OOPS LAB RECORD

Name: Manika Rajpal

Sap Id: 500084172

Roll No.: R214220669

Batch 3

❖ Experiment 1

➢ Lab Exercise

1. Java Versions
2. JDK and JRE

   JRE: The Java runtime environment is a collection of software tools for creating Java applications. It's the NM implementation. It is a real thing. It comprises a collection of libraries as well as other files that JVM uses during execution.

   JDK: The Java Development Kit (JDK) is a software development environment that is used to create Java applications.

   Java application development It is a real thing. It includes the Java Runtime Environment (JRE) as well as developer tools.

3. Setting Path

   For setting the permanent path of JDK,follow these steps:

   Go to MyComputer properties -> advanced tab -> environment variables -> new tab of user variable -> write path in variable name -> write path of bin folder in variable value -> ok -> ok -> ok

4. Code Editors

Source-code editors have features specifically designed to simplify and speed up typing of source code, such as syntax highlighting, indentation, autocomplete and brace matching functionality. These editors also provide a convenient way to run a compiler, interpreter, debugger, or other program relevant for the software-development process.

A source-code editor can check syntax while code is being entered and immediately warn of syntax problems.

A few source-code editors compress source code, typically converting common keywords into single-byte stokens, removing unnecessary whitespace, and converting numbers to a binary form.

Some of the best code editors for java are Eclipse, Kite, IntelliJ Idea, Blue J, Apache NetBeans.

1) Eclipse :Eclipse is an IDE for writing and compiling Java programmes.

It highlights the syntax you have written.

This tool enables you to easily debug the programme.

2) Kite

Kite is IDE for Java that automatically completes multiple line codes.

This editor supports more than 16 languages.

It helps you to code faster with no hassle.

3) IntelliJ IDEA

IntelliJ IDEA is a multi-purpose IDE which focuses on Java development.

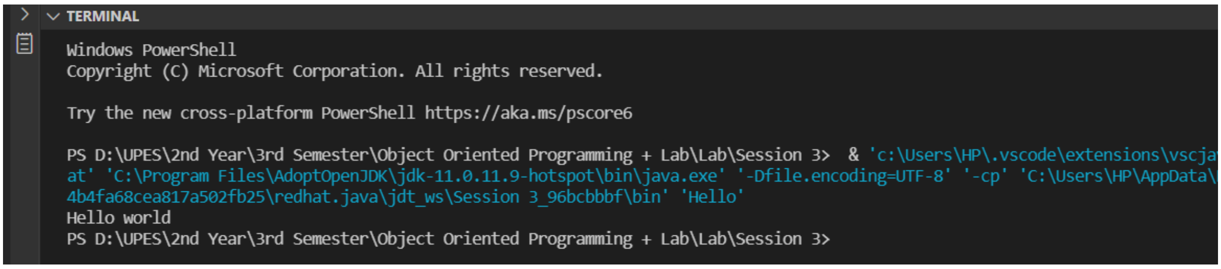
It \sis one of the best Java compiler that offers advanced support for the web, mobile, and also hybrid application development.

5. Sample Hello World Program.

- Code

```
class Hello{
    public static void main(String args[]){
     System.out.println("Hello world");
    }
}
```

- Output



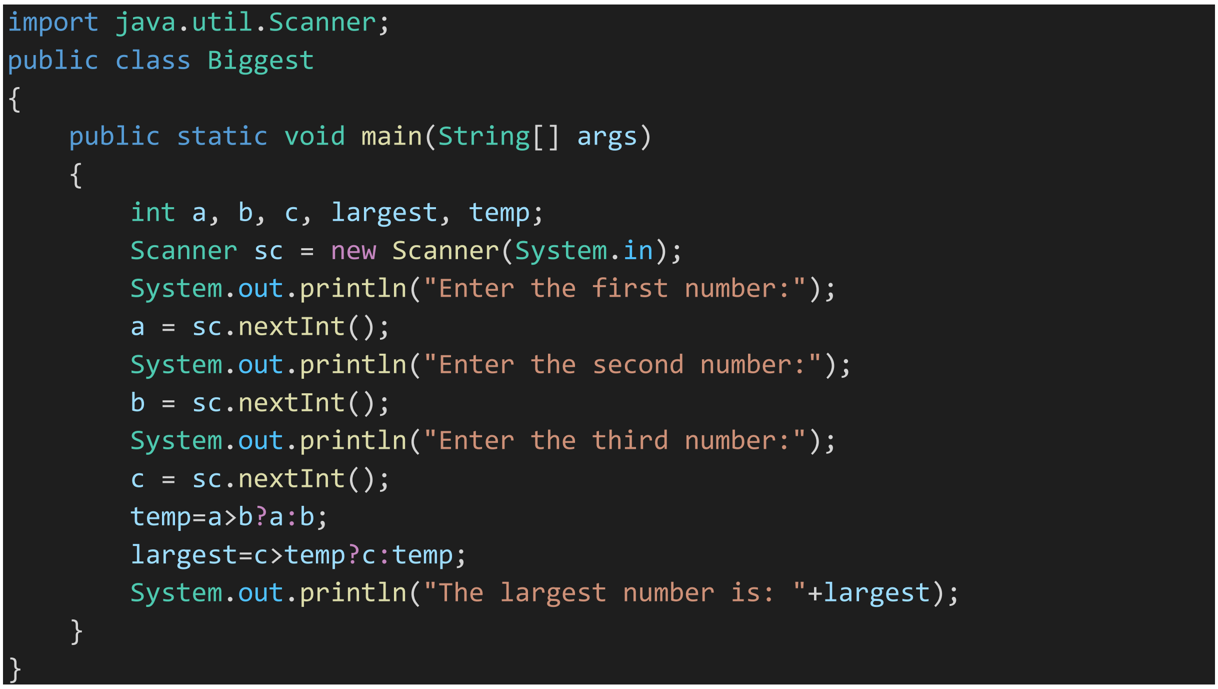
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>  & 'c:\Users\HP\.vscode\extensions\vscja
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\HP\AppData\
4b4fa68cea817a502fb25\redhat.java\jdt_ws\Session 3_96bcbbbf\bin' 'Hello'
Hello world
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>
```

❖ Experiment 2

➢ Lab Exercise

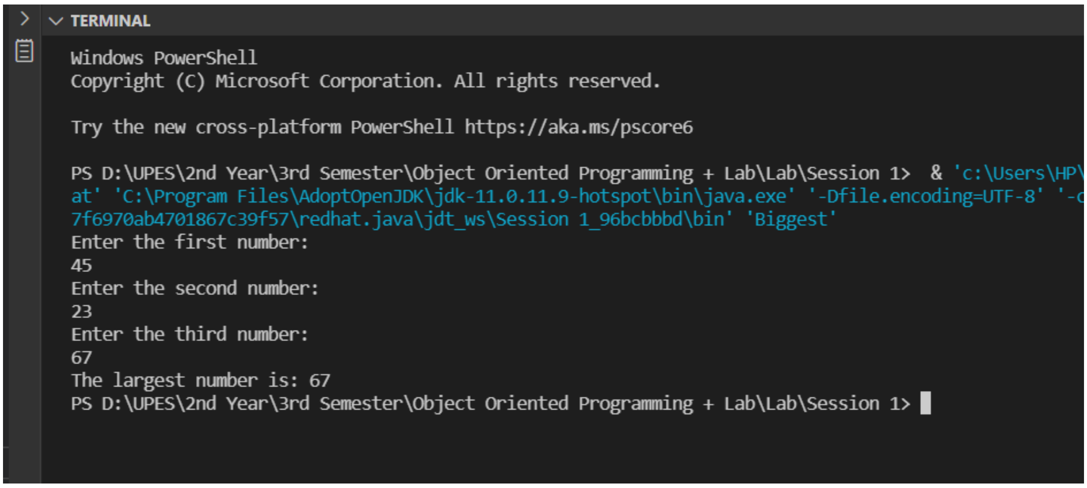
1. To find the largest of three numbers.

- Code



