JAVA – WEEK 3

CONSTRUCTORS & CLASSES

CHIRAN JEEVI

2019103013

LEVEL-1

1) Define a class called with data members : numerator(int) and denominator(int). The member methods are constructor with two arguments of type integer, default constructor in which always denominator is initialised with value 1, set and get methods, void display to print the details of the object Rational. Create a Test class to create objects of Rational and invoke the methods. Validate the input in the set methods.

CODE:

Rational.java

```
public class Rational
{
  int num;
  int denom;

  //DEFAULT CONSTRUCTOR
  public Rational () {
    denom = 1;
  }
  //PARAMETERIZED CONSTRUCTOR
  public Rational (int n, int d) {
    num = n;
    denom = d;
  }

  //GET METHODS: only return type
  public int getNum () {
    return num;
  }
  public int getDenom () {
```

```
    return denom;
  }

  //SET METHODS: only parameter, NO return
  public void setNum (int n) {
    num = n;
  }
  public void setDenom (int d) {
    denom = d;
  }

  //DISPLAY
  public void display () {
    System.out.println(" Numerator = " + num + " Denominator = " + denom);
  }
}
```

TestRational.java

```
import java.util.*;

public class TestRational
{
  public static void main(String[] args)
  {
    Scanner input = new Scanner(System.in);

    int n, d;

    System.out.print("\n Enter Numerator   : ");
    n = input.nextInt();
    System.out.print(" Enter Denominator : ");
    d = input.nextInt();

    Rational r1 = new Rational();
    Rational r2 = new Rational(n,d);

    System.out.println();

    r1.display();
    r2.display();
```

```
        System.out.println();

        System.out.println(" r2.getNum() = " + r2.getNum() + "\n r2.getDenom = " + r2.getDenom());

        r1.setNum(89);
        r1.setDenom(34);
        System.out.println("\n New r1 after setting num & denom : ");
        r1.display();

        System.out.println();

        input.close();
    }
}
```

OUTPUT:

```
Windows PowerShell
PS C:\Users\Chiran\Desktop\ONLINE CLASS\SEM 5\JAVA LAB\W3> javac TestRational.java
PS C:\Users\Chiran\Desktop\ONLINE CLASS\SEM 5\JAVA LAB\W3> java TestRational

 Enter Numerator   : 45
 Enter Denominator : 5

 Numerator = 0 Denominator = 1
 Numerator = 45 Denominator = 5

 r2.getNum() = 45
 r2.getDenom = 5

 New r1 after setting num & denom :
 Numerator = 89 Denominator = 34

PS C:\Users\Chiran\Desktop\ONLINE CLASS\SEM 5\JAVA LAB\W3> java TestRational

 Enter Numerator   : 3456
 Enter Denominator : 789

 Numerator = 0 Denominator = 1
 Numerator = 3456 Denominator = 789

 r2.getNum() = 3456
 r2.getDenom = 789

 New r1 after setting num & denom :
 Numerator = 89 Denominator = 34

PS C:\Users\Chiran\Desktop\ONLINE CLASS\SEM 5\JAVA LAB\W3>
```

2) Define a class called Employee with datamembers: empid(int), firstname(String), lastname(String), salary(double). The salary is initialised with value of 1000 while declaring. The member methods are:

Employee()

Employee (int id, String fn, String ln)

setEmpid(int)

setFirstName(String)

setLastName(String)

int getEmpid()

String getFirstName()

String getLastName( )

double getSalary()

void display() – call the get methods to print the instance fields using this reference

String toString()

Validate the inputs in set methods.

Create a test class to create objects and test the methods

Employee.java

```
import java.util.*;

public class Employee
{
   int empid;
   String firstname, lastname;
   double salary = 1000;
   //int count=0;
   //int [] eidArr = new int[20];
   int totalEmployees=0;

   public Employee () {
      empid = 0;
      firstname = null;
      lastname  = null;
      totalEmployees++;
   }

   public Employee (int id, String fn, String ln, double sal)
   {
      setEmpid(id);
      setFirstname(fn);
      setLastname(ln);
      setSalary(sal);

      totalEmployees++;
   }

   //SET
   public void setEmpid (int id) {
   if (id>0) {
      empid = id;
   }
      else
         System.out.println("\n invalid Employee Id");
   }
   public void setFirstname (String fn) {
      if (fn!=null)
         firstname = String.valueOf(fn);
      else
```

