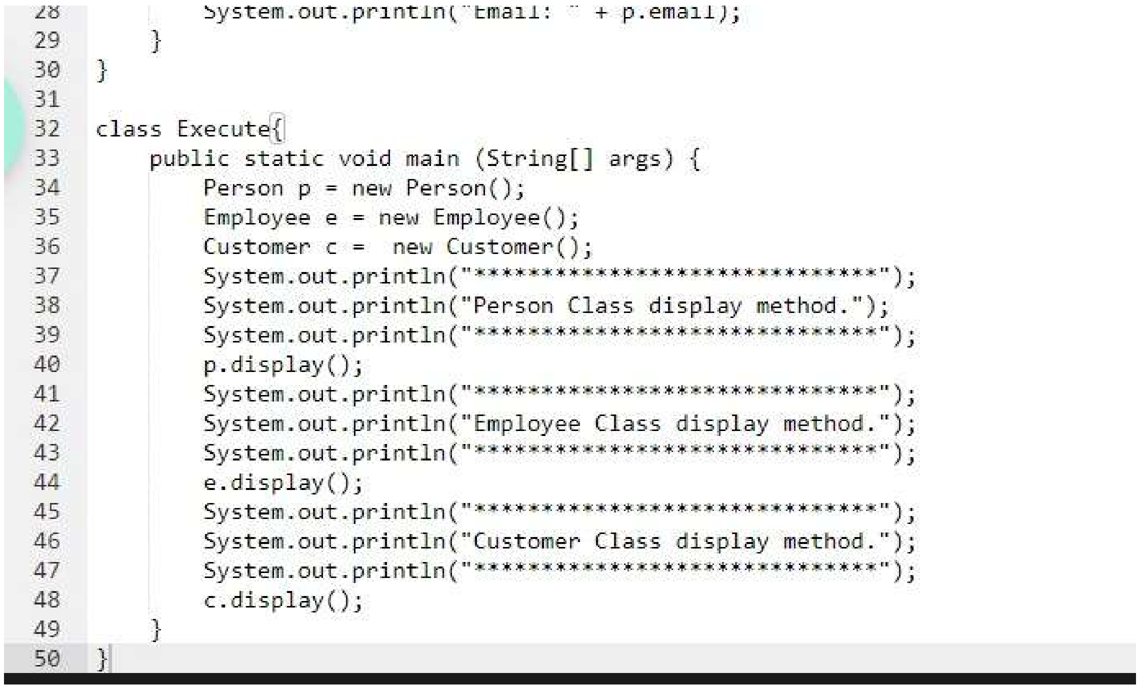
Execution Result

Output:

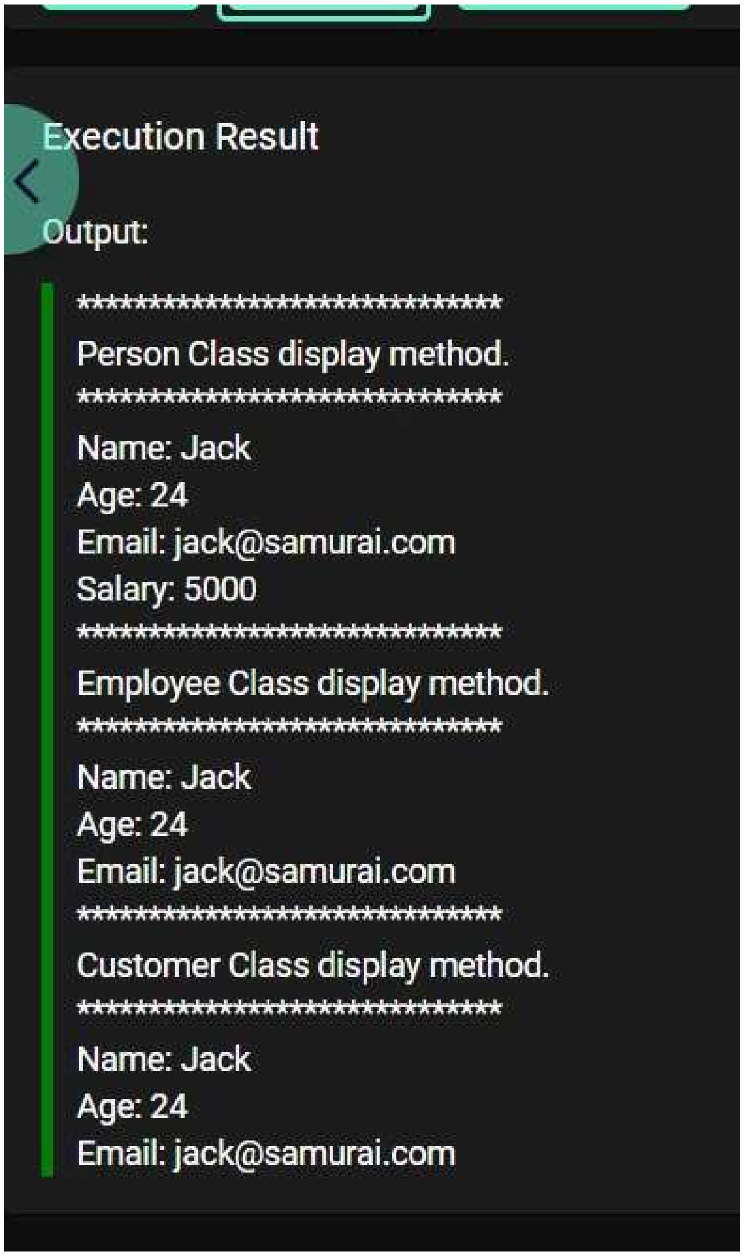
```
Account number is...Acc12345
Gold loan amount is...500000.0
House loan amount is...200000.0
Vehicle loan amount is...100000.0
Caution money is...2000
```



```java
class Person{
    private int salary = 5000;
    public String name = "Jack";
    protected int age = 24;
    String email = "jack@samurai.com";

    public void display(){
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Email: " + email);
        System.out.println("Salary: " + salary);
    }
}

class Employee extends Person {
    public void display(){
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Email: " + email);
    }
}

class Customer {
    public void display(){
        Person p = new Person();
        System.out.println("Name: " + p.name);
        System.out.println("Age: " + p.age);
        System.out.println("Email: " + p.email);
    }
}
```

