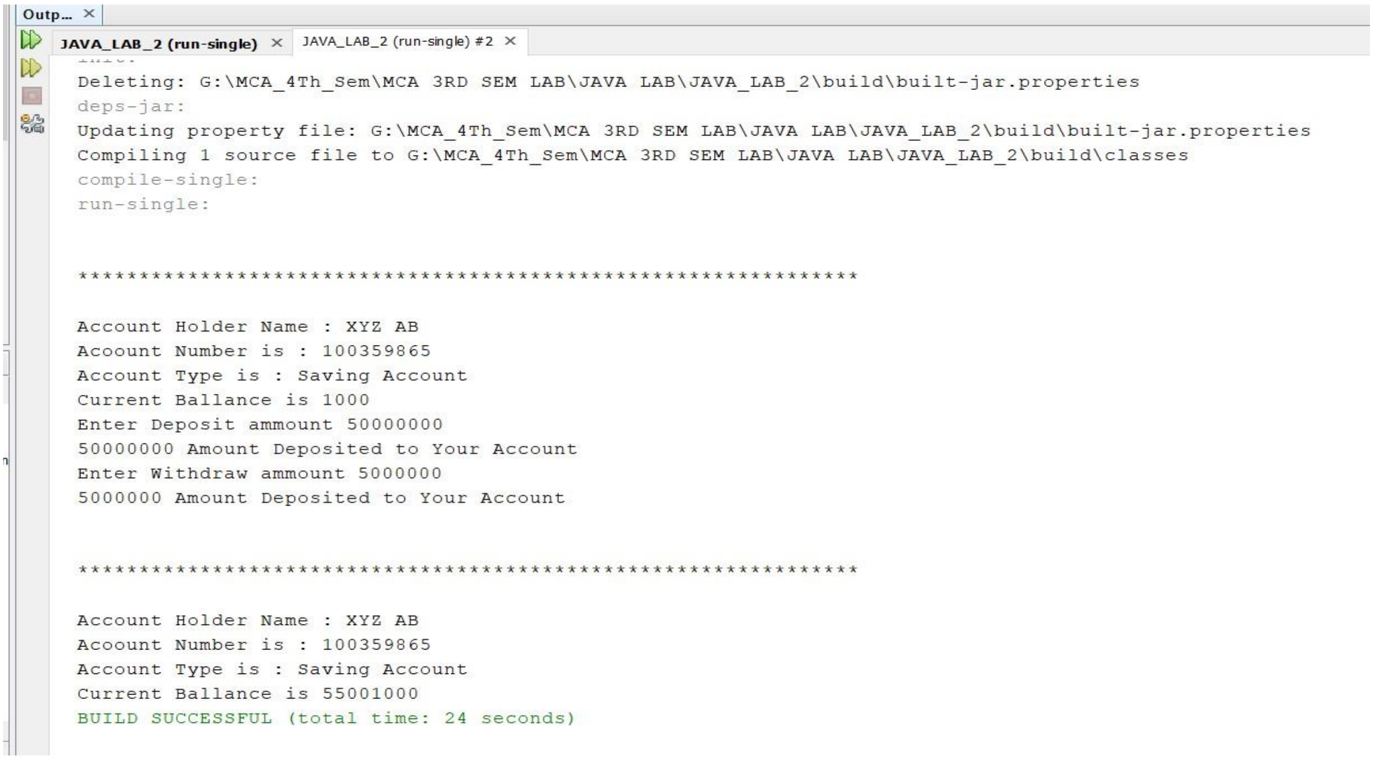
Java Lab Assignment – 2

Name : Rahul Gupta

Roll: MCA/10030/19

Prog-1 Create a class Account with two overloaded constructors. First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, address, type of account and current balance. Account class is having methods Deposit(), WithDraw(), and GetBalance().Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

```
package java_lab_2.QUES_1;


import java.util.Scanner;


public class Account {
    private String accountHolder;
    private int accountNumber;


    private String address;
    private String accountType;
    private int currentBallance;
    public Account(String accountHolder,int accountNumber,int ammount)
    {
      this.accountHolder=accountHolder;
      this.accountNumber=accountNumber;
      this.currentBallance=ammount;
    }
    public Account(String accountHolder,int accountNumber,String address,String
accountType,int currentBalance)
    {
        this.accountHolder=accountHolder;
      this.accountNumber=accountNumber;
      this.address=address;
      this.accountType=accountType;
      this.currentBallance=currentBalance;
    }
    public void deposit(int dept)
    {
        currentBallance=currentBallance+dept;
        System.out.println(dept+" Amount Deposited to Your Account");



    }
    public void withdraw(int withD)
    {
        this.currentBallance=this.currentBallance-withD;
        System.out.println(withD+" Ammount WithDrawn From Your Account Successfully");


    }
    public void getBalance()
    {

System.out.println("\n\n********************************************************