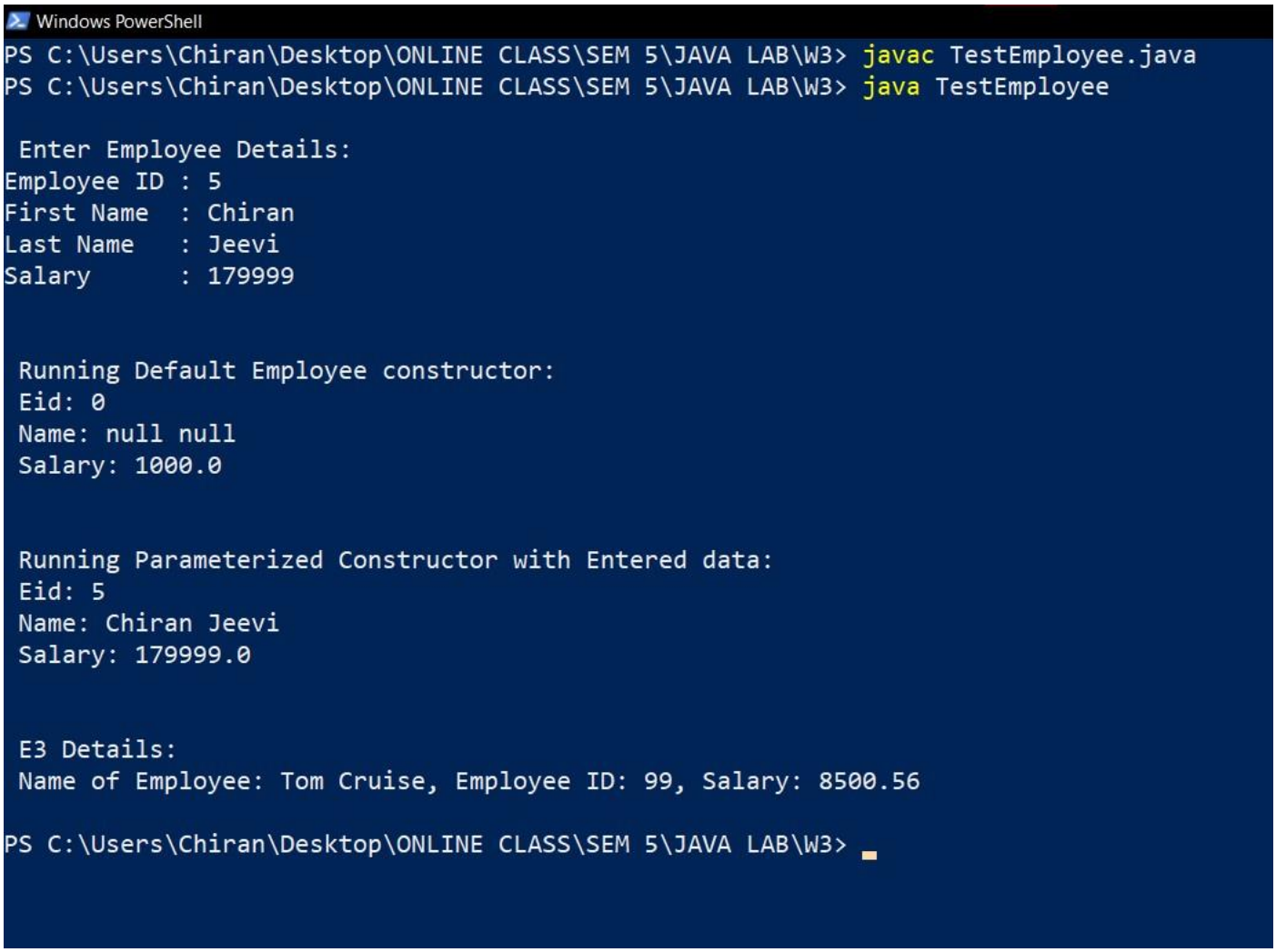
```
        System.out.println("\n invalid firstname");
    }
    public void setLastname (String ln) {
        if (ln!=null)
            lastname  = String.valueOf(ln);
        else
            System.out.println("\n invalid lastname");
    }
    public void setSalary (double sal) {
        if (sal>0)
            salary = sal;
        else
            System.out.println("\n invalid salary");
    }

    //GET
    public int getEmpid () {
        return empid;
    }
    public String getFirstname () {
        return firstname;
    }
    public String getLastname () {
        return lastname;
    }
    public double getSalary () {
        return salary;
    }

    public void display ()
    {
        System.out.println(" Eid: " + empid);
        System.out.println(" Name: " + firstname + " " + lastname);
        System.out.println(" Salary: " + salary);
    }

    public String toString () {
        return String.format(" Name of Employee: %s %s, Employee ID: %d, Salary: %.2f", firstname, lastname,
empid, salary);
    }
}
```