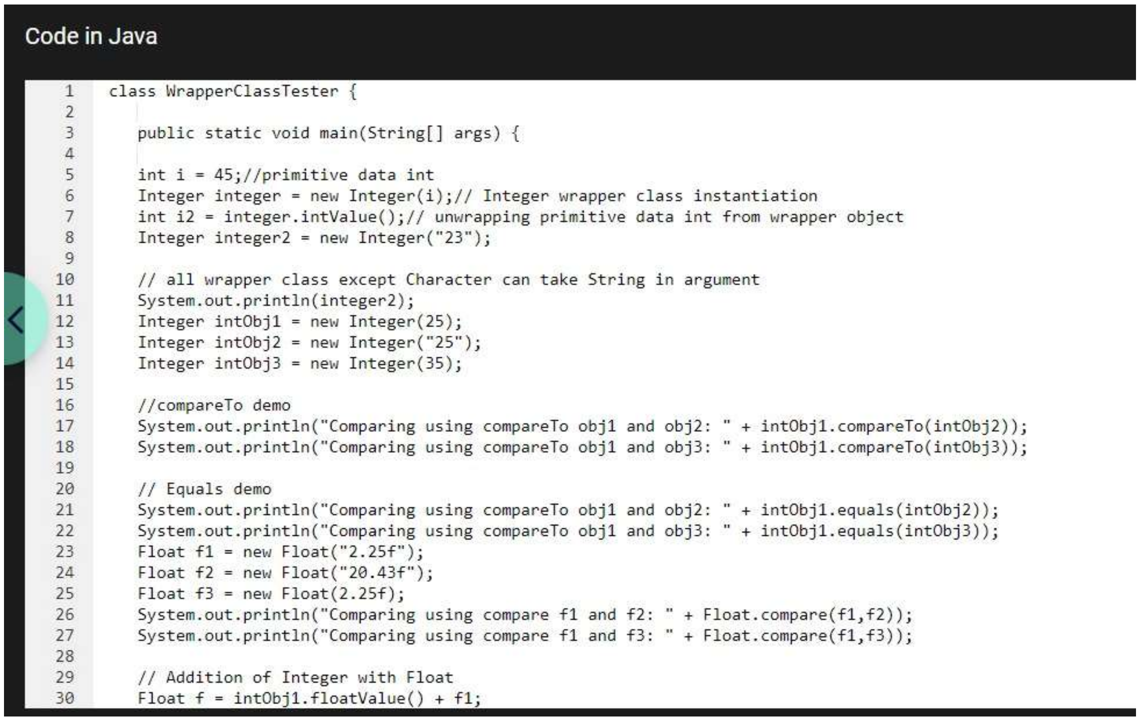


```java
        System.out.println("Email: " + p.email);
    }
}

class Execute{
    public static void main (String[] args) {
        Person p = new Person();
        Employee e = new Employee();
        Customer c =  new Customer();
        System.out.println("******************************");
        System.out.println("Person Class display method.");
        System.out.println("******************************");
        p.display();
        System.out.println("******************************");
        System.out.println("Employee Class display method.");
        System.out.println("******************************");
        e.display();
        System.out.println("******************************");
        System.out.println("Customer Class display method.");
        System.out.println("******************************");
        c.display();
    }
}
```



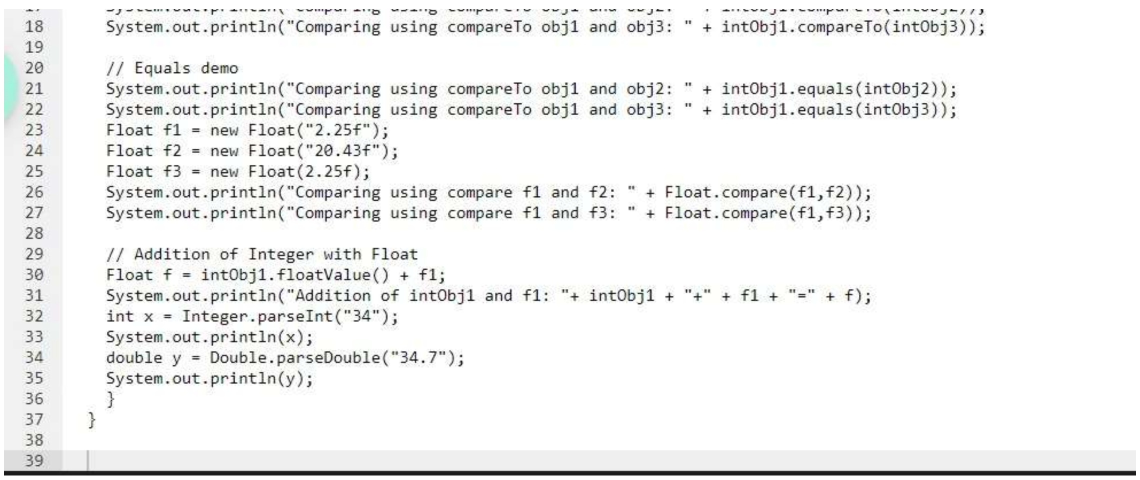
Output:

```
******************************
Person Class display method.
******************************
Name: Jack
Age: 24
Email: jack@samurai.com
Salary: 5000
******************************
Employee Class display method.
******************************
Name: Jack
Age: 24
Email: jack@samurai.com
******************************
Customer Class display method.
******************************
Name: Jack
Age: 24
Email: jack@samurai.com
```



