Object Oriented Programming

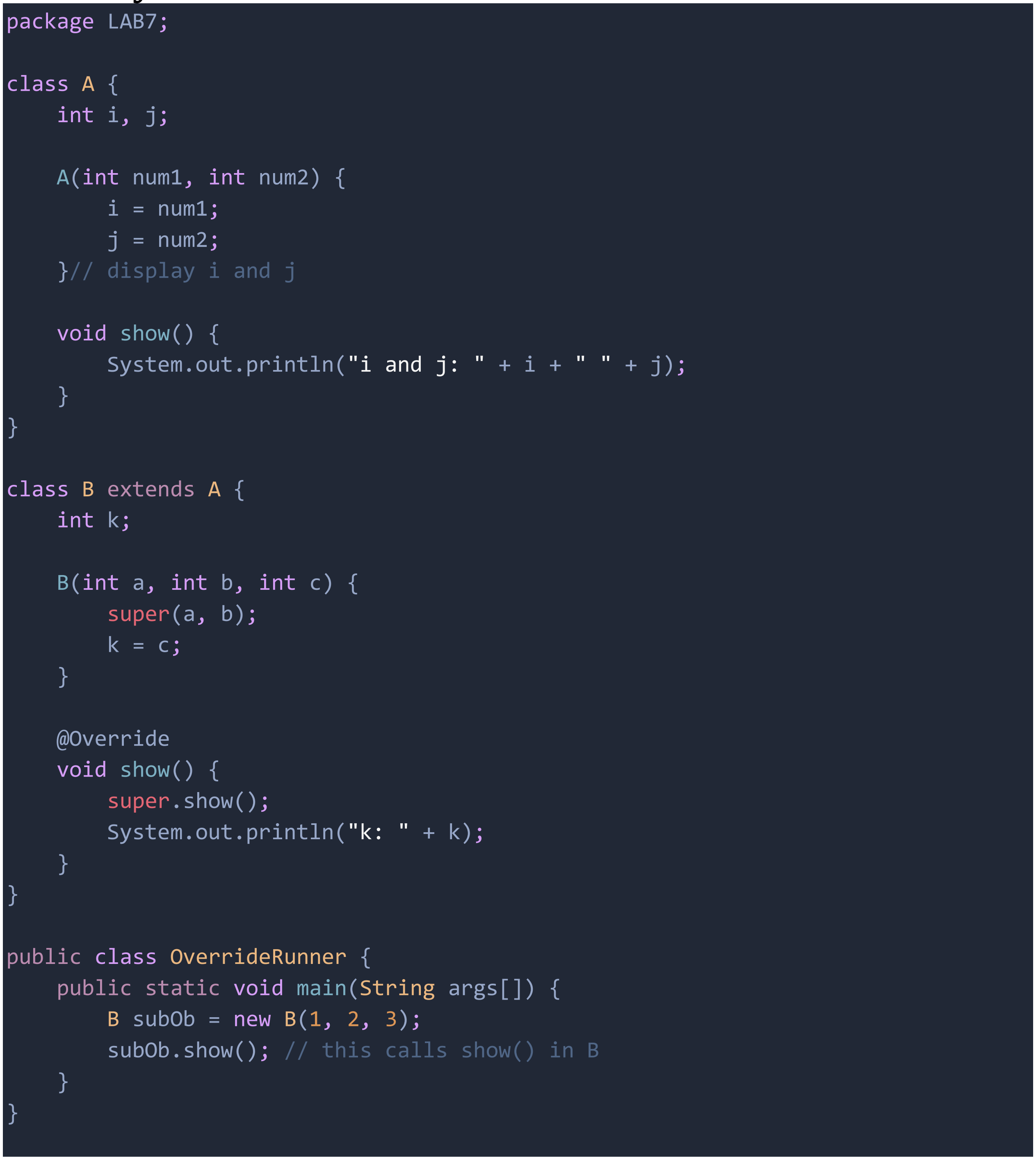
Lab Assignment 7

SUBMITTED BY:

Hasaan Ahmad SP22-BSE-017

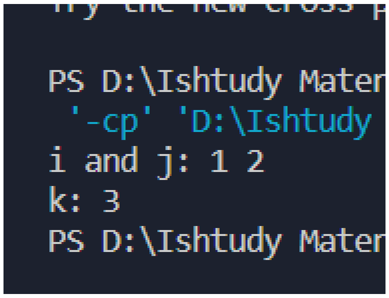
SUBMITTED TO: Sir Muzaffar Iqbal

Activity 1:



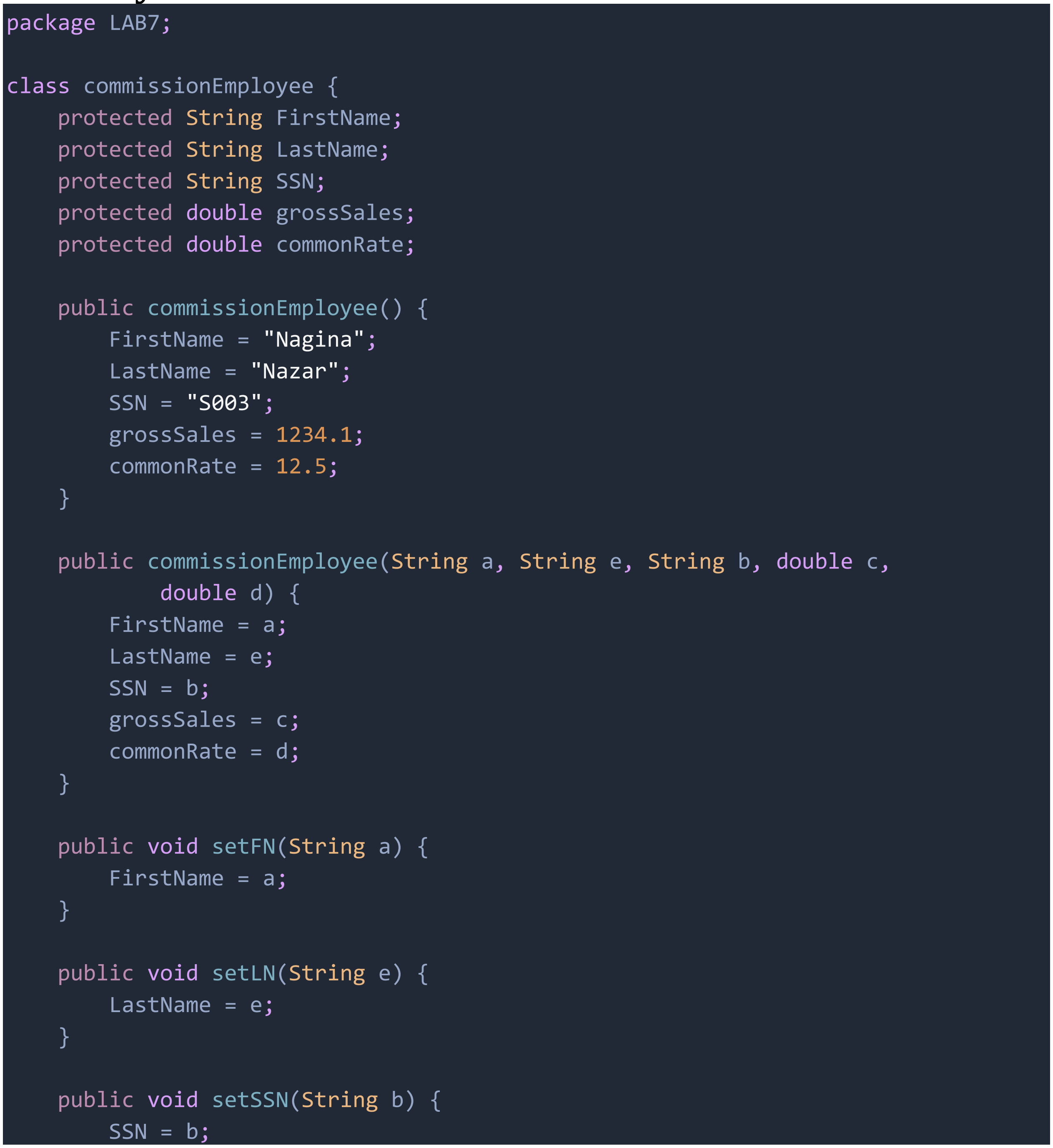
```java
package LAB7;

class A {
    int i, j;

    A(int num1, int num2) {
        i = num1;
        j = num2;
    }// display i and j

    void show() {
        System.out.println("i and j: " + i + " " + j);
    }
}

class B extends A {
    int k;

    B(int a, int b, int c) {
        super(a, b);
        k = c;
    }

    @Override
    void show() {
        super.show();
        System.out.println("k: " + k);
    }
}

public class OverrideRunner {
    public static void main(String args[]) {
        B subOb = new B(1, 2, 3);
        subOb.show(); // this calls show() in B
    }
}
```