```java
import java.util.Scanner;
public class Biggest
{
    public static void main(String[] args)
    {
        int a, b, c, largest, temp;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first number:");
        a = sc.nextInt();
        System.out.println("Enter the second number:");
        b = sc.nextInt();
        System.out.println("Enter the third number:");
        c = sc.nextInt();
        temp=a>b?a:b;
        largest=c>temp?c:temp;
        System.out.println("The largest number is: "+largest);
    }
}
```

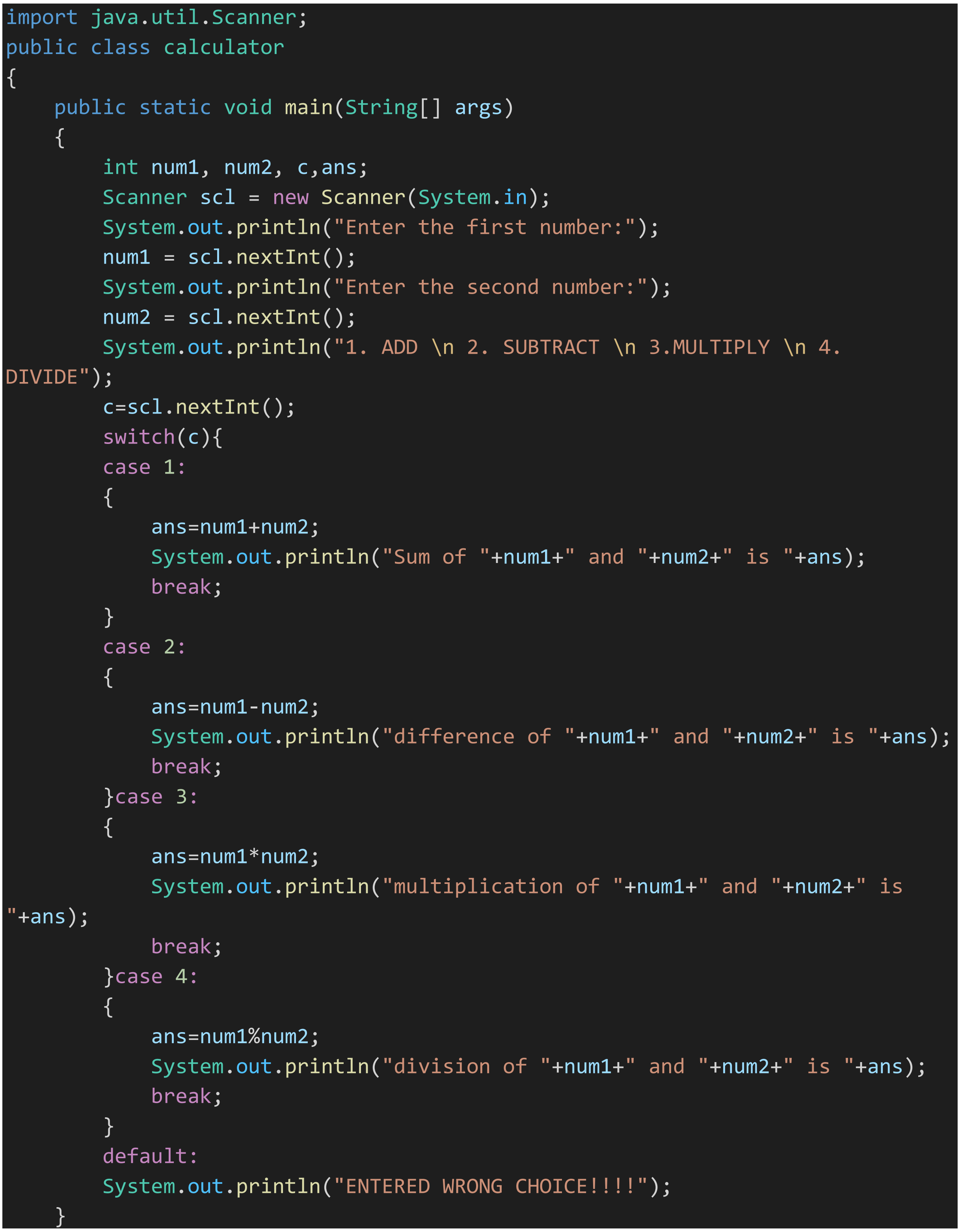
- Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 1>  & 'c:\Users\HP\
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-c
7f6970ab4701867c39f57\redhat.java\jdt_ws\Session 1_96bcbbbd\bin' 'Biggest'
Enter the first number:
45
Enter the second number:
23
Enter the third number:
67
The largest number is: 67
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 1> █
```