*********\n\n"+"Account Holder Name : "+accountHolder+"\nAcoount Number is :
"+accountNumber+"\nAccount Type is : "+accountType+"\nCurrent Ballance is
"+currentBallance);
    }


    public static void main(String []args)
    {
        Account ac=new Account("XYZ AB",100359865,"xyz city Madhya-Pradesh
102","Saving Account",1000);
       ac.getBalance();
      System.out.print("Enter Deposit ammount ");
       Scanner sc=new Scanner(System.in);
        int depositAmmount=sc.nextInt();

       ac.deposit(depositAmmount);

            System.out.print("Enter Withdraw ammount ");

        int withDraw=sc.nextInt();

       ac.deposit(withDraw);
       ac.getBalance();
    }
}
```

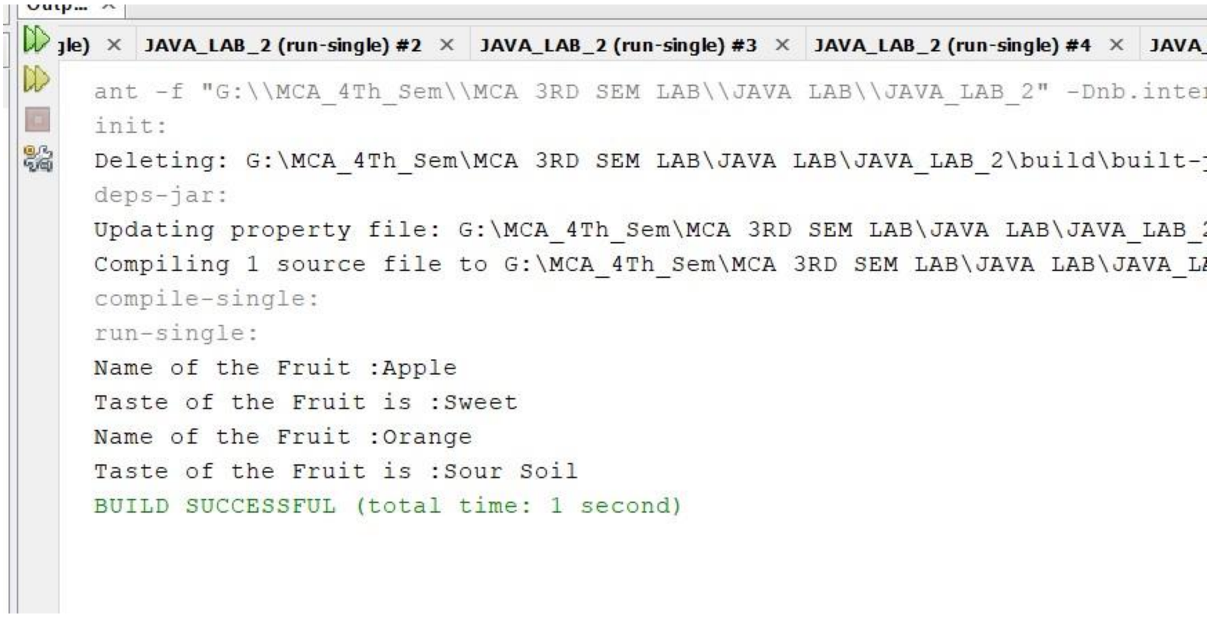


```
Deleting: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-jar.properties
deps-jar:
Updating property file: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-jar.properties
Compiling 1 source file to G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\classes
compile-single:
run-single:


****************************************************************

Account Holder Name : XYZ AB
Acoount Number is : 100359865
Account Type is : Saving Account
Current Ballance is 1000
Enter Deposit ammount 50000000
50000000 Amount Deposited to Your Account
Enter Withdraw ammount 5000000
5000000 Amount Deposited to Your Account


****************************************************************

Account Holder Name : XYZ AB
Acoount Number is : 100359865
Account Type is : Saving Account
Current Ballance is 55001000
BUILD SUCCESSFUL (total time: 24 seconds)
```