Output:

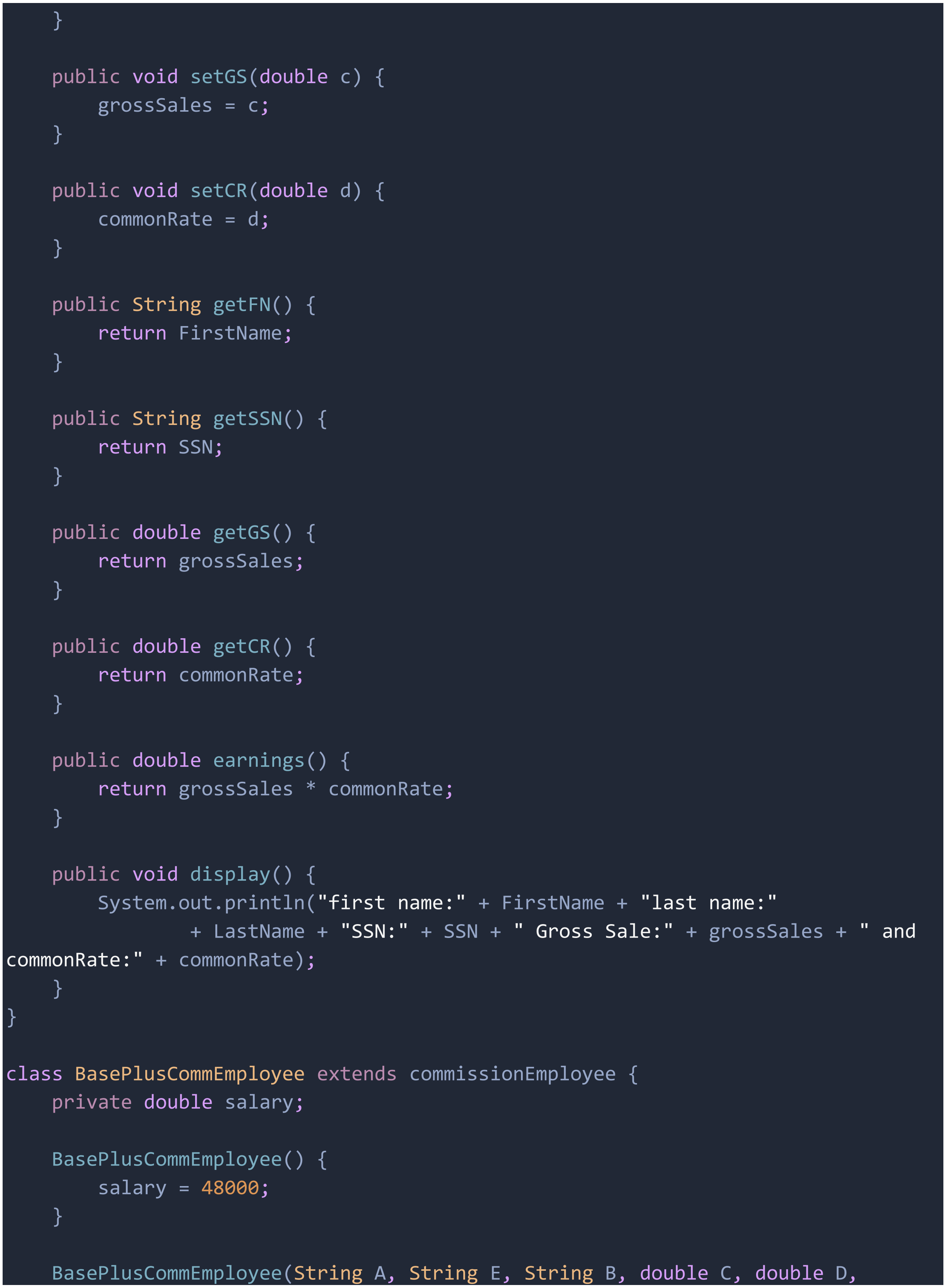


```
Try the new cross p

PS D:\Ishtudy Mater
 '-cp' 'D:\Ishtudy
i and j: 1 2
k: 3
PS D:\Ishtudy Mater
```