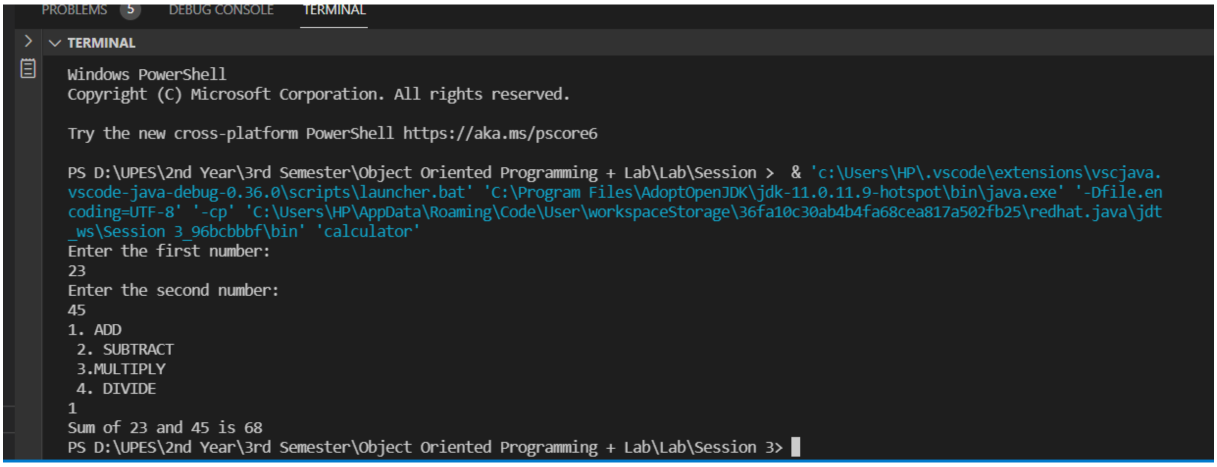
2. Write a program to implement a command line calculator. (Try for Add sub Mul Division in same program for 2 digits.) Integer.parseInt will be used For e.g. java calc 20 + 30 Output should be Sum of 20 and 30 is 50 java calc 50 * 30 Output should be Product of 50 and 30 is 1500 .

   - Code



```
import java.util.Scanner;
public class calculator
{
    public static void main(String[] args)
    {
        int num1, num2, c,ans;
        Scanner scl = new Scanner(System.in);
        System.out.println("Enter the first number:");
        num1 = scl.nextInt();
        System.out.println("Enter the second number:");
        num2 = scl.nextInt();
        System.out.println("1. ADD \n 2. SUBTRACT \n 3.MULTIPLY \n 4.
DIVIDE");
        c=scl.nextInt();
        switch(c){
        case 1:
        {
            ans=num1+num2;
            System.out.println("Sum of "+num1+" and "+num2+" is "+ans);
            break;
        }
        case 2:
        {
            ans=num1-num2;
            System.out.println("difference of "+num1+" and "+num2+" is "+ans);
            break;
        }case 3:
        {
            ans=num1*num2;
            System.out.println("multiplication of "+num1+" and "+num2+" is
"+ans);
            break;
        }case 4:
        {
            ans=num1%num2;
            System.out.println("division of "+num1+" and "+num2+" is "+ans);
            break;
        }
        default:
        System.out.println("ENTERED WRONG CHOICE!!!!");
    }
}
}
```