TestEmployee.java

```
import java.util.Scanner;

public class TestEmployee
{
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);

      int eid;
      String fname = new String(), lname = new String();
      double sal;

      System.out.println("\n Enter Employee Details: ");

      System.out.print("Employee ID : ");
      eid = input.nextInt();
      input.nextLine();

      System.out.print("First Name  : ");
      fname = input.nextLine();

      System.out.print("Last Name   : ");
      lname = input.nextLine();

      System.out.print("Salary      : ");
      sal = input.nextDouble();

      System.out.println("\n\n Running Default Employee constructor: ");
      Employee e1 = new Employee();
      e1.display();

      System.out.println("\n\n Running Parameterized Constructor with Entered data:");
      Employee e2 = new Employee(eid, fname, lname, sal);
      e2.display();

      Employee e3 = new Employee();
      e3.setEmpid(99);
      e3.setFirstname("Tom");
      e3.setLastname("Cruise");
      e3.setSalary(8500.56);
```

```
    String e3Deets = e3.toString();
    System.out.println("\n\n E3 Details: ");
    System.out.println(e3Deets + "\n");


    input.close();
  }
}
```

OUTPUT:



LEVEL-2

1. Define a class called Book with data members: title(String), authorName(String), year of publication(int) and price(double). The methods are: default constructor, constructor with four arguments, void display() that prints the details of Book object using Sytem.out.printf.

Define a test class in which an array of Book Objects are created and the details of Book Object printed. The size of the array has to be obtained from the user.

Book.java

```
public class Book
{
   String title = new String();
   String author = new String();
   int yearOfPublication;
   double price;

   public Book ()
   {
      title = null;
      author = null;
      yearOfPublication = 0;
      price = 500.00;
   }

   public Book (String book, String writer, int year, double cost)
   {
      setTitle ( book);
      setAuthor ( writer);
      setYearOfPublication ( year);
      setPrice ( cost);
   }

   public void setTitle (String book) {
      if (book!=null)
         title = String.valueOf(book);
      else
         title = null;
   }
   public void setAuthor (String writer)
```

```
  {
    if (writer!=null)
      author = String.valueOf(writer);
    else
      author = null;
  }
  public void setYearOfPublication (int year) {
    if (year>0)
      yearOfPublication = year;
    else
      yearOfPublication = 0;
  }
  public void setPrice (double cost) {
    if (cost>0)
      price = cost;
    else
      price = 0;
  }


  public void display ()
  {
    System.out.println(" Book Details: ");
    System.out.printf("Title: %s \nAuthor: %s \nYear Of Publication: %d \nPrice: %.2f\n \n", title, author, y
earOfPublication, price);
  }

}
```