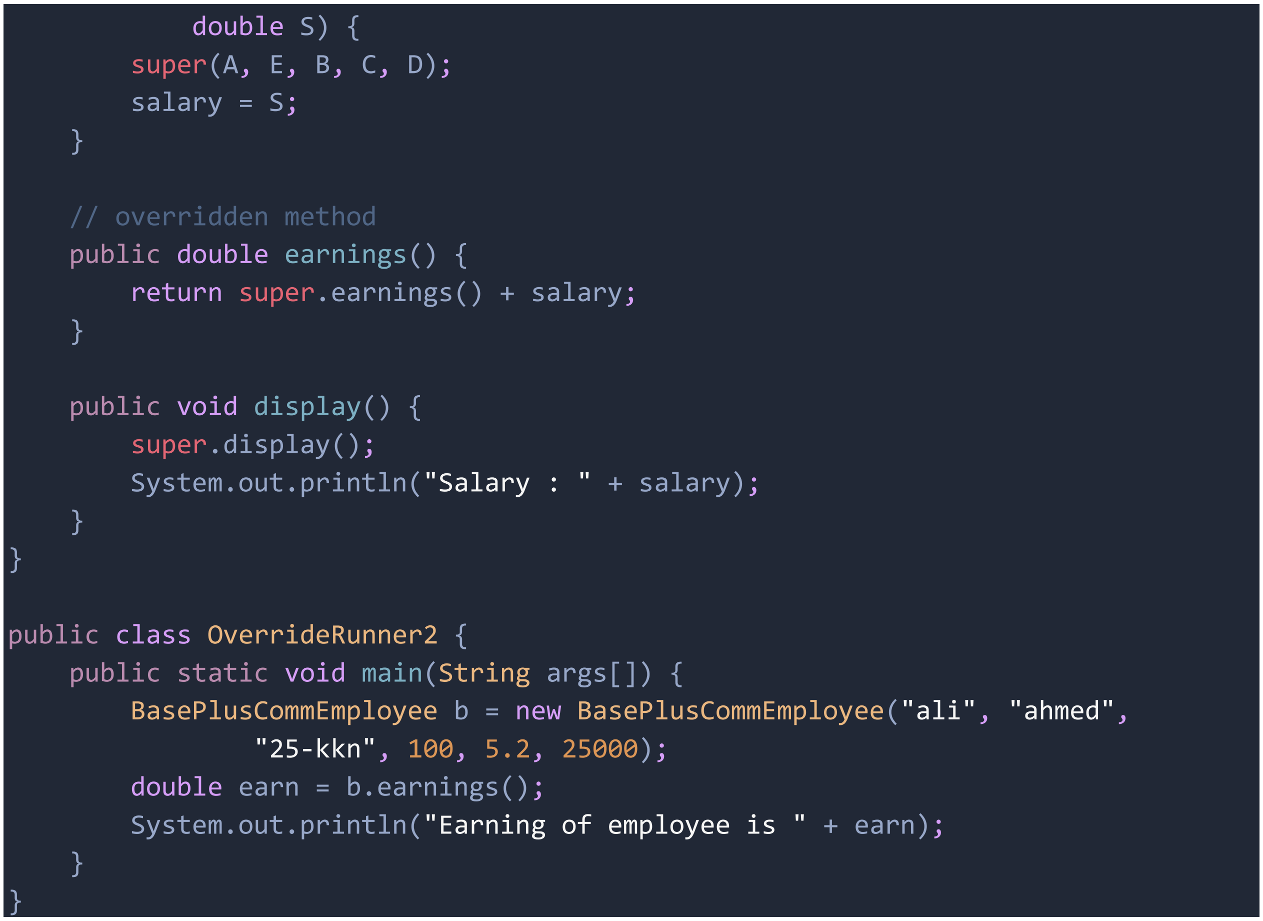
Activity 2:



```
package LAB7;

class commissionEmployee {
    protected String FirstName;
    protected String LastName;
    protected String SSN;
    protected double grossSales;
    protected double commonRate;

    public commissionEmployee() {
        FirstName = "Nagina";
        LastName = "Nazar";
        SSN = "S003";
        grossSales = 1234.1;
        commonRate = 12.5;
    }

    public commissionEmployee(String a, String e, String b, double c,
            double d) {
        FirstName = a;
        LastName = e;
        SSN = b;
        grossSales = c;
        commonRate = d;
    }

    public void setFN(String a) {
        FirstName = a;
    }

    public void setLN(String e) {
        LastName = e;
    }

    public void setSSN(String b) {
        SSN = b;
```

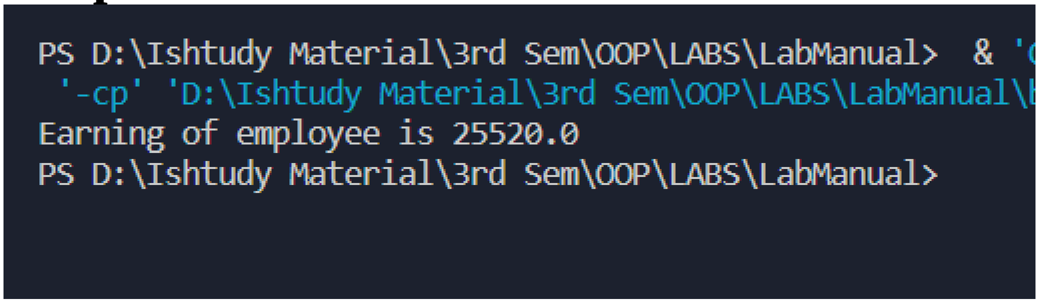


```
    }

    public void setGS(double c) {
        grossSales = c;
    }

    public void setCR(double d) {
        commonRate = d;
    }

    public String getFN() {
        return FirstName;
    }

    public String getSSN() {
        return SSN;
    }

    public double getGS() {
        return grossSales;
    }

    public double getCR() {
        return commonRate;
    }

    public double earnings() {
        return grossSales * commonRate;
    }

    public void display() {
        System.out.println("first name:" + FirstName + "last name:"
                + LastName + "SSN:" + SSN + " Gross Sale:" + grossSales + " and
commonRate:" + commonRate);
    }
}

class BasePlusCommEmployee extends commissionEmployee {
    private double salary;

    BasePlusCommEmployee() {
        salary = 48000;
    }

    BasePlusCommEmployee(String A, String E, String B, double C, double D,
```



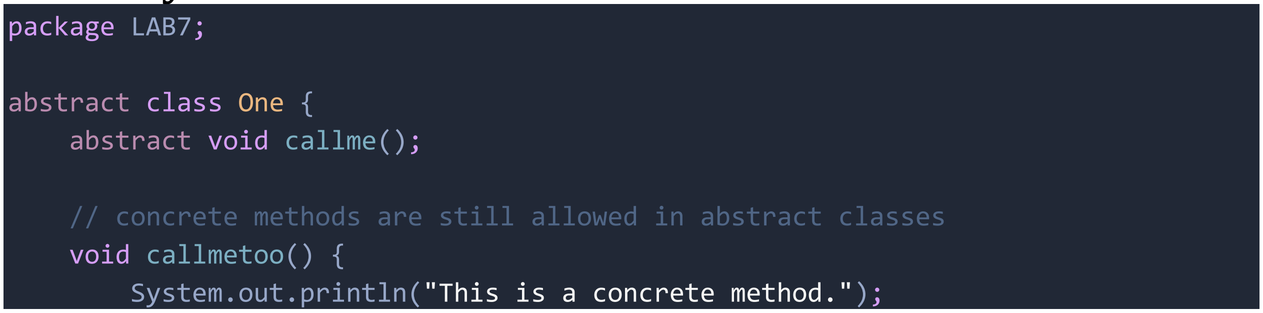
```java
            double S) {
        super(A, E, B, C, D);
        salary = S;
    }

    // overridden method
    public double earnings() {
        return super.earnings() + salary;
    }

    public void display() {
        super.display();
        System.out.println("Salary : " + salary);
    }
}

public class OverrideRunner2 {
    public static void main(String args[]) {
        BasePlusCommEmployee b = new BasePlusCommEmployee("ali", "ahmed",
                "25-kkn", 100, 5.2, 25000);
        double earn = b.earnings();
        System.out.println("Earning of employee is " + earn);
    }
}
```