- Output

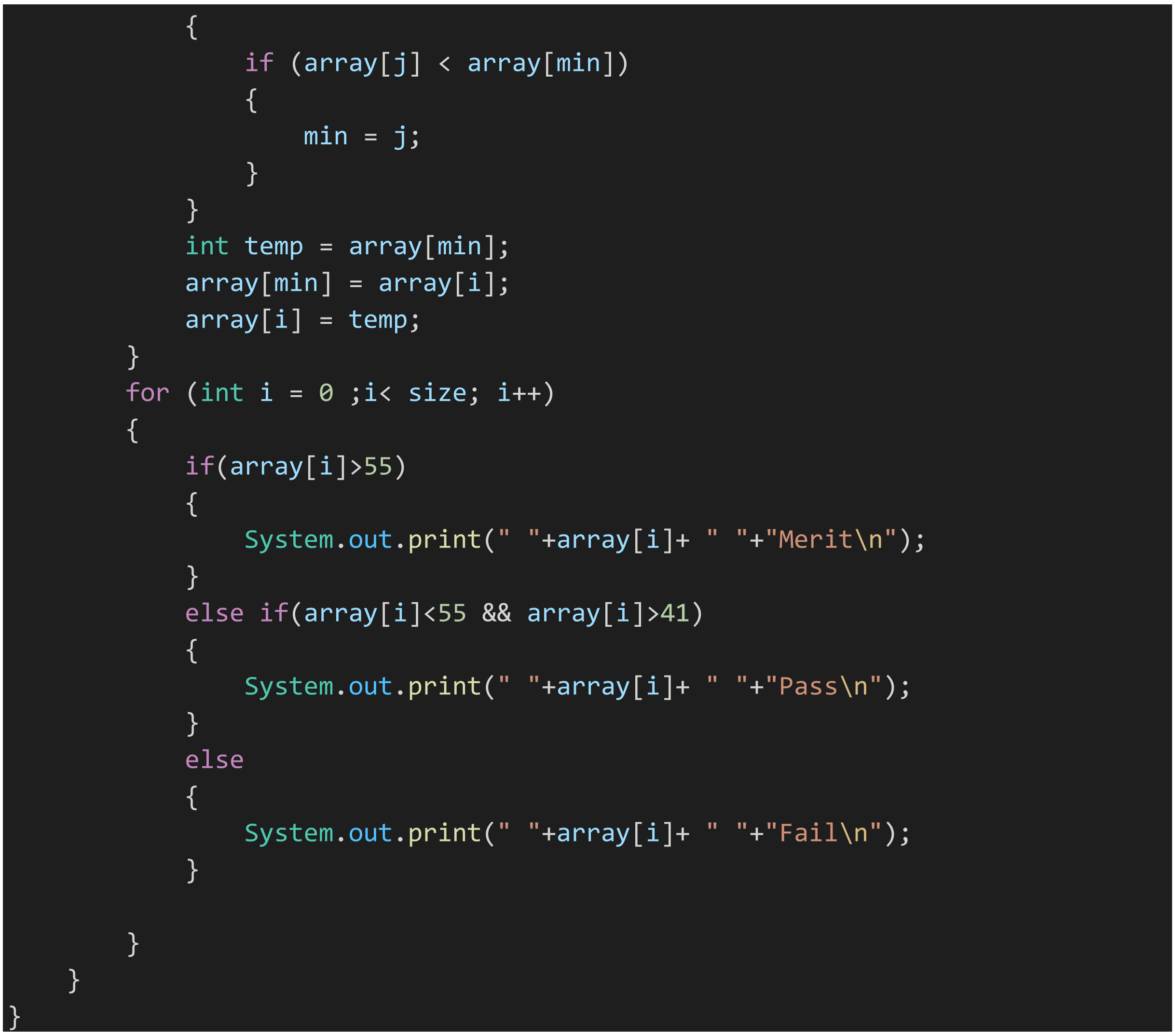


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session >  & 'c:\Users\HP\.vscode\extensions\vscjava.
vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.en
coding=UTF-8' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\workspaceStorage\36fa10c30ab4b4fa68cea817a502fb25\redhat.java\jdt
_ws\Session 3_96bcbbbf\bin' 'calculator'
Enter the first number:
23
Enter the second number:
45
1. ADD
 2. SUBTRACT
 3.MULTIPLY
 4. DIVIDE
1
Sum of 23 and 45 is 68
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>
```

3. Write a program to accept 10 student's marks in an array, arrange it into ascending order, convert into the following grades and print marks and grades in the tabular form. Between 40 and 50 : PASS Between 51 and 75 : MERIT and above : DISTINCTION.

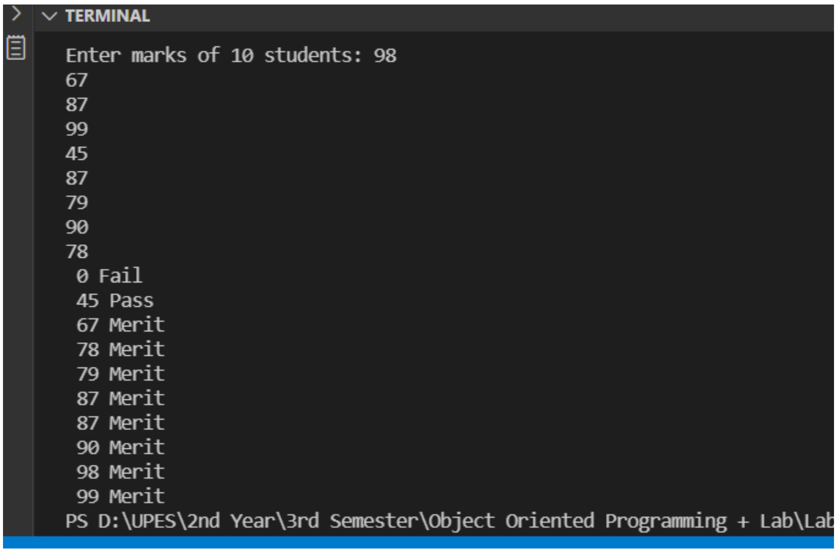
- Code

```
import java.util.Scanner;

public class marks
{
    public static void main(String args[])
    {
        int size;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter marks of 10 students: ");
        int[] array = new int[10];
        size=array.length;
        for(int i=1; i<10; i++)
        {
            array[i] = sc.nextInt();
        }
        for (int i = 0 ;i< size-1; i++)
        {
            int min = i;
            for (int j = i+1; j<size; j++)
```



```
            {
                if (array[j] < array[min])
                {
                    min = j;
                }
            }
            int temp = array[min];
            array[min] = array[i];
            array[i] = temp;
        }
        for (int i = 0 ;i< size; i++)
        {
            if(array[i]>55)
            {
                System.out.print(" "+array[i]+ " "+"Merit\n");
            }
            else if(array[i]<55 && array[i]>41)
            {
                System.out.print(" "+array[i]+ " "+"Pass\n");
            }
            else
            {
                System.out.print(" "+array[i]+ " "+"Fail\n");
            }