P-2 Create a base class Fruit which has name ,taste and size as its attributes. A method called eat() is created which describes the name of the fruit and its taste. Inherit the same in 2 other class Apple and Orange and override the eat() method to represent each fruit taste.

```
package java_lab_2.QUES_2;
class Fruit{
    public String taste,name;
    //public int size;
    public void eat(String name, String taste)
   {
   System.out.println("Name of the Fruit :"+ name);
   System.out.println("Taste of the Fruit is :"+ taste);
   }
}
class Apple extends Fruit
```

```
{


   @Override
   public void eat(String name, String taste)
   {
      super.eat(name, taste);
  }
}
class Orange extends Fruit
{
     @Override
  public void eat(String name, String taste)
   {


       super.eat(name, taste);
   }
}
public class Main {
    public static void main(String args[])
    {
        Fruit obj1 = new Apple();
        obj1.eat("Apple","Sweet");


        Fruit obj2 = new Orange();
        obj2.eat("Orange","Sour Soil");
    }


}
```

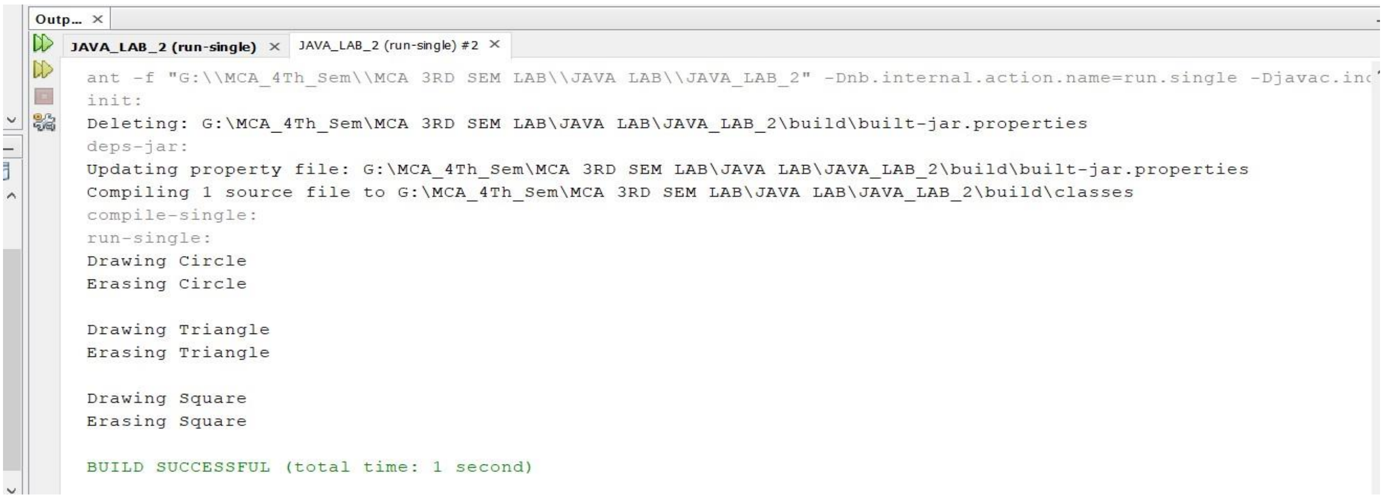

Outp... ×
gle) ×
JAVA_LAB_2 (run-single) #2 ×
JAVA_LAB_2 (run-single) #3 ×
JAVA_LAB_2 (run-single) #4 ×
JAVA_

```
ant -f "G:\\MCA_4Th_Sem\\MCA 3RD SEM LAB\\JAVA LAB\\JAVA_LAB_2" -Dnb.inte
init:
Deleting: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-
deps-jar:
Updating property file: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_
Compiling 1 source file to G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_L
compile-single:
run-single:
Name of the Fruit :Apple
Taste of the Fruit is :Sweet
Name of the Fruit :Orange
Taste of the Fruit is :Sour Soil
BUILD SUCCESSFUL (total time: 1 second)
```

p-3 Write a program to create a class named shape. It should contain 2 methods- draw() and erase() which should print “Drawing Shape” and “Erasing Shape” respectively.