Output:

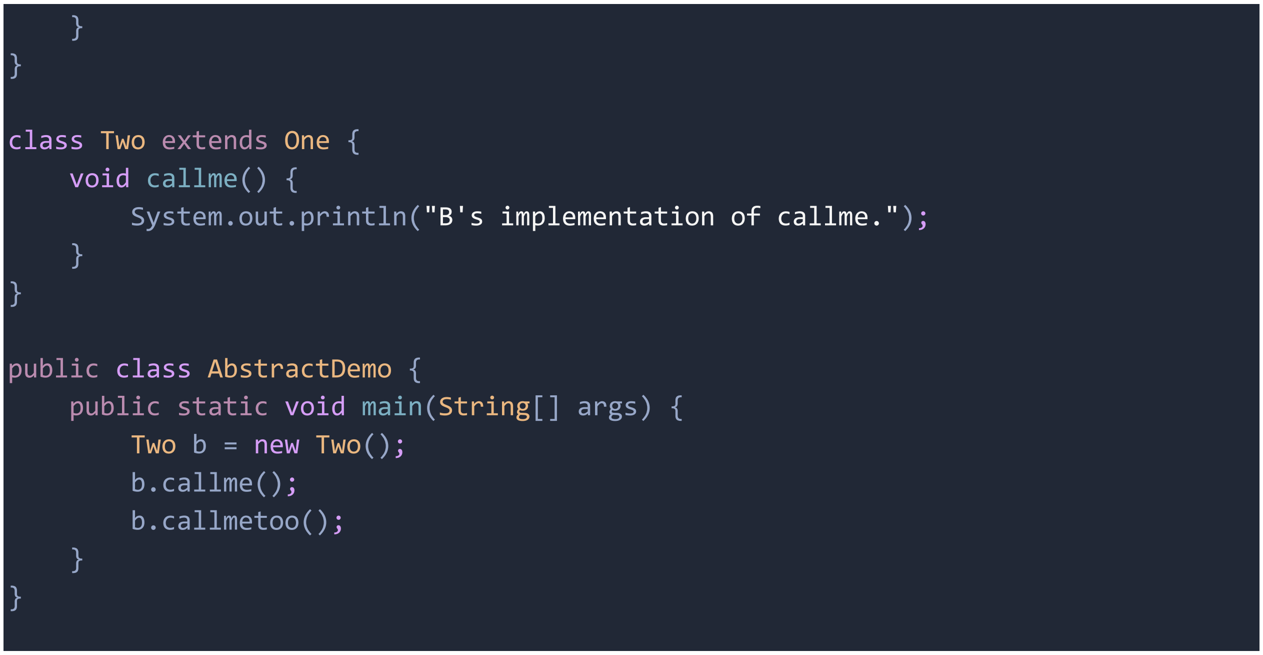


```
PS D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual>  & '
 '-cp' 'D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual\
Earning of employee is 25520.0
PS D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual>
```

Activity 3:

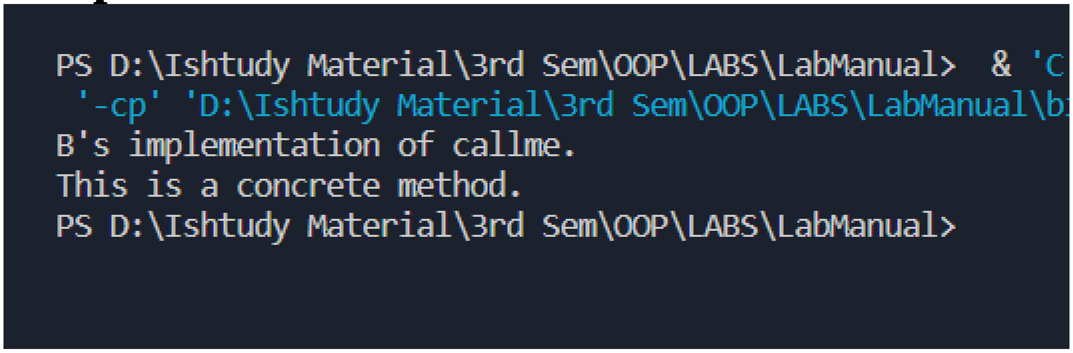


```java
package LAB7;

abstract class One {
    abstract void callme();

    // concrete methods are still allowed in abstract classes
    void callmetoo() {
        System.out.println("This is a concrete method.");
```



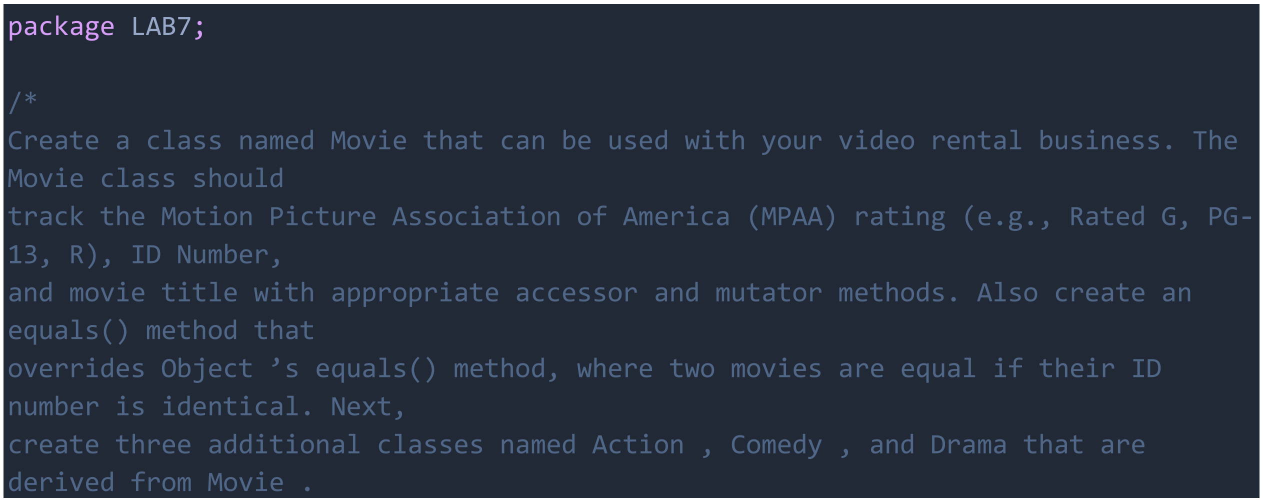
```
    }
}

class Two extends One {
    void callme() {
        System.out.println("B's implementation of callme.");
    }
}

public class AbstractDemo {
    public static void main(String[] args) {
        Two b = new Two();
        b.callme();
        b.callmetoo();
    }
}
```

Output:



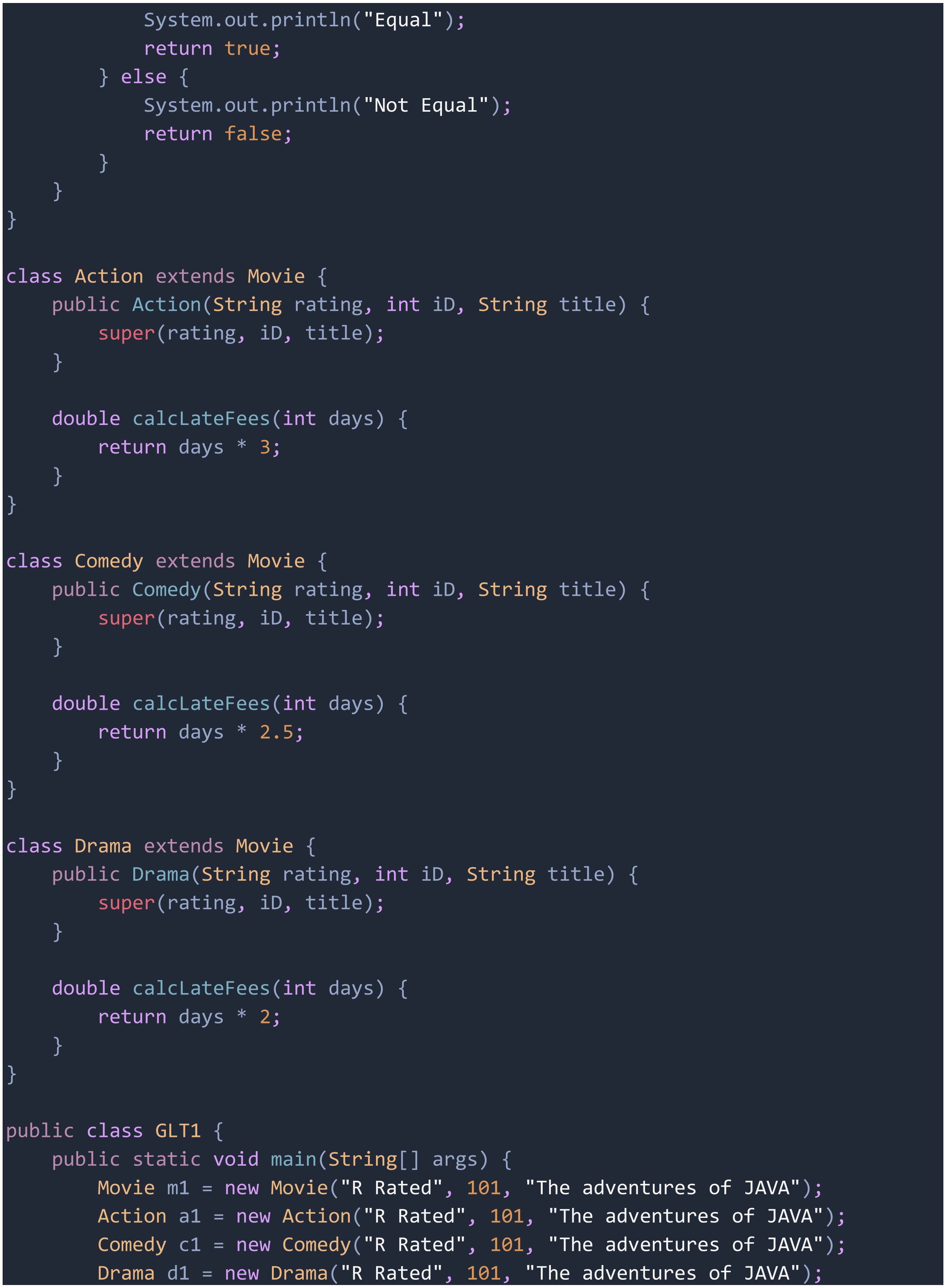
```
PS D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual>  & 'C
 '-cp' 'D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual\b
B's implementation of callme.
This is a concrete method.
PS D:\Ishtudy Material\3rd Sem\OOP\LABS\LabManual>
```

Graded Lab Task 1:



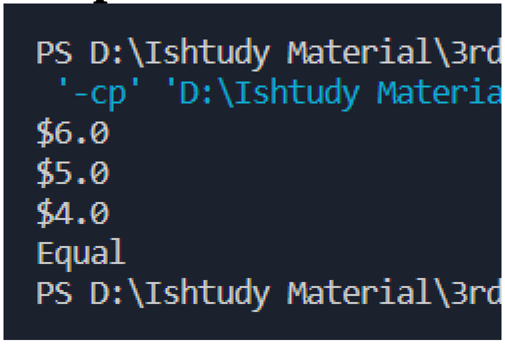
```
package LAB7;

/*
Create a class named Movie that can be used with your video rental business. The
Movie class should
track the Motion Picture Association of America (MPAA) rating (e.g., Rated G, PG-
13, R), ID Number,
and movie title with appropriate accessor and mutator methods. Also create an
equals() method that
overrides Object ’s equals() method, where two movies are equal if their ID
number is identical. Next,
create three additional classes named Action , Comedy , and Drama that are
derived from Movie .
```

```
Finally, create an overridden method named calcLateFees that takes as input the
number of days a movie
is late and returns the late fee for that movie. The default late fee is $2/day.
Action movies have a late fee
of $3/day, comedies are $2.50/day, and dramas are $2/day. Test your classes from
a main method
*/

class Movie {
    String rating;
    int ID;
    String Title;

    public Movie(String rating, int iD, String title) {
        this.rating = rating;
        ID = iD;
        Title = title;
    }

    public String getRating() {
        return rating;
    }

    public void setRating(String rating) {
        this.rating = rating;
    }

    public int getID() {
        return ID;
    }

    public void setID(int iD) {
        ID = iD;
    }

    public String getTitle() {
        return Title;
    }

    public void setTitle(String title) {
        Title = title;
    }

    Boolean equals(Movie m1) {
        if (this.ID == m1.ID) {
```



```
            System.out.println("Equal");
            return true;
        } else {
            System.out.println("Not Equal");
            return false;
        }
    }
}

class Action extends Movie {
    public Action(String rating, int iD, String title) {
        super(rating, iD, title);
    }

    double calcLateFees(int days) {
        return days * 3;
    }
}

class Comedy extends Movie {
    public Comedy(String rating, int iD, String title) {
        super(rating, iD, title);
    }

    double calcLateFees(int days) {
        return days * 2.5;
    }
}

class Drama extends Movie {
    public Drama(String rating, int iD, String title) {
        super(rating, iD, title);
    }

    double calcLateFees(int days) {
        return days * 2;
    }
}

public class GLT1 {
    public static void main(String[] args) {
        Movie m1 = new Movie("R Rated", 101, "The adventures of JAVA");
        Action a1 = new Action("R Rated", 101, "The adventures of JAVA");
        Comedy c1 = new Comedy("R Rated", 101, "The adventures of JAVA");
        Drama d1 = new Drama("R Rated", 101, "The adventures of JAVA");
        System.out.println("$" + a1.calcLateFees(2));
        System.out.println("$" + c1.calcLateFees(2));
        System.out.println("$" + d1.calcLateFees(2));
        m1.equals(a1);
    }

}
```