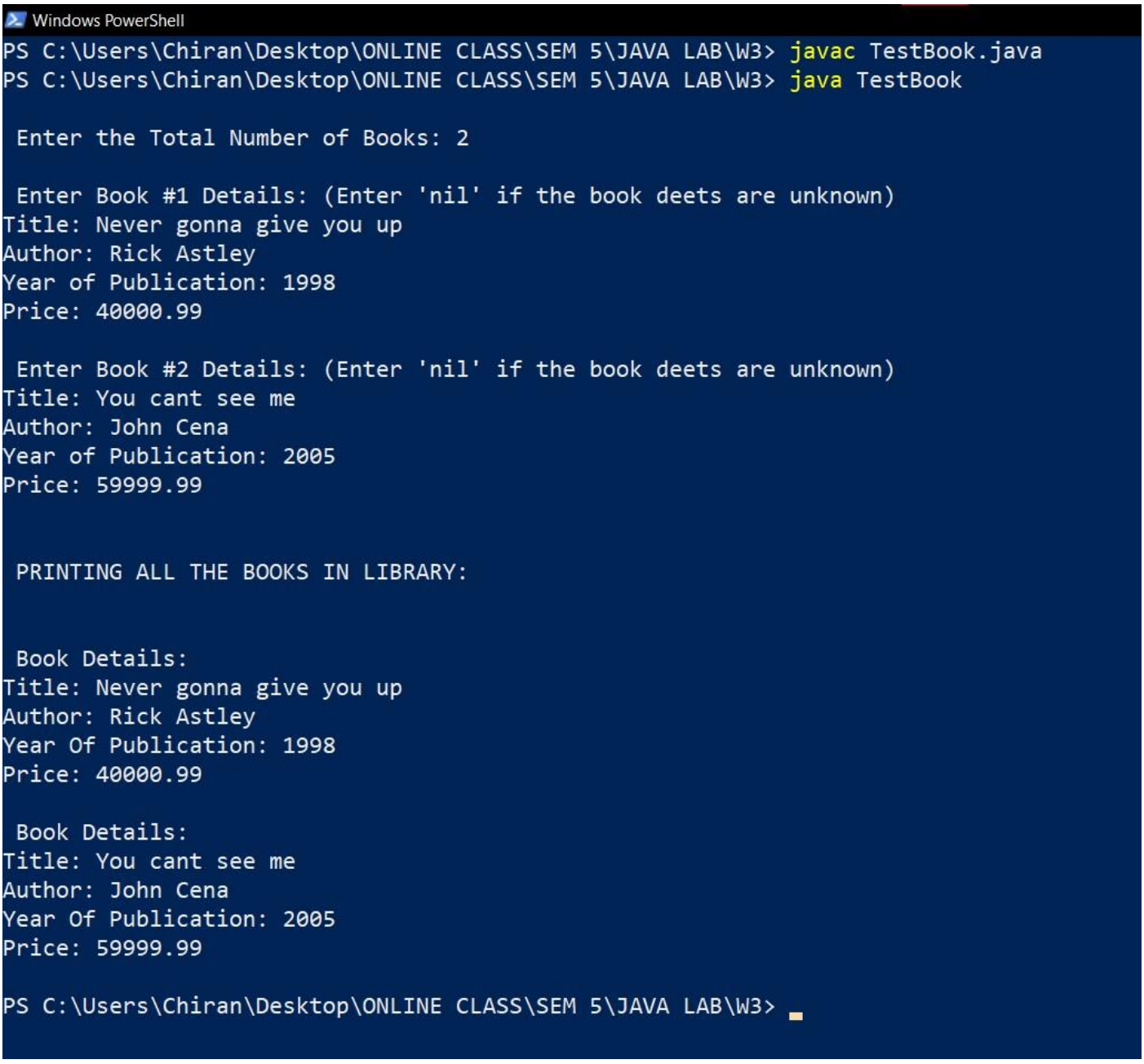
TestBook.java

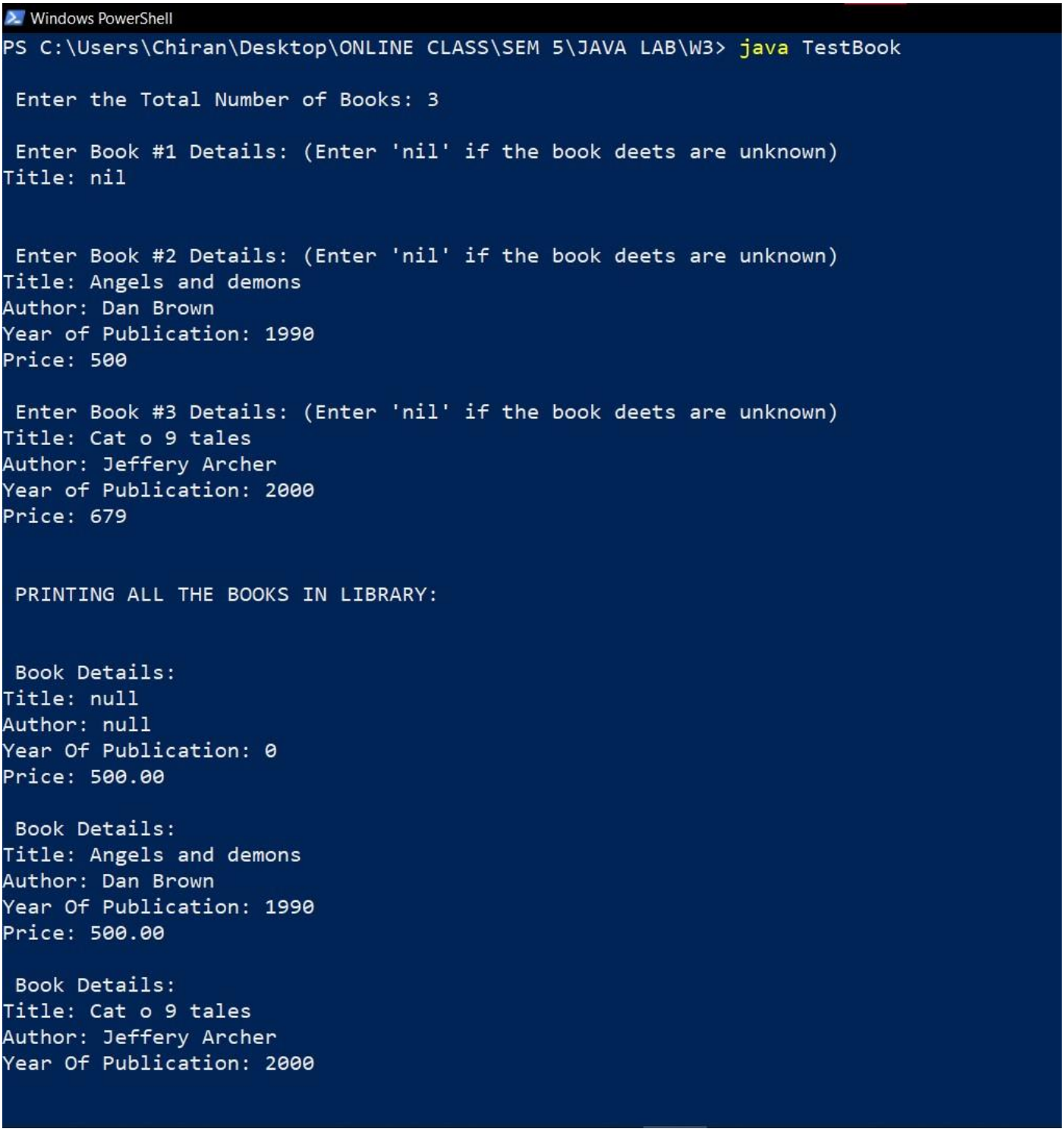
```
import java.util.Scanner;

public class TestBook
{
  public static void main(String[] args)
  {
    Scanner input = new Scanner(System.in);

    int i, n;
```

```
System.out.print("\n Enter the Total Number of Books: ");
n = input.nextInt();

Book [] library = new Book[n];

for (i=0; i<n; i++)
{
   input.nextLine();

   String book, writer;
   int year;
   double cost;

   System.out.println("\n Enter Book #" + (i+1) + " Details: (Enter 'nil' if the book deets are unknown)");

   System.out.print("Title: ");
   book = input.nextLine();

   if (book.equals("nil"))
   {
      //CALLS DEFAULT CONSTRUCTOR
      library[i] = new Book();
      continue;
   }

   System.out.print("Author: ");
   writer = input.nextLine();

   System.out.print("Year of Publication: ");
   year =  input.nextInt();

   System.out.print("Price: ");
   cost = input.nextDouble();

   library[i] = new Book(book, writer, year, cost);
}

System.out.println("\n");

System.out.println(" PRINTING ALL THE BOOKS IN LIBRARY: \n\n");
for (i=0; i<n; i++)
{
```

```
        library[i].display();
    }

    input.close();

  }
}
```