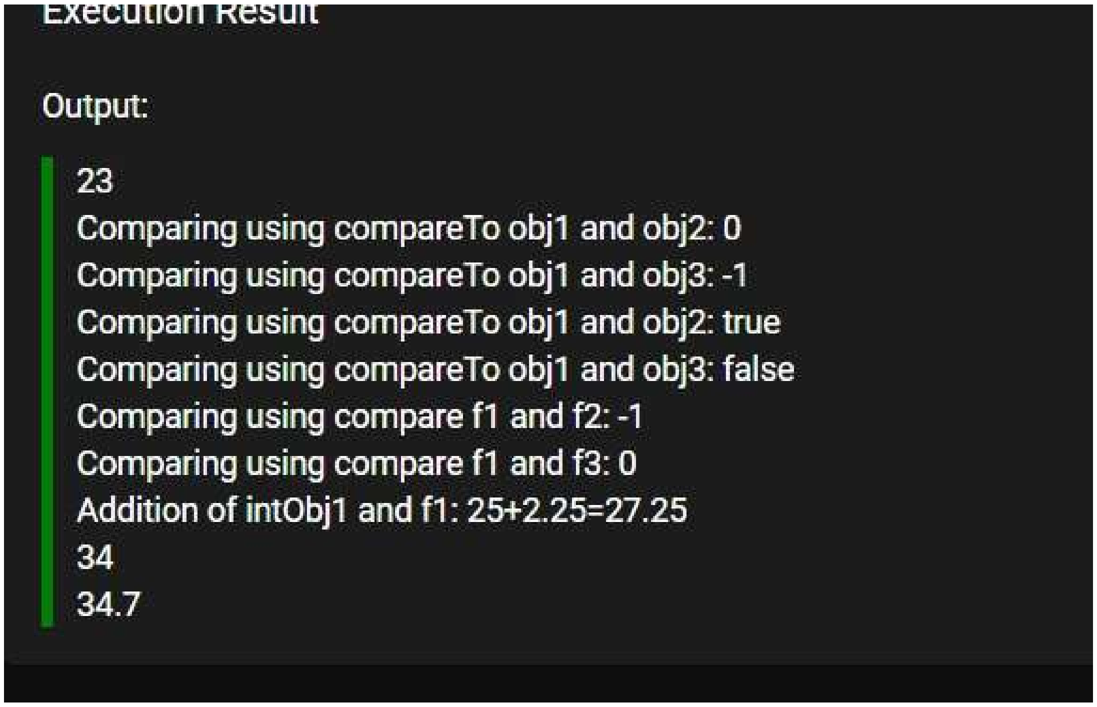
Code in Java

```
class WrapperClassTester {

    public static void main(String[] args) {

    int i = 45;//primitive data int
    Integer integer = new Integer(i);// Integer wrapper class instantiation
    int i2 = integer.intValue();// unwrapping primitive data int from wrapper object
    Integer integer2 = new Integer("23");

    // all wrapper class except Character can take String in argument
    System.out.println(integer2);
    Integer intObj1 = new Integer(25);
    Integer intObj2 = new Integer("25");
    Integer intObj3 = new Integer(35);

    //compareTo demo
    System.out.println("Comparing using compareTo obj1 and obj2: " + intObj1.compareTo(intObj2));
    System.out.println("Comparing using compareTo obj1 and obj3: " + intObj1.compareTo(intObj3));

    // Equals demo
    System.out.println("Comparing using compareTo obj1 and obj2: " + intObj1.equals(intObj2));
    System.out.println("Comparing using compareTo obj1 and obj3: " + intObj1.equals(intObj3));
    Float f1 = new Float("2.25f");
    Float f2 = new Float("20.43f");
    Float f3 = new Float(2.25f);
    System.out.println("Comparing using compare f1 and f2: " + Float.compare(f1,f2));
    System.out.println("Comparing using compare f1 and f3: " + Float.compare(f1,f3));

    // Addition of Integer with Float
    Float f = intObj1.floatValue() + f1;
```



```
    System.out.println("Comparing using compareTo obj1 and obj3: " + intObj1.compareTo(intObj3));

    // Equals demo
    System.out.println("Comparing using compareTo obj1 and obj2: " + intObj1.equals(intObj2));
    System.out.println("Comparing using compareTo obj1 and obj3: " + intObj1.equals(intObj3));
    Float f1 = new Float("2.25f");
    Float f2 = new Float("20.43f");
    Float f3 = new Float(2.25f);
    System.out.println("Comparing using compare f1 and f2: " + Float.compare(f1,f2));
    System.out.println("Comparing using compare f1 and f3: " + Float.compare(f1,f3));

    // Addition of Integer with Float
    Float f = intObj1.floatValue() + f1;
```