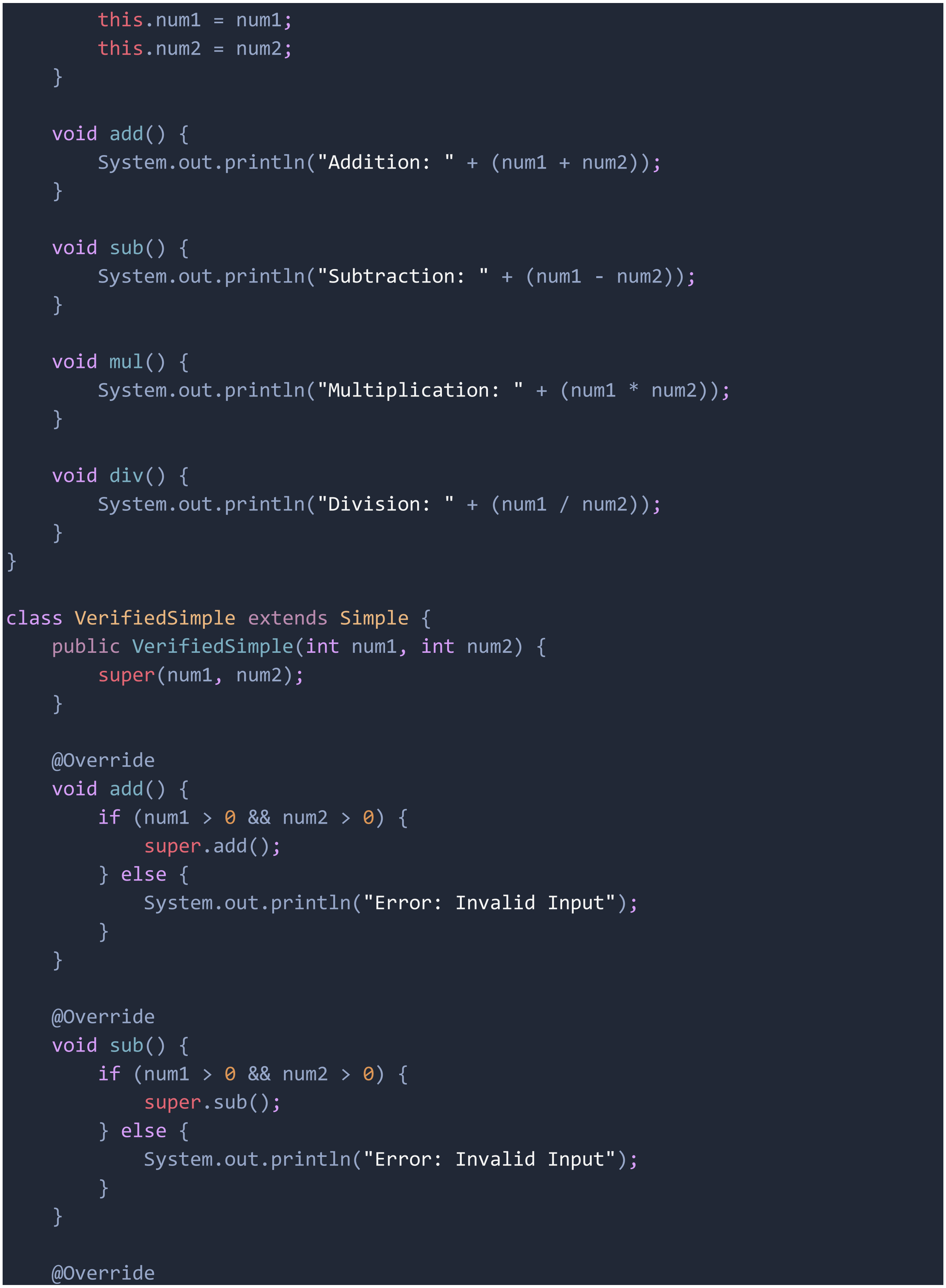
Output:



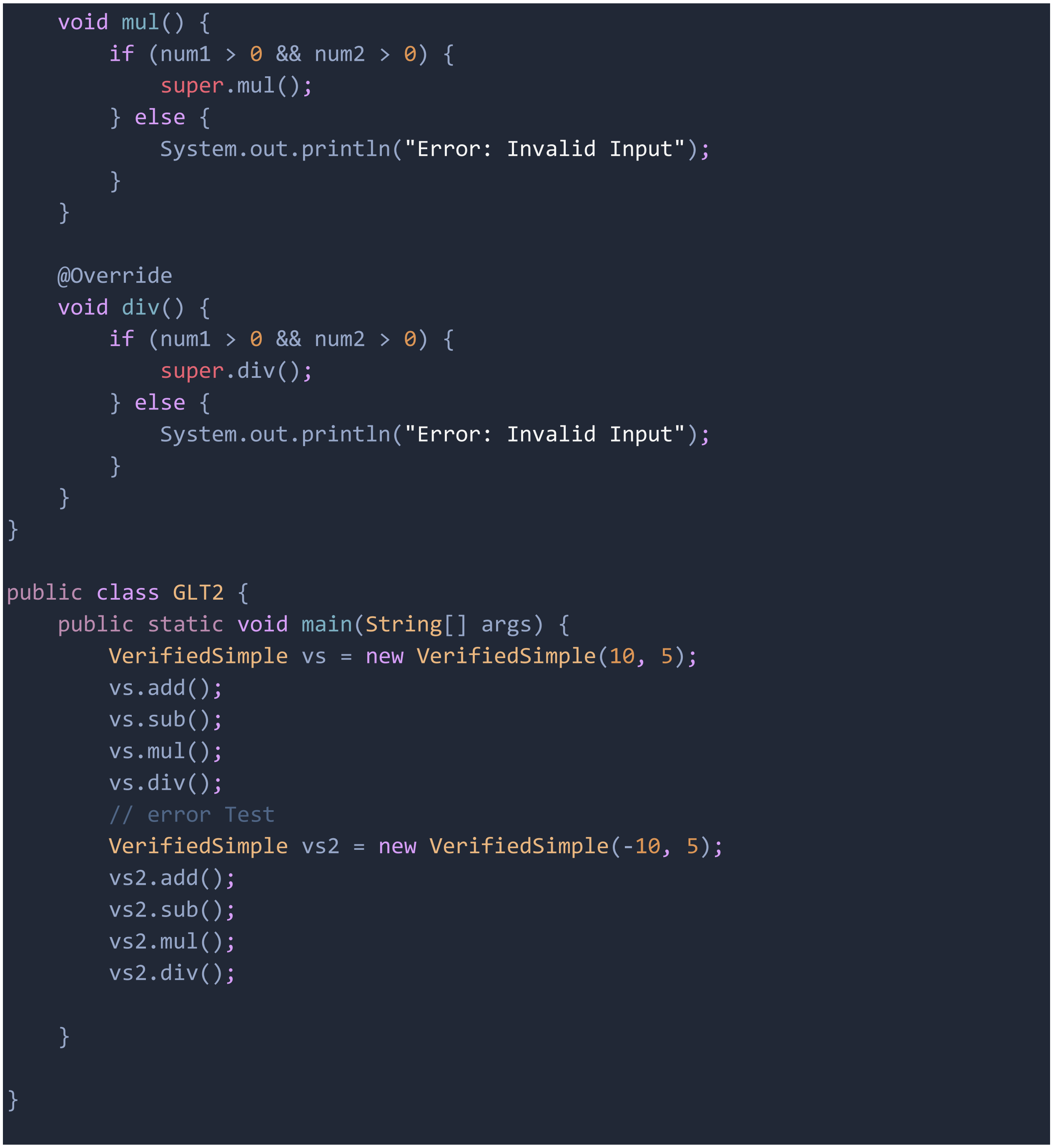
```
PS D:\Ishtudy Material\3rd
 '-cp' 'D:\Ishtudy Materia
$6.0
$5.0
$4.0
Equal
PS D:\Ishtudy Material\3rd
```