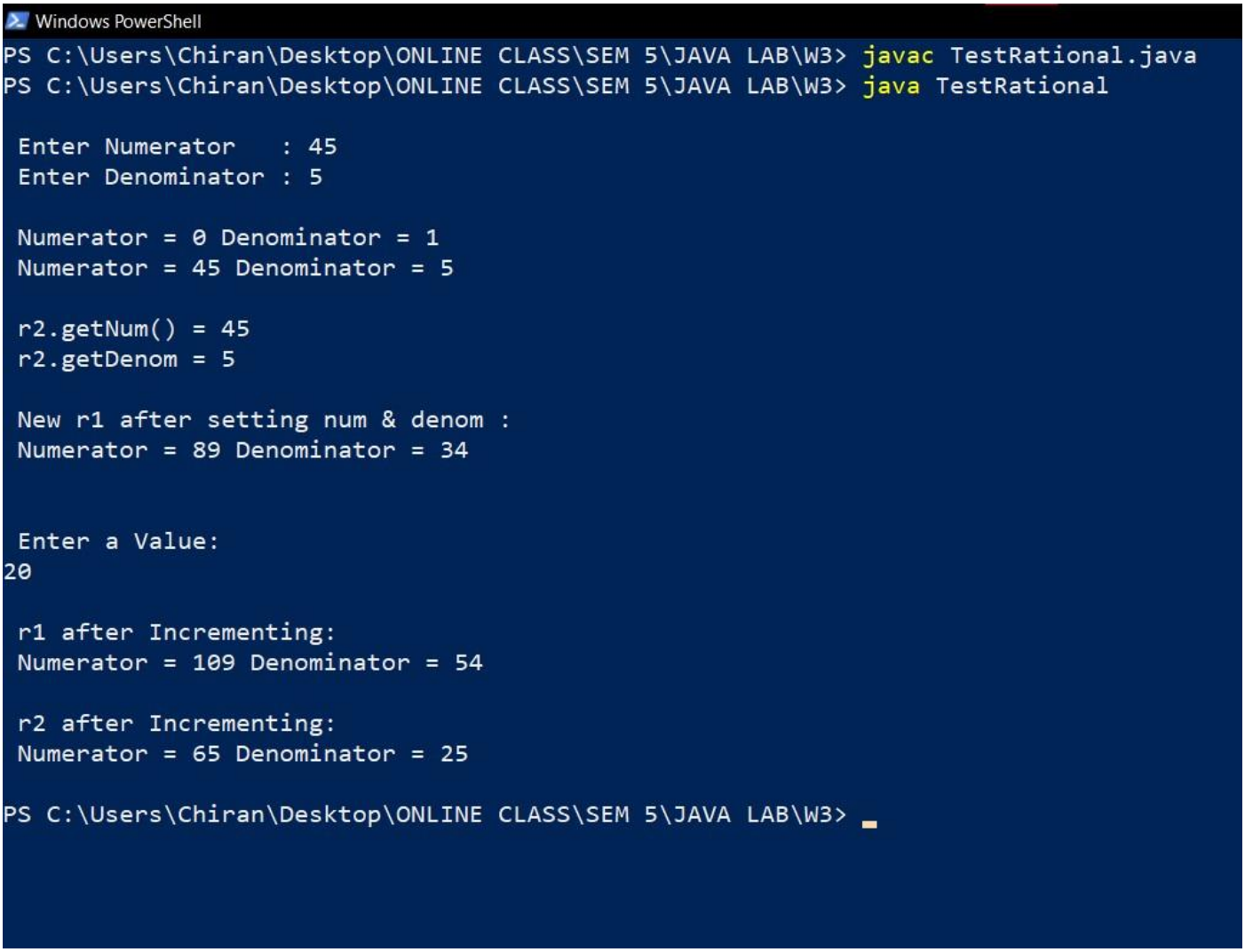
OUTPUT:



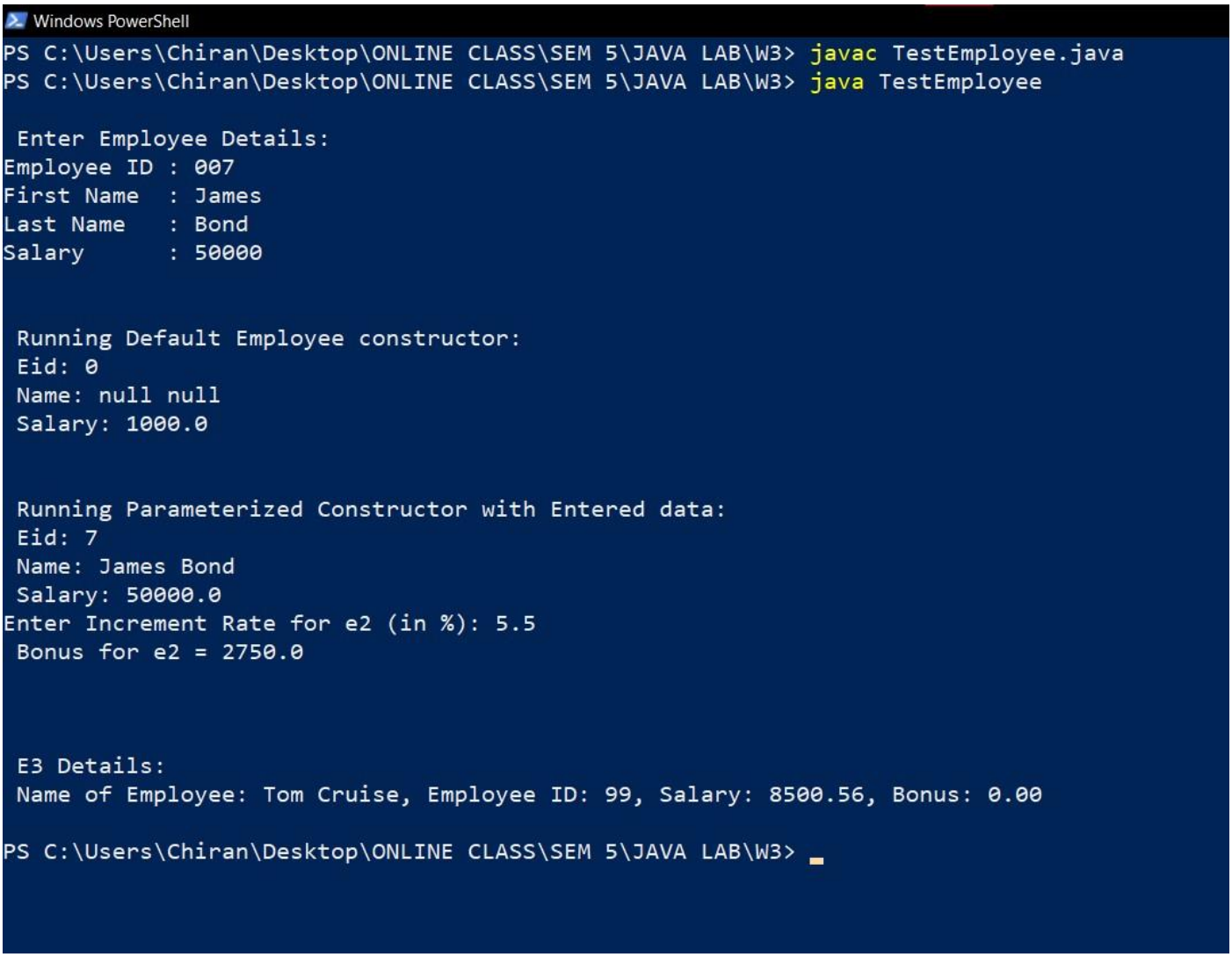

Windows PowerShell
PS C:\Users\Chiran\Desktop\ONLINE CLASS\SEM 5\JAVA LAB\W3> java TestBook
Enter the Total Number of Books: 3
Enter Book #1 Details: (Enter 'nil' if the book deets are unknown)
Title: nil
Enter Book #2 Details: (Enter 'nil' if the book deets are unknown)
Title: Angels and demons
Author: Dan Brown
Year of Publication: 1990
Price: 500
Enter Book #3 Details: (Enter 'nil' if the book deets are unknown)
Title: Cat o 9 tales
Author: Jeffery Archer
Year of Publication: 2000
Price: 679
PRINTING ALL THE BOOKS IN LIBRARY:
Book Details:
Title: null
Author: null
Year Of Publication: 0
Price: 500.00
Book Details:
Title: Angels and demons
Author: Dan Brown
Year Of Publication: 1990
Price: 500.00
Book Details:
Title: Cat o 9 tales
Author: Jeffery Archer
Year Of Publication: 2000

LEVEL-3

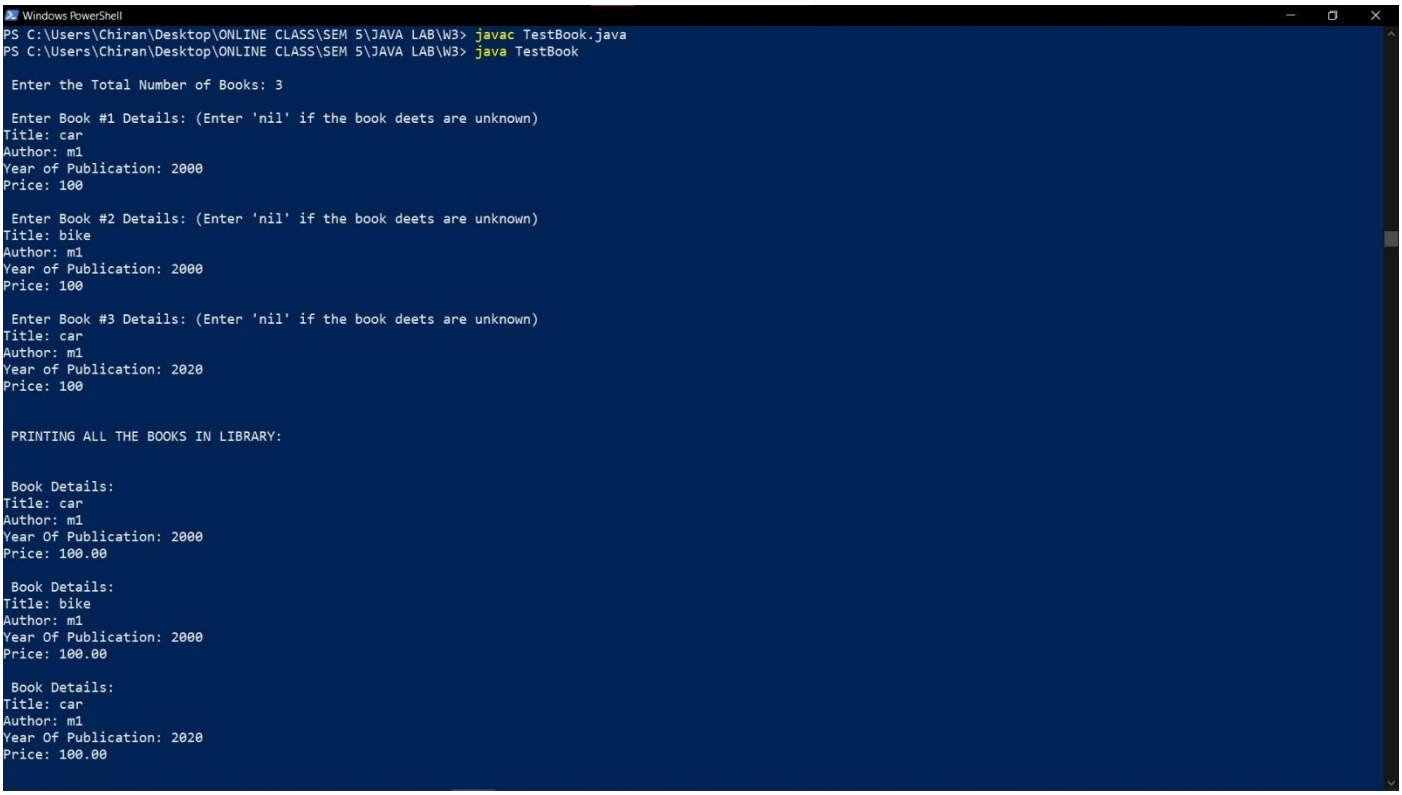
1. Define a method void increment(int val) for the class Rational that increments the numerator and denominator by the val. Print the details of Rational object before and after calling of increment. (2)