        }
    }
}
```

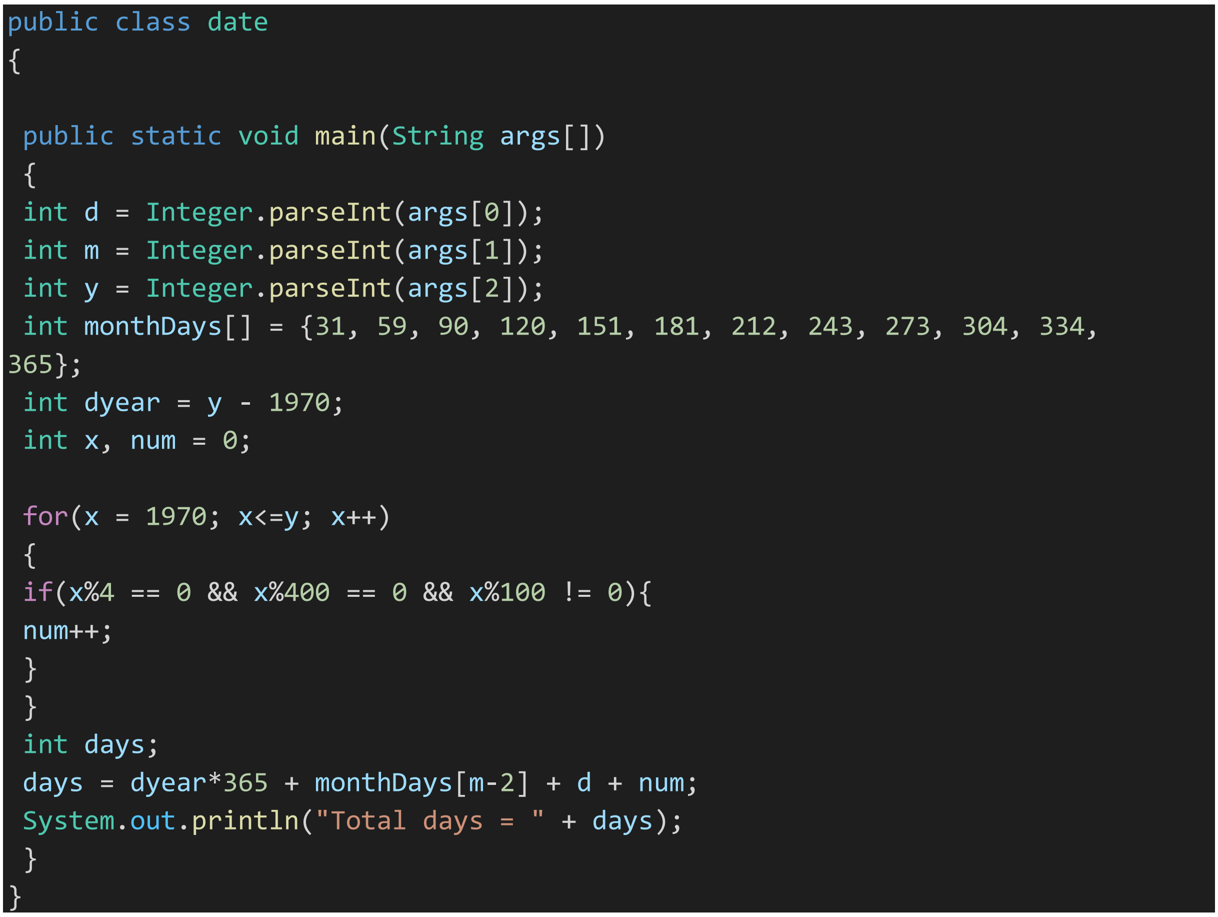
- Output



```
Enter marks of 10 students: 98
67
87
99
45
87
79
90
78
 0 Fail
 45 Pass
 67 Merit
 78 Merit
 79 Merit
 87 Merit
 87 Merit
 90 Merit
 98 Merit
 99 Merit
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab
```

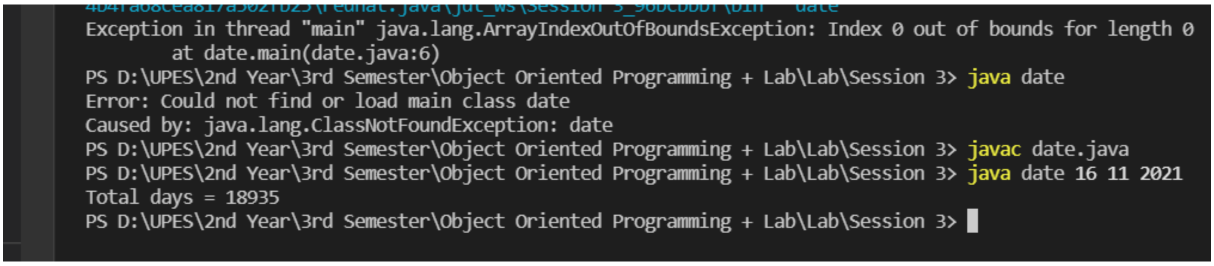
4.WAP to Take input as DD MM YYYY(04 08 2021) in command line and calculate number of days since 1 January 1970.

- Code



```
public class date
{

 public static void main(String args[])
 {
 int d = Integer.parseInt(args[0]);
 int m = Integer.parseInt(args[1]);
 int y = Integer.parseInt(args[2]);
 int monthDays[] = {31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 
365};
 int dyear = y - 1970;
 int x, num = 0;

 for(x = 1970; x<=y; x++)
 {
 if(x%4 == 0 && x%400 == 0 && x%100 != 0){
 num++;
 }
 }
 int days;
 days = dyear*365 + monthDays[m-2] + d + num;
 System.out.println("Total days = " + days);
 }
}
```

- Output



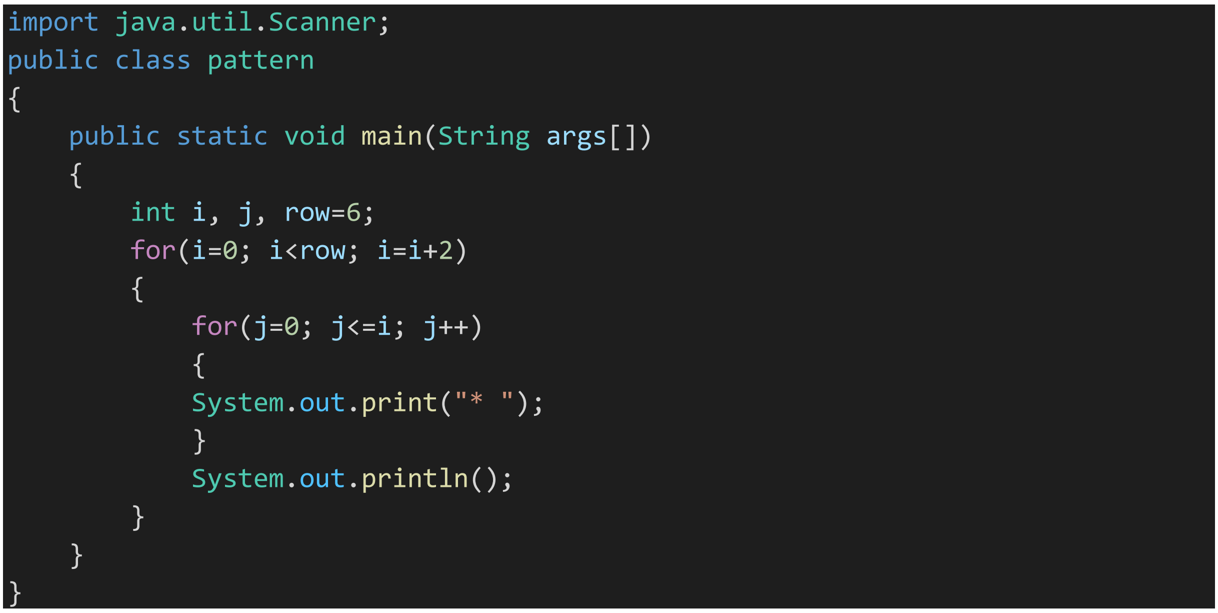
```
4b4fa68cea817a3021b25\redhat.java\jdt_ws\Session 3_96bcbbbf\bin' 'date'
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0
        at date.main(date.java:6)
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3> java date
Error: Could not find or load main class date
Caused by: java.lang.ClassNotFoundException: date
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3> javac date.java
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3> java date 16 11 2021
Total days = 18935
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3> █
```

5. WAP to print the following pattern using loops

```
*

***
```

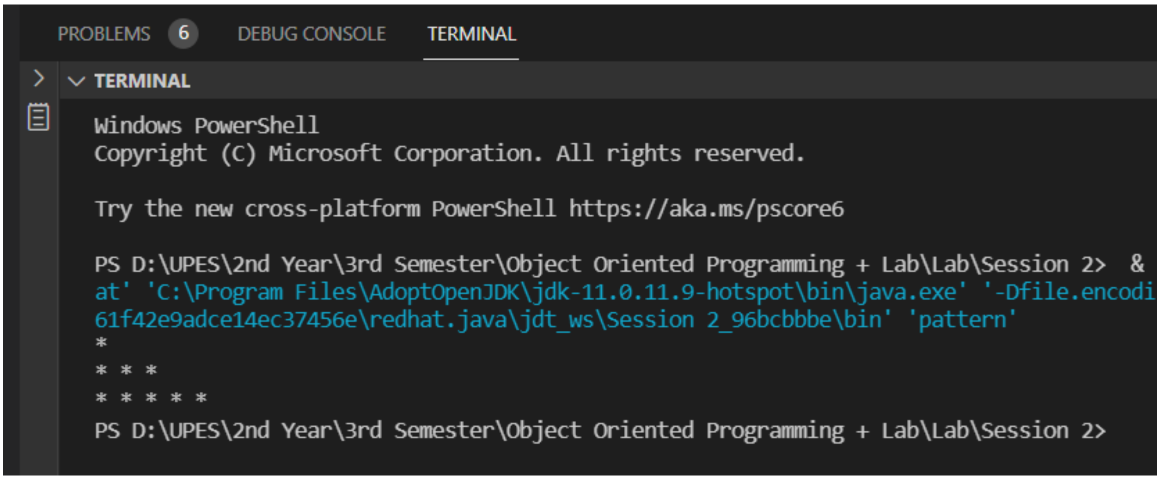
*****

- Code



```java
import java.util.Scanner;
public class pattern
{
    public static void main(String args[])
    {
        int i, j, row=6;
        for(i=0; i<row; i=i+2)
        {
            for(j=0; j<=i; j++)
            {
            System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