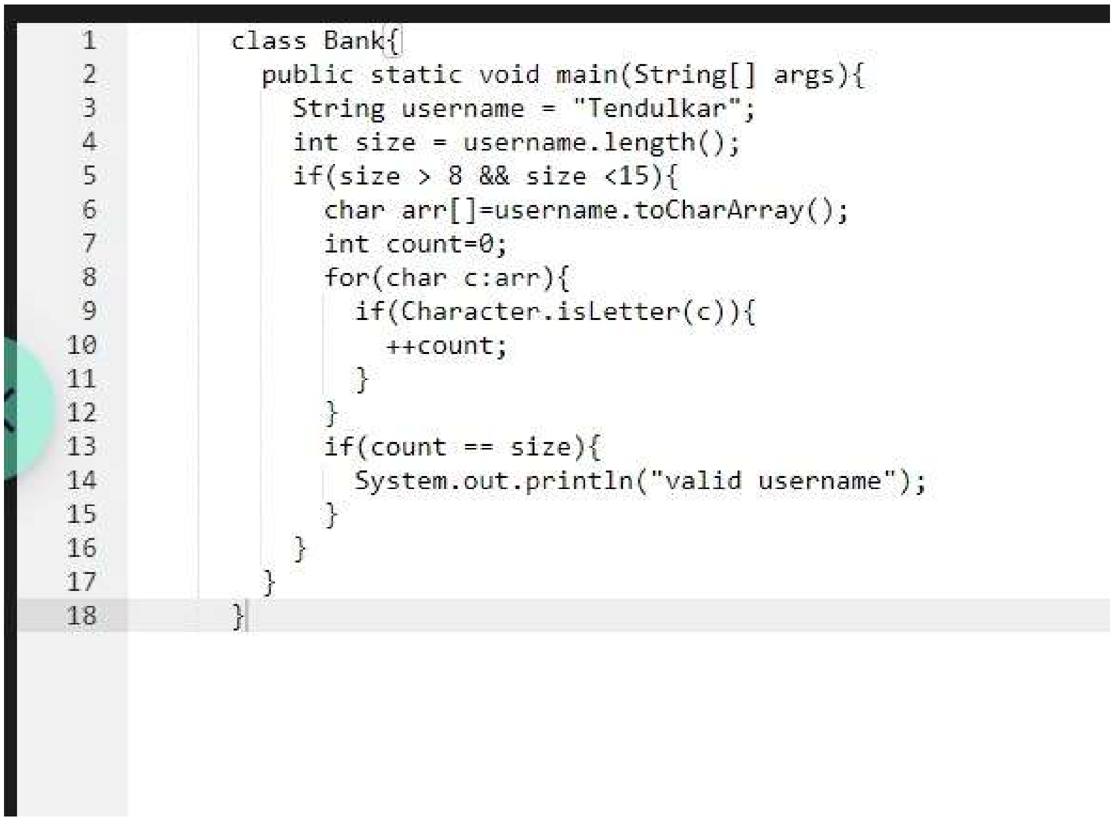
```
    System.out.println("Addition of intObj1 and f1: "+ intObj1 + "+" + f1 + "=" + f);
    int x = Integer.parseInt("34");
    System.out.println(x);
    double y = Double.parseDouble("34.7");
    System.out.println(y);
    }
}

```



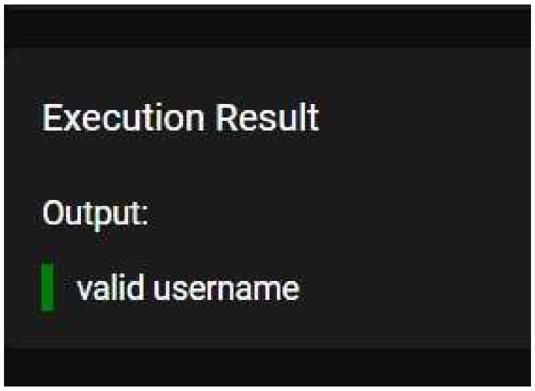
Execution Result

Output:

```
23
Comparing using compareTo obj1 and obj2: 0
Comparing using compareTo obj1 and obj3: -1
Comparing using compareTo obj1 and obj2: true
Comparing using compareTo obj1 and obj3: false
Comparing using compare f1 and f2: -1
Comparing using compare f1 and f3: 0
Addition of intObj1 and f1: 25+2.25=27.25
34
34.7
```

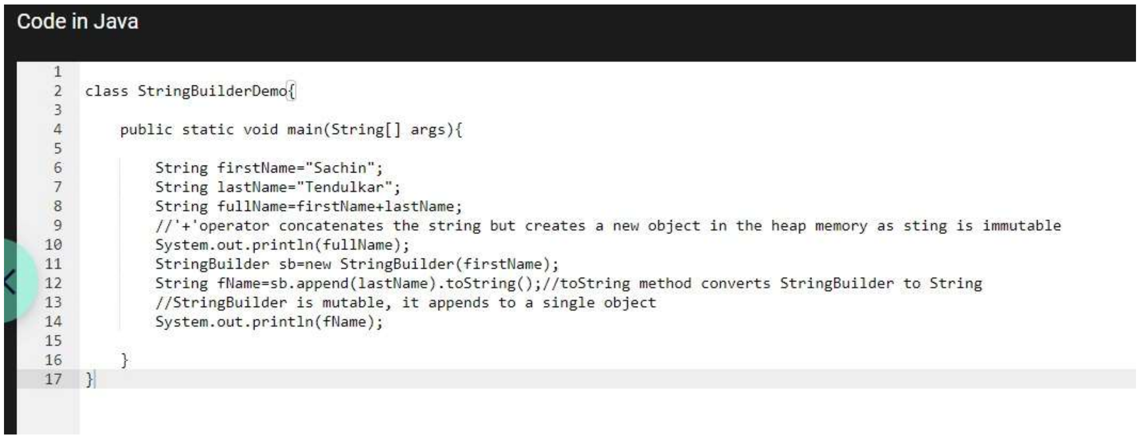


```java
class Bank{
  public static void main(String[] args){
    String username = "Tendulkar";
    int size = username.length();
    if(size > 8 && size <15){
      char arr[]=username.toCharArray();
      int count=0;
      for(char c:arr){
        if(Character.isLetter(c)){
          ++count;
        }
      }
      if(count == size){
        System.out.println("valid username");
      }
    }
  }
}
```