> **For this class we have three sub classes- Circle, Triangle and Square and each class override the parent class functions- draw () and erase ().**
> **The draw() method should print “Drawing Circle”, “Drawing Triangle”, “Drawing Square” respectively.**
> **The erase() method should print “Erasing Circle”, “Erasing Triangle”, “Erasing Square” respectively.**
> **Create objects of Circle, Triangle and Square in the following way and observe the polymorphic nature of the class by calling draw() and erase() method using each object.**
> **Shape c=new Circle();**
> **Shape t=new Triangle();**
> **Shape s=new Square();**

```
package java_lab_2.QUES_3.Main;


class Shape{
    public void draw()
    {
        System.out.println("Drawing Shape");
    }
    public void erase()
    {
        System.out.println("Erasing Shape");
    }
}
class Circle extends Shape{
     public void draw()
    {
        System.out.println("Drawing Circle");
    }
    public void erase()
    {
        System.out.println("Erasing Circle\n");
    }
}
class Triangle extends Shape{
   public void draw()
    {
        System.out.println("Drawing Triangle");
    }
    public void erase()
    {
        System.out.println("Erasing Triangle\n");
    }
}
class Square extends Shape{
     public void draw()
    {
        System.out.println("Drawing Square");
    }
    public void erase()
    {
        System.out.println("Erasing Square\n");
    }
}
public class main {
    public static void main(String args[])
    {


        Circle c=new Circle();
        c.draw();
        c.erase();
        Triangle t=new Triangle();
        t.draw();
        t.erase();
        Square s=new Square();
        s.draw();
        s.erase();
    }
}
```



```
ant -f "G:\\MCA_4Th_Sem\\MCA 3RD SEM LAB\\JAVA LAB\\JAVA_LAB_2" -Dnb.internal.action.name=run.single -Djavac.inc
init:
Deleting: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-jar.properties
deps-jar:
Updating property file: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-jar.properties
Compiling 1 source file to G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\classes
compile-single:
run-single:
Drawing Circle
Erasing Circle

Drawing Triangle
Erasing Triangle

Drawing Square
Erasing Square

BUILD SUCCESSFUL (total time: 1 second)
```

P-4 Write a program to maintain data about Books in a library of a college. Create a class Book which has following members:

private int bookNo
private String title
private String publication
private String author
private float price

Getter and setter methods for all members.
Create a class Computer derived from Book having following members
private String type (It could be Networking, DataStructure, DBMS)

Setter and getter method for type instance variable

Create a class Mathematics derived from Book having following members
private String type (It could be Algebra, Geometry)

Setter and getter method for type instance variable

Create a class TestBook having main method. Create an object of Computer. Scan data from user, set and print details of Computer book.
Create an object of Mathematics class. Scan data from user, set and print details of Mathematics book.