- Output



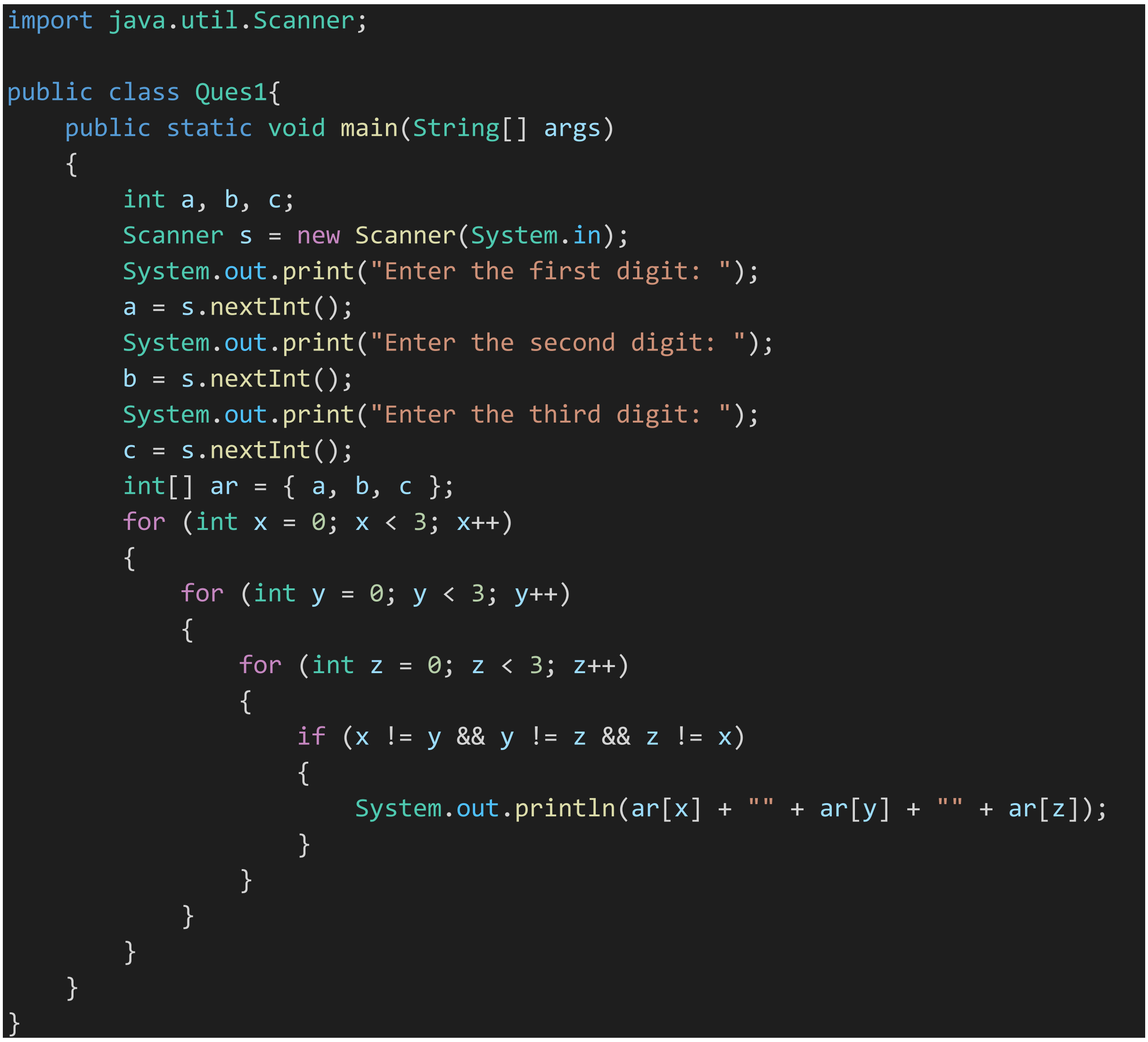
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>  & 
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encodi
61f42e9adce14ec37456e\redhat.java\jdt_ws\Session 2_96bcbbbe\bin' 'pattern'
*
* * *
* * * * *
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>
```

❖ Experiment 3

➢ Lab Exercise

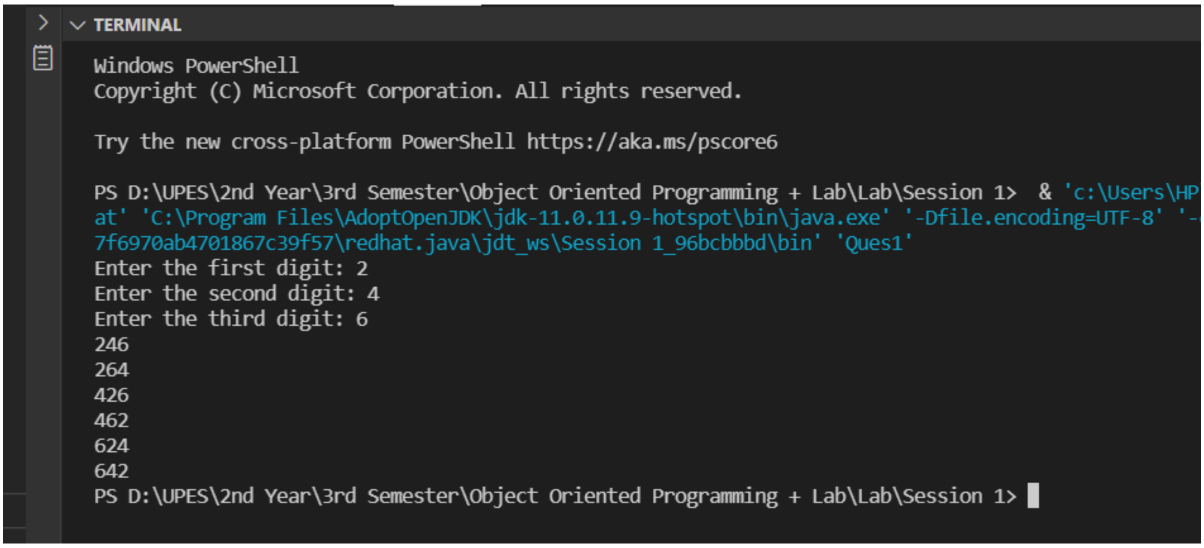
1. Write a program to accept three digits (i.e., 0 - 9) and print all its possible combinations. (For example if the three digits are 1, 2, 3 than all possible combinations are : 123, 132, 213, 231, 312, 321.)

- Code