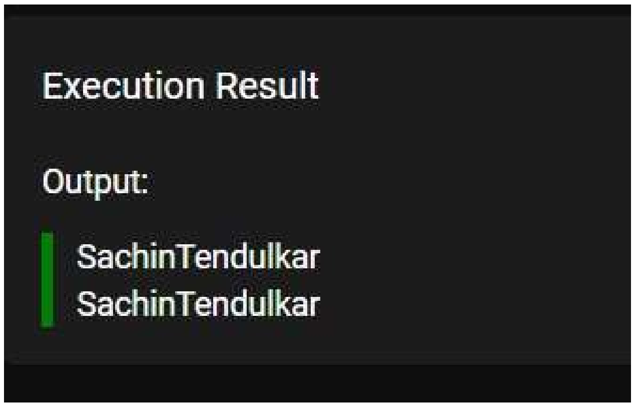

Execution Result

Output:

```
valid username
```

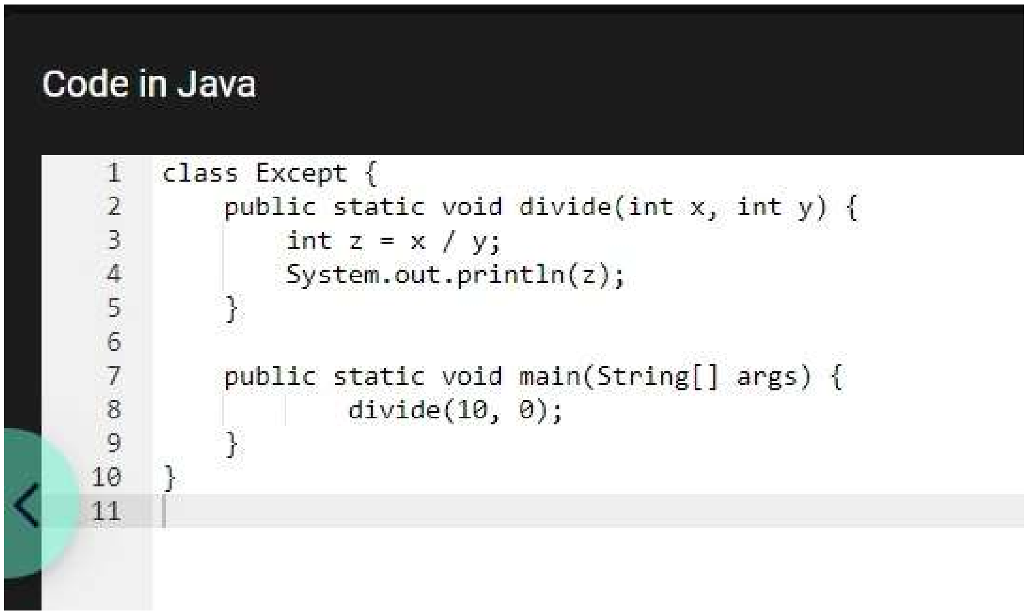

Code in Java

```java

class StringBuilderDemo{

    public static void main(String[] args){

        String firstName="Sachin";
        String lastName="Tendulkar";
        String fullName=firstName+lastName;
        //'+'operator concatenates the string but creates a new object in the heap memory as sting is immutable
        System.out.println(fullName);
        StringBuilder sb=new StringBuilder(firstName);
        String fName=sb.append(lastName).toString();//toString method converts StringBuilder to String
        //StringBuilder is mutable, it appends to a single object
        System.out.println(fName);

    }
}
```

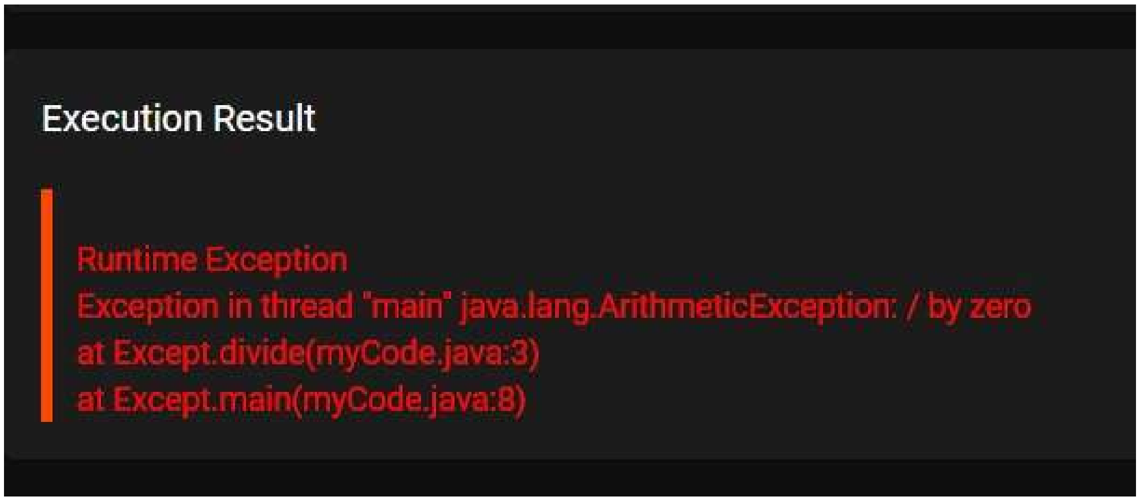

Execution Result

Output:

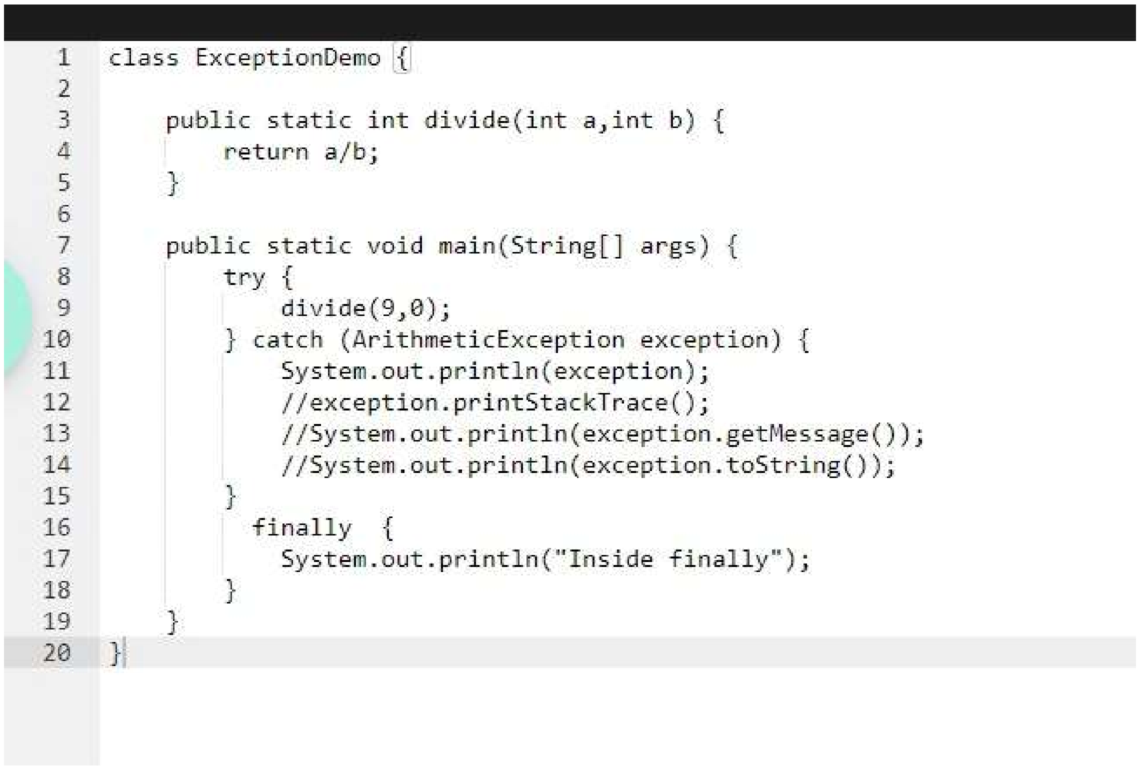
```
SachinTendulkar
SachinTendulkar
```


Code in Java

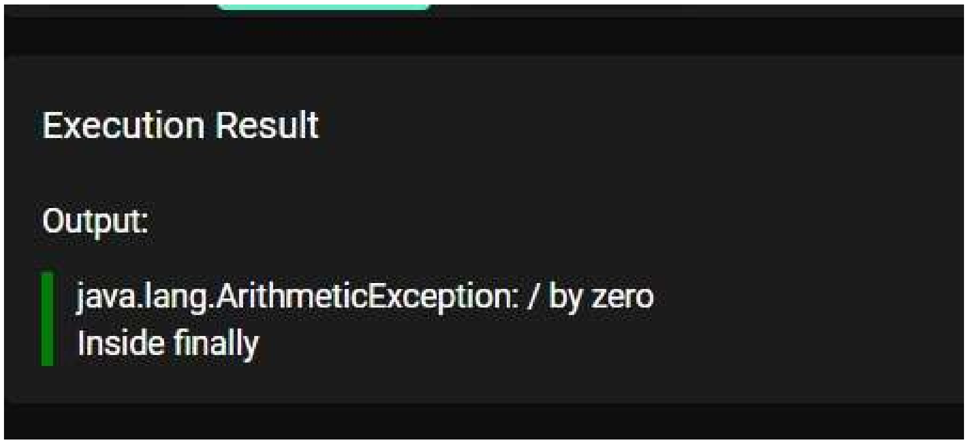
```java
 1  class Except {
 2      public static void divide(int x, int y) {
 3          int z = x / y;
 4          System.out.println(z);
 5      }
 6
 7      public static void main(String[] args) {
 8              divide(10, 0);
 9      }
10  }
11
```


Execution Result

```
Runtime Exception
Exception in thread "main" java.lang.ArithmeticException: / by zero
at Except.divide(myCode.java:3)
at Except.main(myCode.java:8)
```



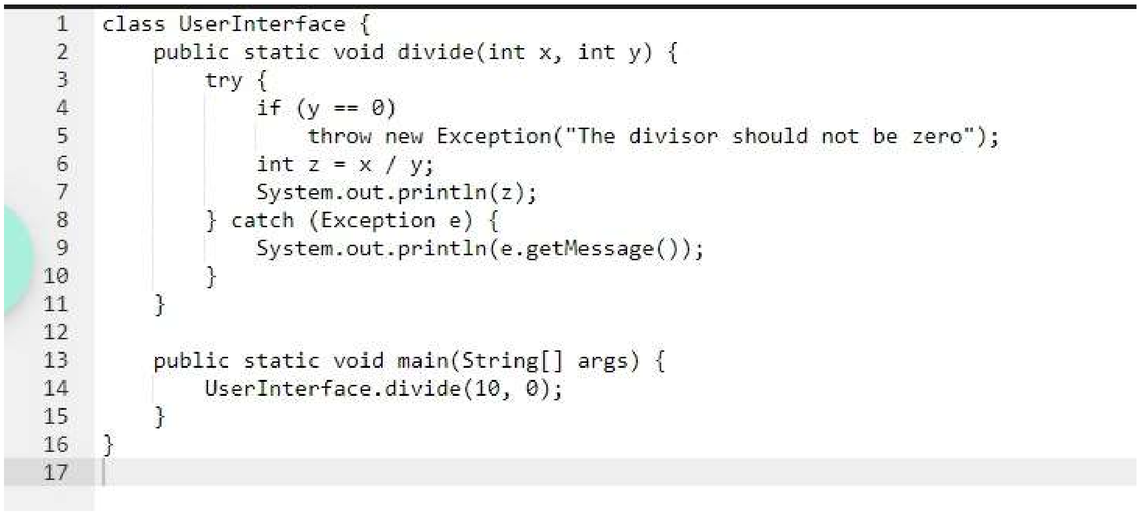
```java
class ExceptionDemo {

    public static int divide(int a,int b) {
        return a/b;
    }

    public static void main(String[] args) {
        try {
            divide(9,0);
        } catch (ArithmeticException exception) {
            System.out.println(exception);
            //exception.printStackTrace();
            //System.out.println(exception.getMessage());
            //System.out.println(exception.toString());
        }
          finally  {
            System.out.println("Inside finally");
        }
    }
}
```