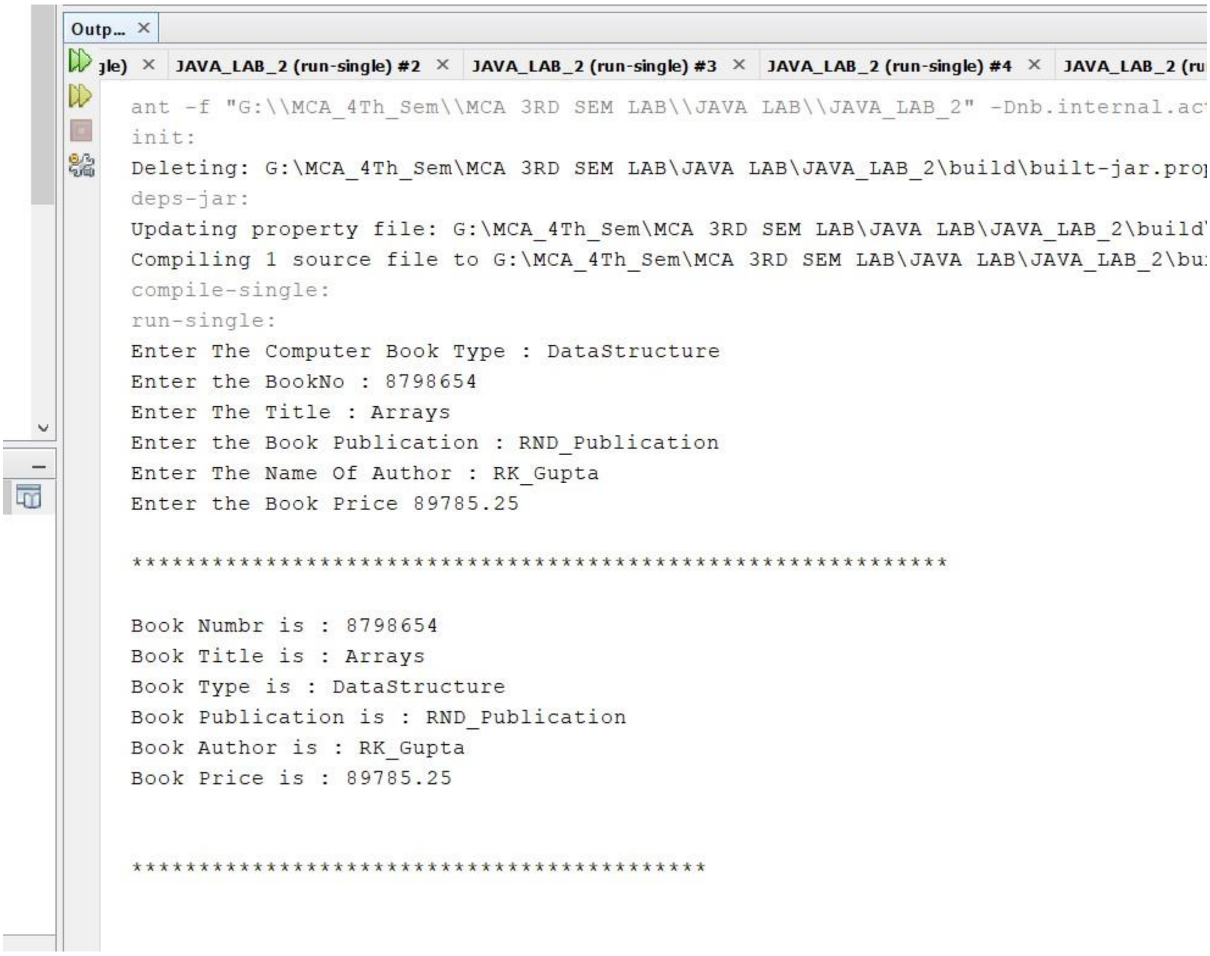
```
package java_lab_2.QUES_4;


import java.util.Scanner;


class Book{
```

```
private int bookNo;
private String title;
private String publication;
private String author;
private float price;

public void setData(int bookNo,String title,String publication,String author,float price)
{
   this.author=author;
   this.bookNo=bookNo;
   this.title=title;
   this.publication=publication;
   this.price=price;
}
public int getBookNo()
{
   return bookNo;
}
public String getTitle()
{
   return title;
}
public String getPublication()
{
   return publication;
}
public String getAuthor()
{
   return author;
}
```

```
    public float getPrice()
    {
        return price;
    }
}

class Computer extends Book
{
    private String type;
    public void setComputerBookData(String type)
    {
        this.type=type;
    }
    public String getComputerBookData()
    {
        return type;
    }

}

class Mathematics extends Book
{
    private String type;
    public void setMathsBookData(String type)
    {
        this.type=type;
    }
    public String getMathsBookData()
    {
        return type;
    }



}
public class TestBook {
    public static void main(String args[])
    {
        Computer c=new Computer();

        Scanner sc=new Scanner(System.in);
        System.out.print("Enter The Computer Book Type : ");
        String compType=sc.next();

        c.setComputerBookData(compType);



       String publication;
       String author;
      float price;
        System.out.print("Enter the BookNo : ");
        int bookNo=sc.nextInt();
        System.out.print("Enter The Title : ");
        String title=sc.next();
        System.out.print("Enter the Book Publication : ");
        publication=sc.next();
        System.out.print("Enter The Name Of Author : ");
        author=sc.next();
        System.out.print("Enter the Book Price ");
        price=sc.nextFloat();
        c.setData(bookNo, title, publication, author, price);
```