```java
import java.util.Scanner;

public class Ques1{
    public static void main(String[] args)
    {
        int a, b, c;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the first digit: ");
        a = s.nextInt();
        System.out.print("Enter the second digit: ");
        b = s.nextInt();
        System.out.print("Enter the third digit: ");
        c = s.nextInt();
        int[] ar = { a, b, c };
        for (int x = 0; x < 3; x++)
        {
            for (int y = 0; y < 3; y++)
            {
                for (int z = 0; z < 3; z++)
                {
                    if (x != y && y != z && z != x)
                    {
                        System.out.println(ar[x] + "" + ar[y] + "" + ar[z]);
                    }
                }
            }
        }
    }
}
```

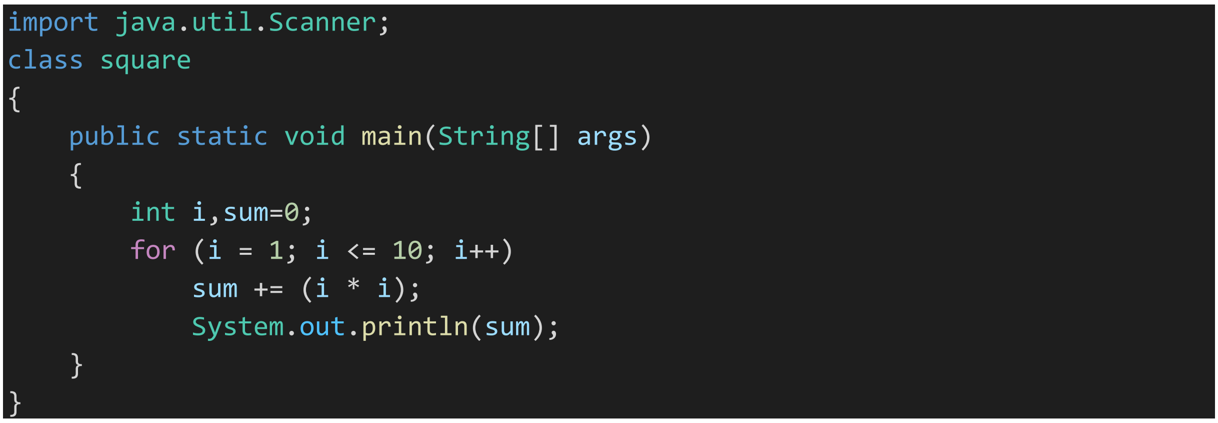
- Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 1>  & 'c:\Users\HP
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '-
7f6970ab4701867c39f57\redhat.java\jdt_ws\Session 1_96bcbbbd\bin' 'Ques1'
Enter the first digit: 2
Enter the second digit: 4
Enter the third digit: 6
246
264
426
462
624
642
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 1> 
```

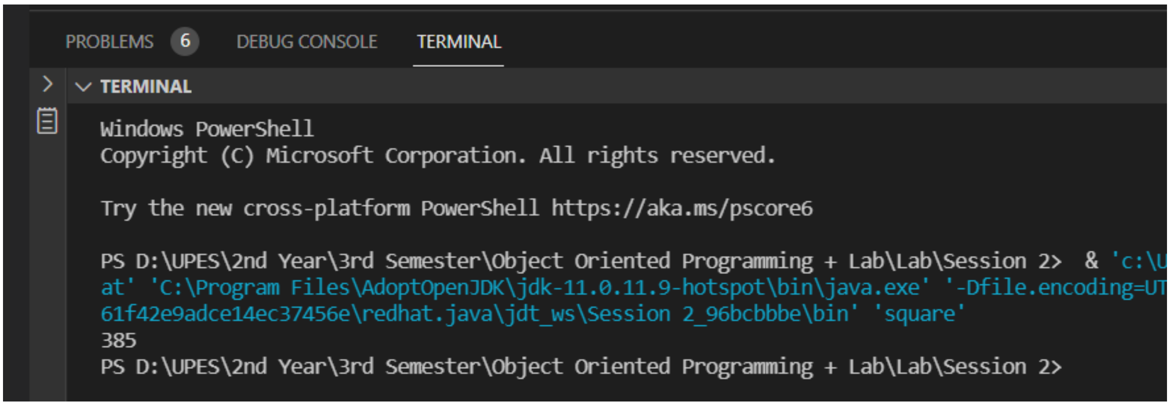
2. Write a Java Program to accept 10 numbers in an array and compute the square of each number. Print the sum of these numbers.

- Code



```java
import java.util.Scanner;
class square
{
    public static void main(String[] args)
    {
        int i,sum=0;
        for (i = 1; i <= 10; i++)
            sum += (i * i);
            System.out.println(sum);
    }
}
```