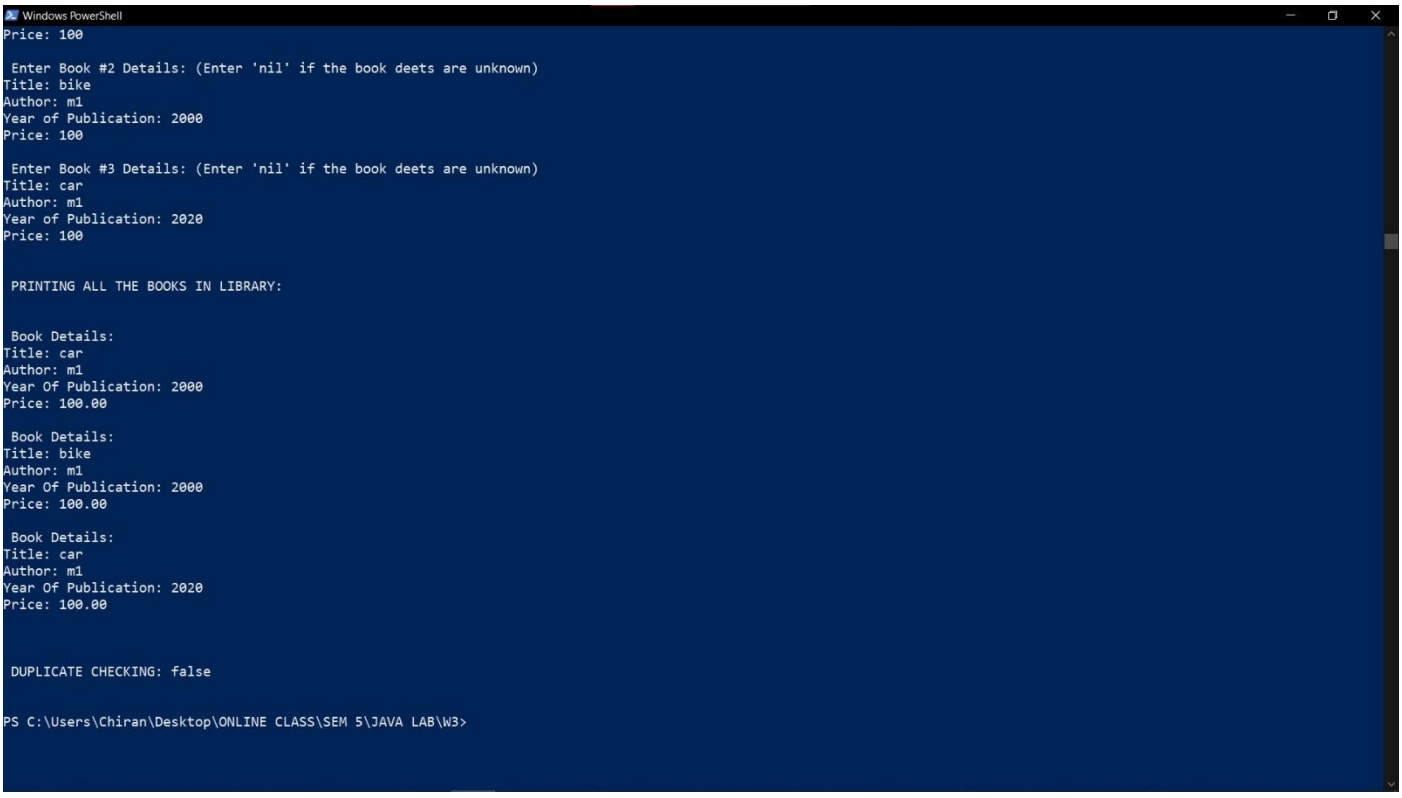


2. Define a method double calculateBonus(double intRate) in class Employee that calculates the bonus (bonus = salary * intRate) when it is called by the object of Employee. (1)



3. Define method Boolean checkDup(Book []) that checks whether duplicate details of book exist in the array of Book.





	Marks
Preparatory Exercises	
Observation	
Spot	
Total	