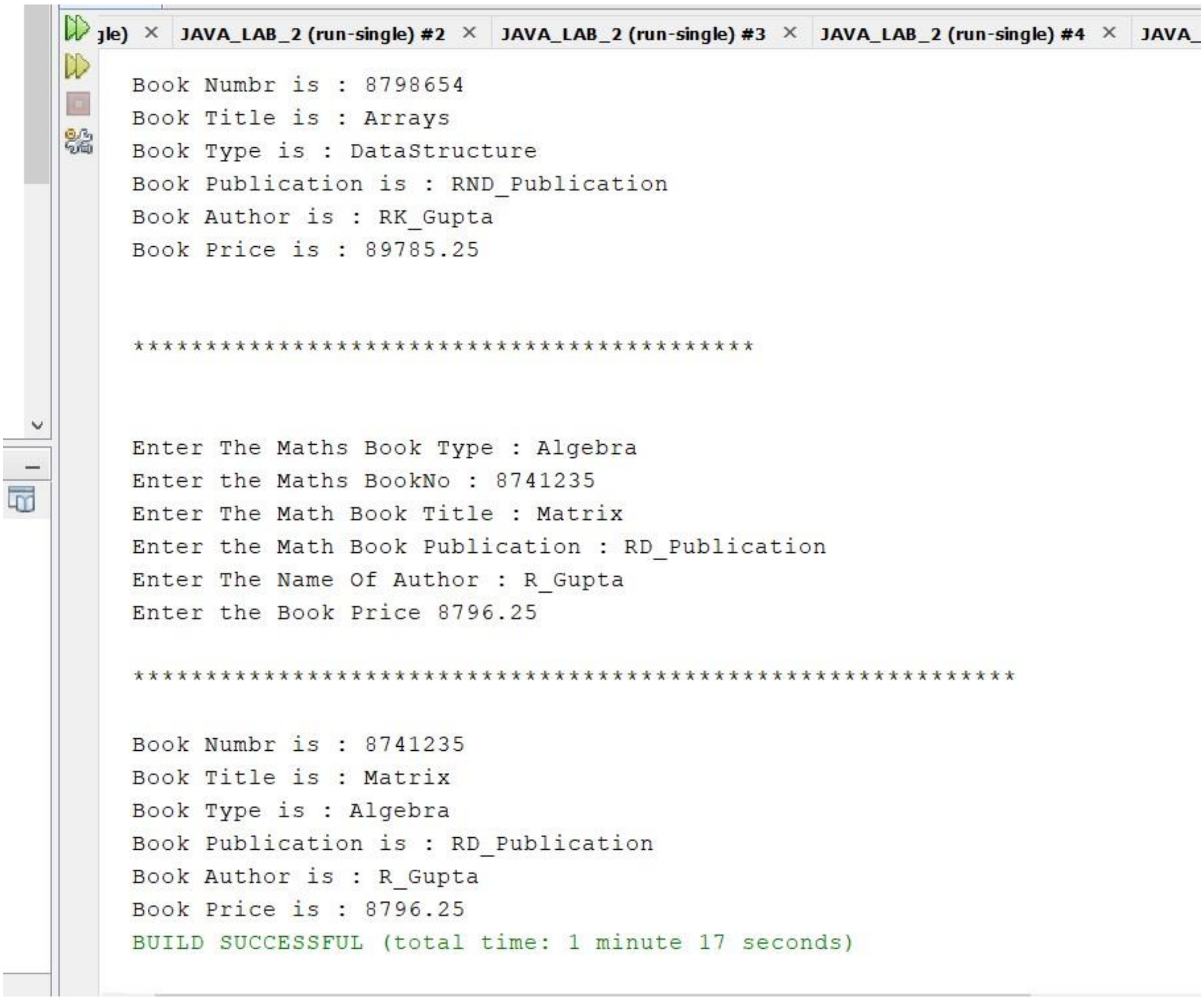
```
System.out.println("\n*********************************************************
****\n"+"\nBook Numbr is : "+c.getBookNo()+"\nBook Title is : "+c.getTitle()+"\nBook
Type is : "+c.getComputerBookData()+"\nBook Publication is :
"+c.getPublication()+"\nBook Author is : "+c.getAuthor()+"\nBook Price is : "+c.getPrice());


    System.out.println("\n\n*******************************************\n\n");
    Mathematics m =new Mathematics();


     System.out.print("Enter The Maths Book Type : ");
    String MathBookType=sc.next();
    m.setMathsBookData(MathBookType);


    System.out.print("Enter the Maths BookNo : ");
     bookNo=sc.nextInt();
    System.out.print("Enter The Math Book Title : ");
    title=sc.next();
    System.out.print("Enter the Math Book Publication : ");
    publication=sc.next();
    System.out.print("Enter The Name Of Author : ");
    author=sc.next();
    System.out.print("Enter the Book Price ");
    price=sc.nextFloat();
    m.setData(bookNo, title, publication, author, price);



System.out.println("\n*********************************************************
****\n"+"\nBook Numbr is : "+m.getBookNo()+"\nBook Title is : "+m.getTitle()+"\nBook
Type is : "+m.getMathsBookData()+"\nBook Publication is : "+m.getPublication()+"\nBook
Author is : "+m.getAuthor()+"\nBook Price is : "+m.getPrice());


  }
```