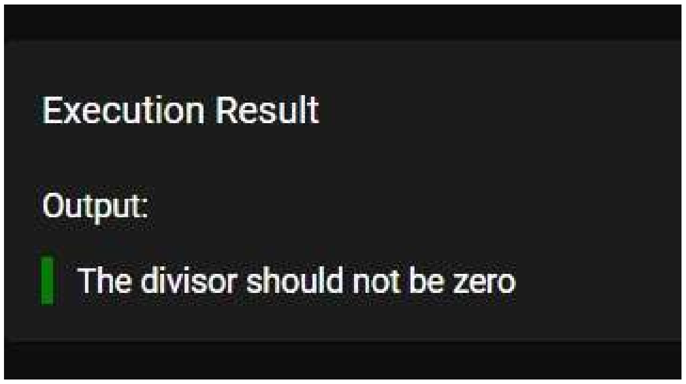
Execution Result

Output:

```
java.lang.ArithmeticException: / by zero
Inside finally
```



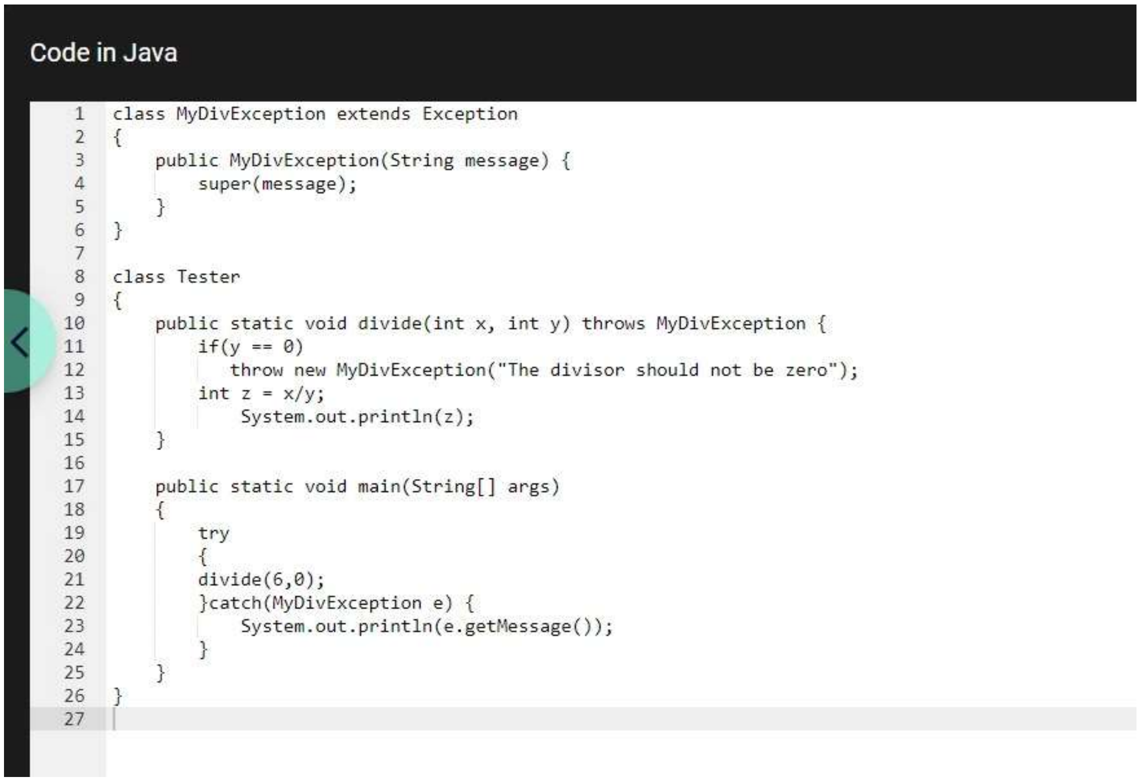
```java
class UserInterface {
    public static void divide(int x, int y) {
        try {
            if (y == 0)
                throw new Exception("The divisor should not be zero");
            int z = x / y;
            System.out.println(z);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    public static void main(String[] args) {
        UserInterface.divide(10, 0);
    }
}
```



Execution Result

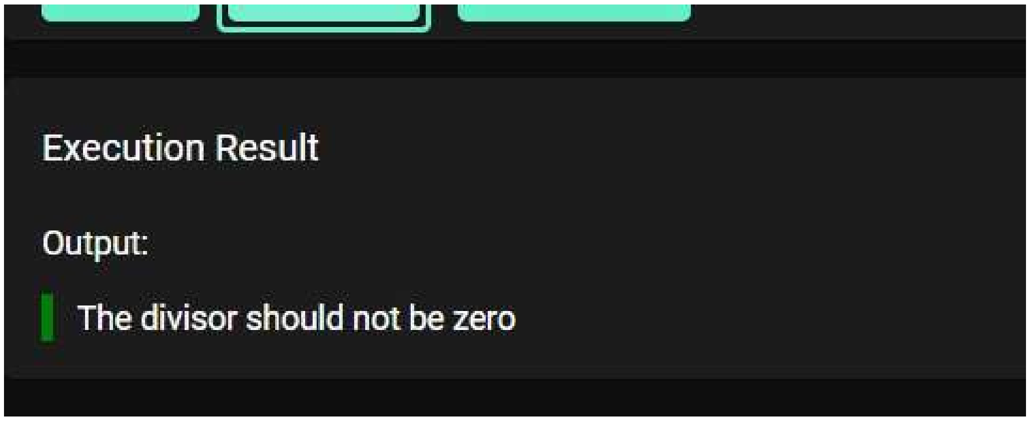
Output:

```
The divisor should not be zero
```