Graded Lab Task 2:

```
package LAB7;

/*
Write a program that declares two classes. The parent class is called Simple that
has two data members
num1 and num2 to store two numbers. It also has four member functions.
The add() function adds two numbers and displays the result. The sub() function
subtracts two numbers
and displays the result.
The mul() function multiplies two numbers and displays the result. The div()
function divides two numbers
and displays the result.
The child class is called VerifiedSimple that overrides all four functions. Each
function in the child class
checks the value of data members. It calls the corresponding member function in
the parent class if the
values are greater than 0. Otherwise it displays error message.
*/
class Simple {
    int num1;
    int num2;

    public Simple(int num1, int num2) {
```

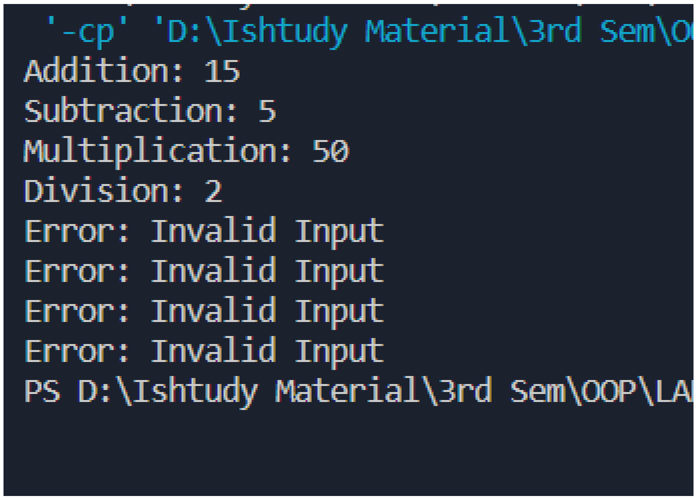


```java
        this.num1 = num1;
        this.num2 = num2;
    }

    void add() {
        System.out.println("Addition: " + (num1 + num2));
    }

    void sub() {
        System.out.println("Subtraction: " + (num1 - num2));
    }

    void mul() {
        System.out.println("Multiplication: " + (num1 * num2));
    }

    void div() {
        System.out.println("Division: " + (num1 / num2));
    }
}

class VerifiedSimple extends Simple {
    public VerifiedSimple(int num1, int num2) {
        super(num1, num2);
    }

    @Override
    void add() {
        if (num1 > 0 && num2 > 0) {
            super.add();
        } else {
            System.out.println("Error: Invalid Input");
        }
    }

    @Override
    void sub() {
        if (num1 > 0 && num2 > 0) {
            super.sub();
        } else {
            System.out.println("Error: Invalid Input");
        }
    }

    @Override
```



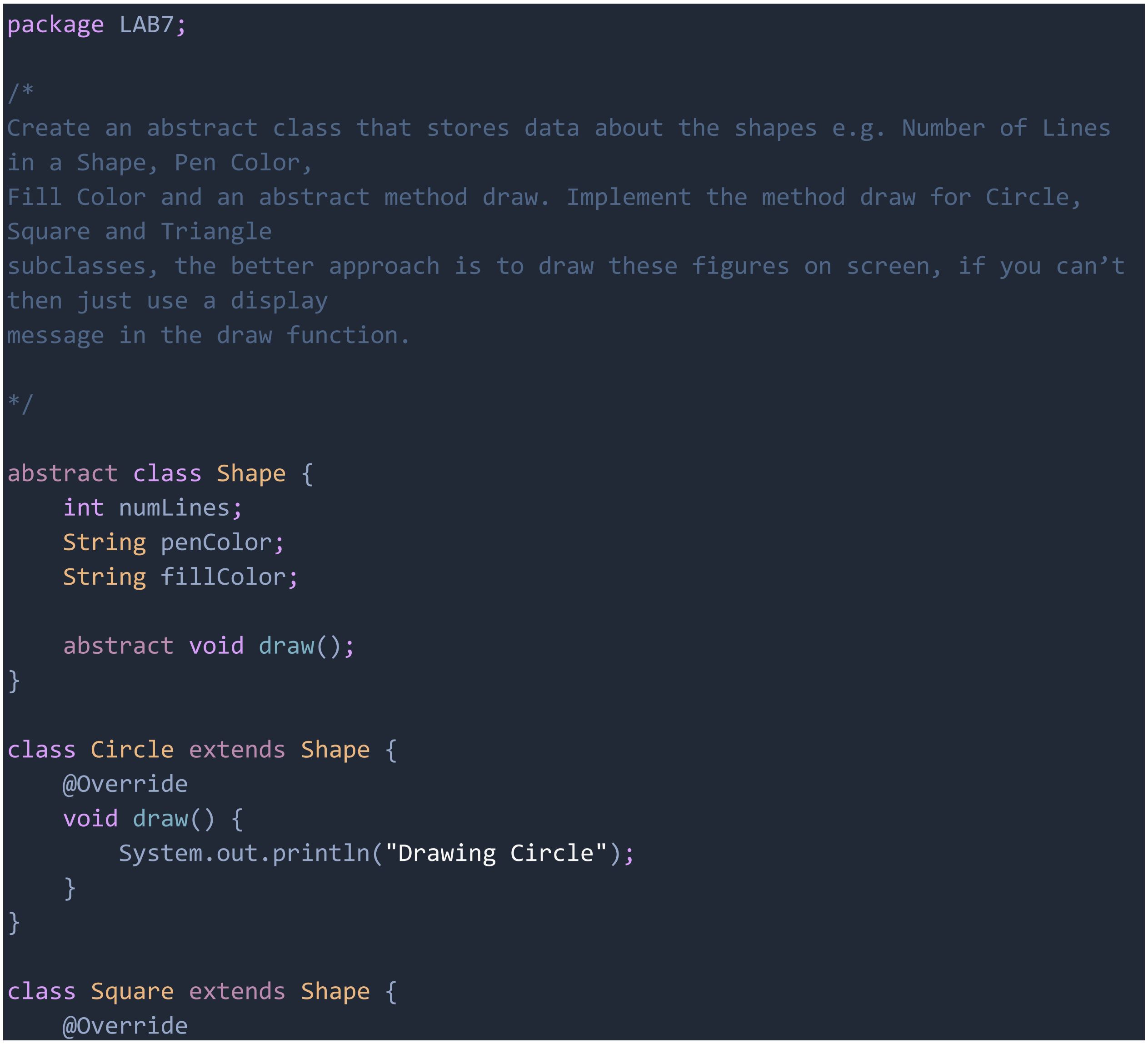
```java
    void mul() {
        if (num1 > 0 && num2 > 0) {
            super.mul();
        } else {
            System.out.println("Error: Invalid Input");
        }
    }

    @Override
    void div() {
        if (num1 > 0 && num2 > 0) {
            super.div();
        } else {
            System.out.println("Error: Invalid Input");
        }
    }
}

public class GLT2 {
    public static void main(String[] args) {
        VerifiedSimple vs = new VerifiedSimple(10, 5);
        vs.add();
        vs.sub();
        vs.mul();
        vs.div();
        // error Test
        VerifiedSimple vs2 = new VerifiedSimple(-10, 5);
        vs2.add();
        vs2.sub();
        vs2.mul();
        vs2.div();

    }

}
```