}



Outp...
gle)
JAVA_LAB_2 (run-single) #2
JAVA_LAB_2 (run-single) #3
JAVA_LAB_2 (run-single) #4
JAVA_LAB_2 (ru

```
ant -f "G:\\MCA_4Th_Sem\\MCA 3RD SEM LAB\\JAVA LAB\\JAVA_LAB_2" -Dnb.internal.ac
init:
Deleting: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build\built-jar.pro
deps-jar:
Updating property file: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\build
Compiling 1 source file to G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\bu
compile-single:
run-single:
Enter The Computer Book Type : DataStructure
Enter the BookNo : 8798654
Enter The Title : Arrays
Enter the Book Publication : RND_Publication
Enter The Name Of Author : RK_Gupta
Enter the Book Price 89785.25

*************************************************************

Book Numbr is : 8798654
Book Title is : Arrays
Book Type is : DataStructure
Book Publication is : RND_Publication
Book Author is : RK_Gupta
Book Price is : 89785.25


*******************************************
```


gle) ×
JAVA_LAB_2 (run-single) #2 ×
JAVA_LAB_2 (run-single) #3 ×
JAVA_LAB_2 (run-single) #4 ×
JAVA_

```
Book Numbr is : 8798654
Book Title is : Arrays
Book Type is : DataStructure
Book Publication is : RND_Publication
Book Author is : RK_Gupta
Book Price is : 89785.25


*******************************************



Enter The Maths Book Type : Algebra
Enter the Maths BookNo : 8741235
Enter The Math Book Title : Matrix
Enter the Math Book Publication : RD_Publication
Enter The Name Of Author : R_Gupta
Enter the Book Price 8796.25

*************************************************************


Book Numbr is : 8741235
Book Title is : Matrix
Book Type is : Algebra
Book Publication is : RD_Publication
Book Author is : R_Gupta
Book Price is : 8796.25
BUILD SUCCESSFUL (total time: 1 minute 17 seconds)
```

P-5 Create a class Point which has following members:

private int x;
private int y;