Code in Java

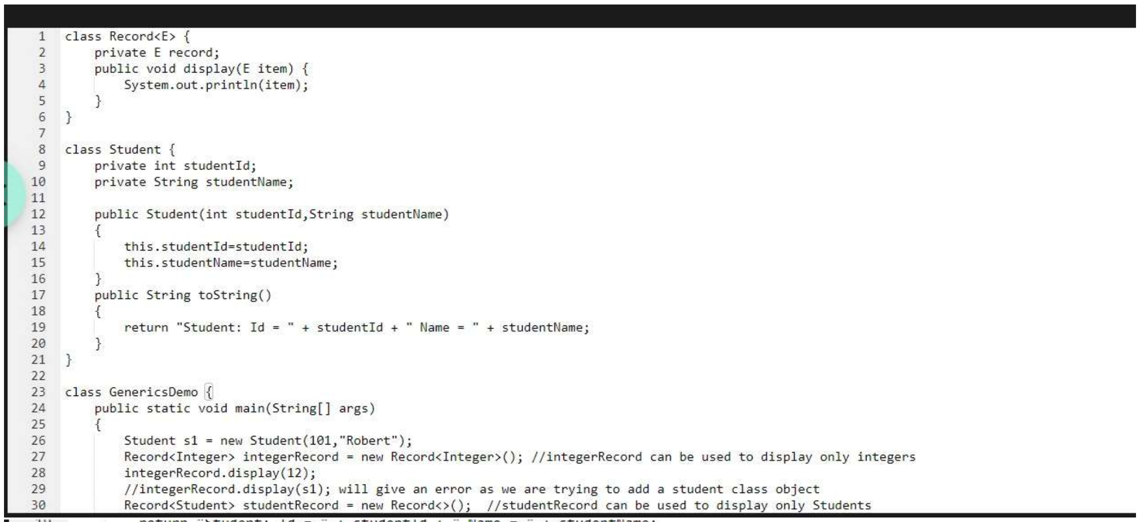
```java
class MyDivException extends Exception
{
    public MyDivException(String message) {
        super(message);
    }
}

class Tester
{
    public static void divide(int x, int y) throws MyDivException {
        if(y == 0)
           throw new MyDivException("The divisor should not be zero");
        int z = x/y;
            System.out.println(z);
    }

    public static void main(String[] args)
    {
        try
        {
        divide(6,0);
        }catch(MyDivException e) {
            System.out.println(e.getMessage());
        }
    }
}
```



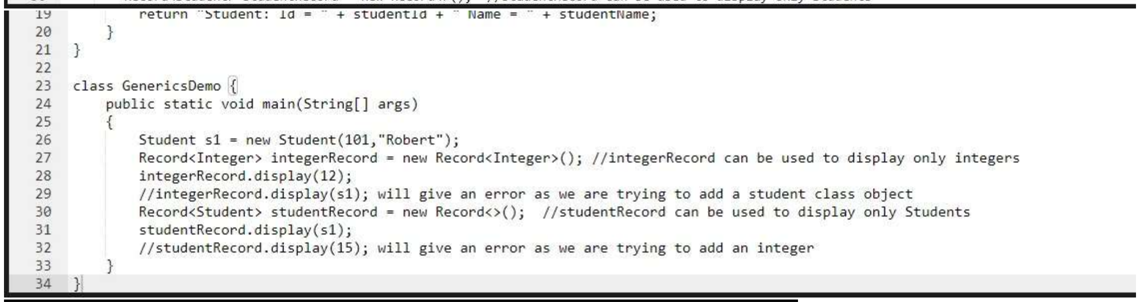
Execution Result

Output:

```
The divisor should not be zero
```

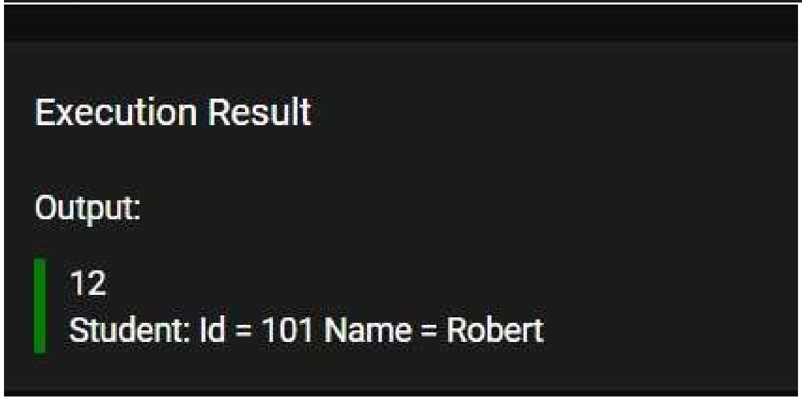


```java
class Record<E> {
    private E record;
    public void display(E item) {
        System.out.println(item);
    }
}

class Student {
    private int studentId;
    private String studentName;

    public Student(int studentId,String studentName)
    {
        this.studentId=studentId;
        this.studentName=studentName;
    }
    public String toString()
    {
        return "Student: Id = " + studentId + " Name = " + studentName;
    }
}

class GenericsDemo {
    public static void main(String[] args)
    {
        Student s1 = new Student(101,"Robert");
        Record<Integer> integerRecord = new Record<Integer>(); //integerRecord can be used to display only integers
        integerRecord.display(12);
        //integerRecord.display(s1); will give an error as we are trying to add a student class object
        Record<Student> studentRecord = new Record<>();  //studentRecord can be used to display only Students
```



```java
        return "Student: Id = " + studentId + " Name = " + studentName;
    }
}

class GenericsDemo {
    public static void main(String[] args)
    {
        Student s1 = new Student(101,"Robert");
        Record<Integer> integerRecord = new Record<Integer>(); //integerRecord can be used to display only integers
        integerRecord.display(12);
        //integerRecord.display(s1); will give an error as we are trying to add a student class object
        Record<Student> studentRecord = new Record<>();  //studentRecord can be used to display only Students
```

```java
        studentRecord.display(s1);
        //studentRecord.display(15); will give an error as we are trying to add an integer
    }
}
```



Execution Result

Output:

```
12
Student: Id = 101 Name = Robert
```