Output:

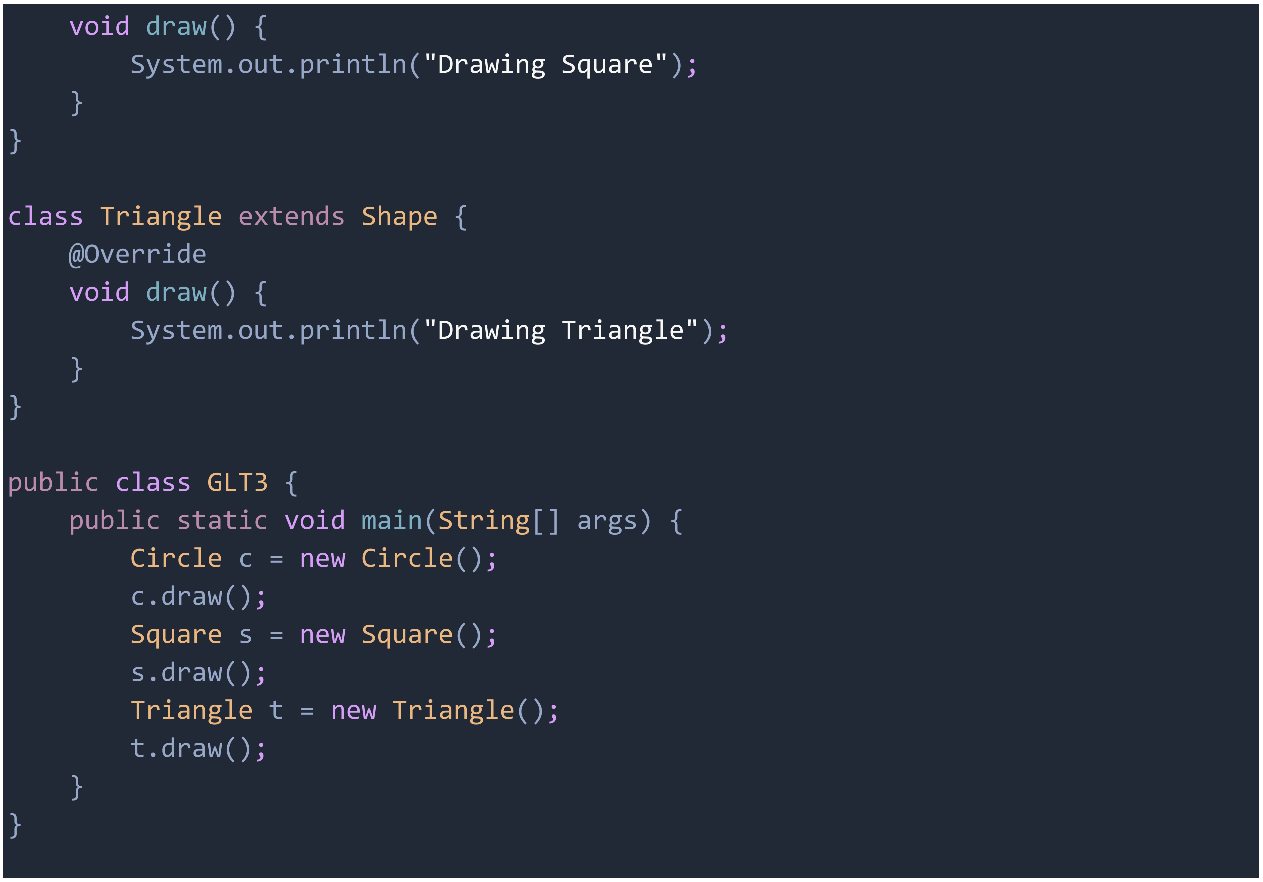


```
 '-cp' 'D:\Ishtudy Material\3rd Sem\O
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2
Error: Invalid Input
Error: Invalid Input
Error: Invalid Input
Error: Invalid Input
PS D:\Ishtudy Material\3rd Sem\OOP\LA
```

Graded Lab Task 3:

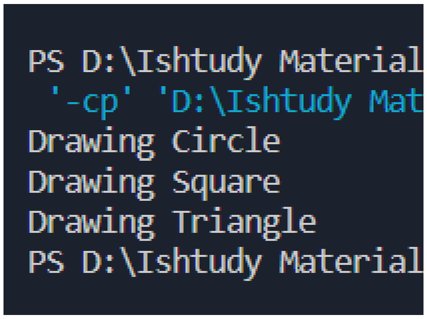


```java
package LAB7;

/*
Create an abstract class that stores data about the shapes e.g. Number of Lines
in a Shape, Pen Color,
Fill Color and an abstract method draw. Implement the method draw for Circle,
Square and Triangle
subclasses, the better approach is to draw these figures on screen, if you can’t
then just use a display
message in the draw function.

*/

abstract class Shape {
    int numLines;
    String penColor;
    String fillColor;

    abstract void draw();
}

class Circle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing Circle");
    }
}

class Square extends Shape {
    @Override
```



```java
    void draw() {
        System.out.println("Drawing Square");
    }
}

class Triangle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing Triangle");
    }
}

public class GLT3 {
    public static void main(String[] args) {
        Circle c = new Circle();
        c.draw();
        Square s = new Square();
        s.draw();
        Triangle t = new Triangle();
        t.draw();
    }
}
```

Output:



```
PS D:\Ishtudy Material
 '-cp' 'D:\Ishtudy Mat
Drawing Circle
Drawing Square
Drawing Triangle
PS D:\Ishtudy Material
```