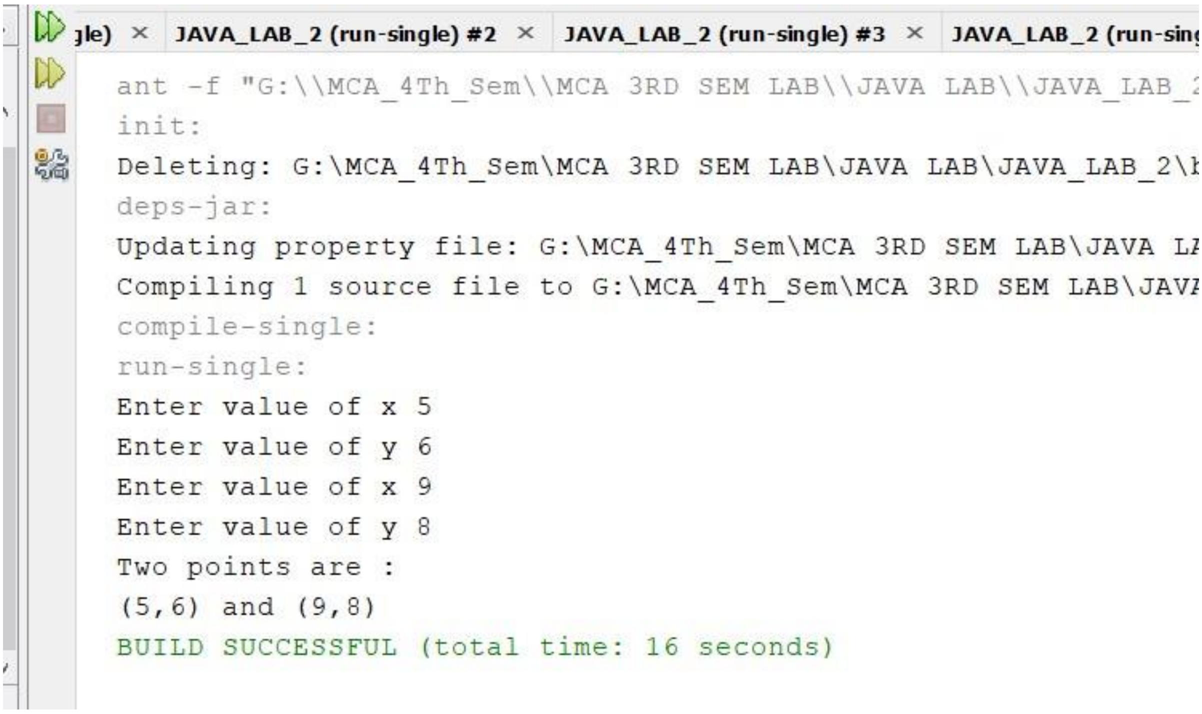
Getter and setter methods for both instance variables

Create a class Line derived from Point having following members:
private Point point1;
private Point point2;

Getter and setter methods for both instance variables

Create a class TestLine having main method. Scan 2 point details for line from user and print them.

```
package java_lab_2.QUES_5;

import java.util.Scanner;

 class Point
   {
      private int x;
      private int y;
      public int getX()
      {
         return(x);
      }
      public int getY()
      {
          return(y);
      }

      public void setXY(int x,int y)
      {
          this.x=x;
          this.y=y;
      }
  }

class Line extends Point
{
    private Point point1;
    private Point point2;

    public Point getPoint1()
    {
        return(point1);
    }
    public Point getPoint2()
    {
        return(point2);
    }
    public void setter(Point p1,Point p2)
    {
        this.point1=p1;
        this.point2=p2;
    }

}
public class TestLine {
     public static void main(String args[])
    {
        int x1,y1,x2,y2;
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter value of x");
        x1= sc.nextInt();
        System.out.print("Enter value of y");
        y1 = sc.nextInt();



        Line obj1 = new Line();
        obj1.setXY(x1, y1);



        System.out.print("Enter value of x");
        x2= sc.nextInt();
        System.out.print("Enter value of y");
        y2 = sc.nextInt();

        Line obj2 = new Line();
        obj2.setXY(x2, y2);

        Line  obj = new Line();
        // Create object for Line Class
          obj.setter(obj1, obj2);
        System.out.println("Two points are :");
        System.out.println("("+
obj.getPoint1().getX()+","+obj.getPoint1().getY()+")"+" and "+
       "("+ obj.getPoint2().getX()+","+obj.getPoint2().getY()+")" );
    }
}
```



```
ant -f "G:\\MCA_4Th_Sem\\MCA 3RD SEM LAB\\JAVA LAB\\JAVA_LAB_
init:
Deleting: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LAB\JAVA_LAB_2\
deps-jar:
Updating property file: G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA LA
Compiling 1 source file to G:\MCA_4Th_Sem\MCA 3RD SEM LAB\JAVA
compile-single:
run-single:
Enter value of x 5
Enter value of y 6
Enter value of x 9
Enter value of y 8
Two points are :
(5,6) and (9,8)
BUILD SUCCESSFUL (total time: 16 seconds)
```