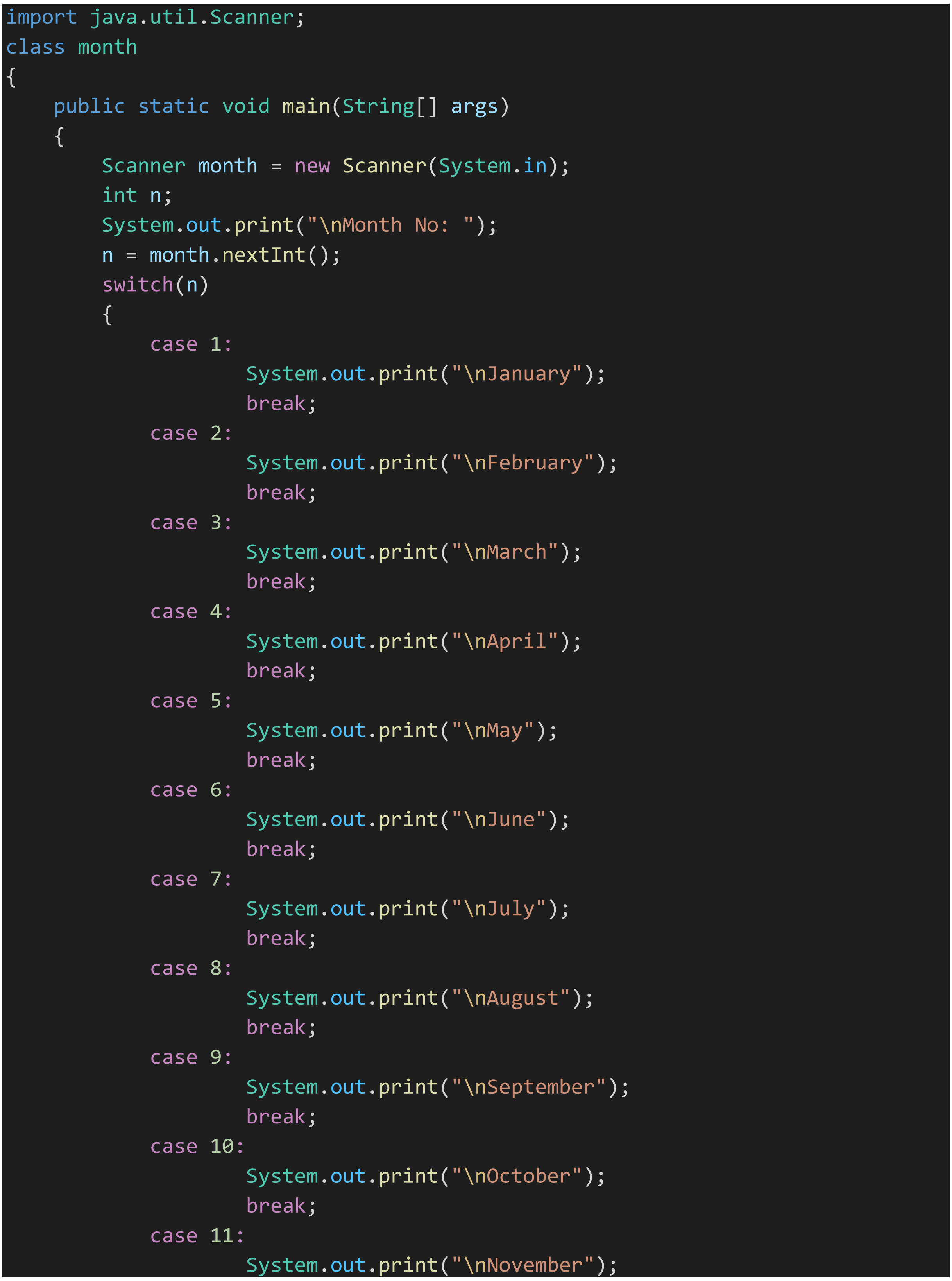
- Output



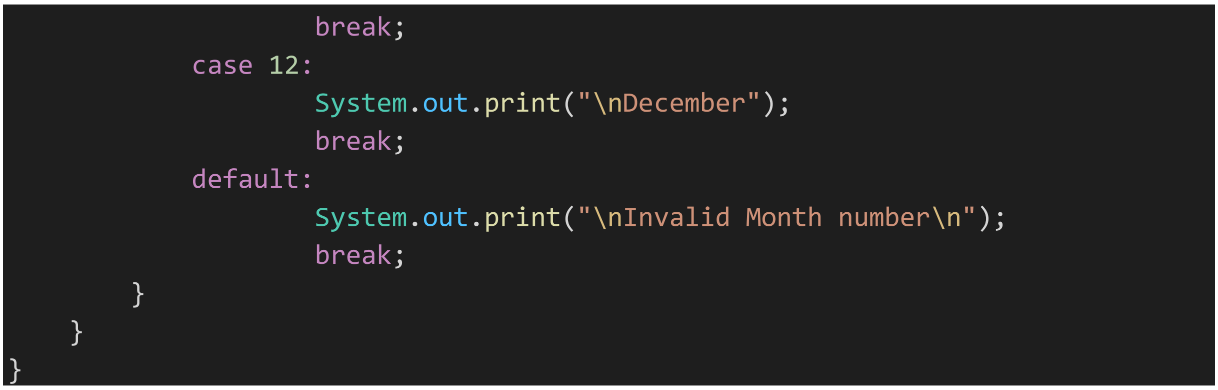
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>  & 'c:\U
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UT
61f42e9adce14ec37456e\redhat.java\jdt_ws\Session 2_96bcbbbe\bin' 'square'
385
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>
```

3. Write a program to input a number of a month (1 - 12) and print its equivalent name of the month. (e.g 1 to Jan, 2 to Feb. 12 to Dec.)

- Code

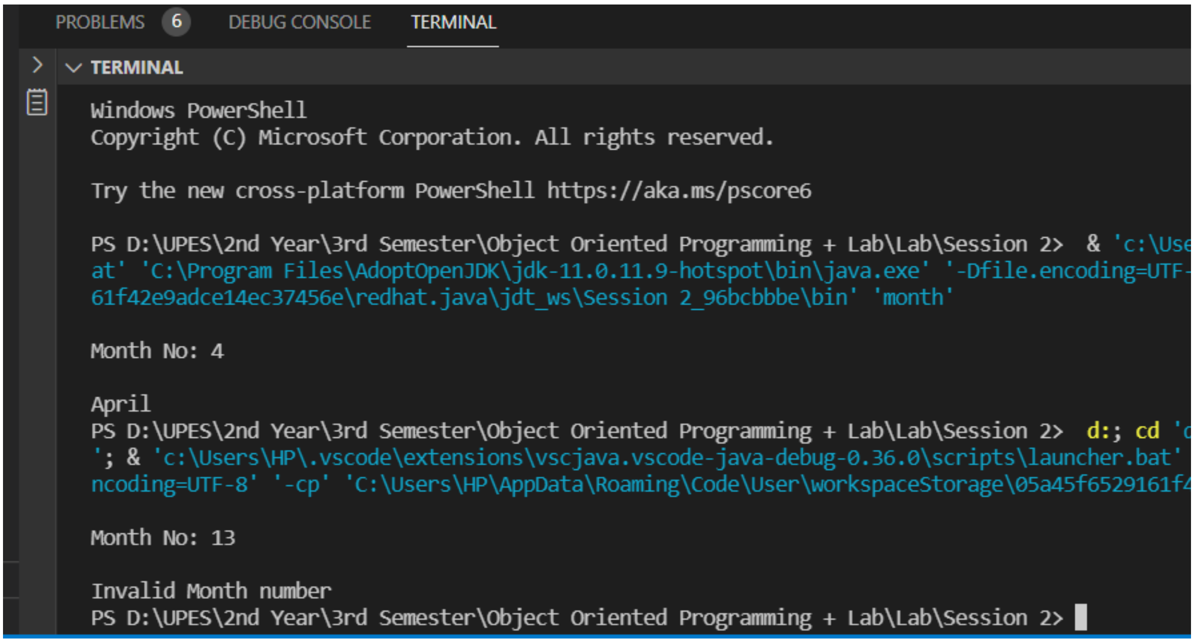


```java
import java.util.Scanner;
class month
{
    public static void main(String[] args)
    {
        Scanner month = new Scanner(System.in);
        int n;
        System.out.print("\nMonth No: ");
        n = month.nextInt();
        switch(n)
        {
            case 1:
                    System.out.print("\nJanuary");
                    break;
            case 2:
                    System.out.print("\nFebruary");
                    break;
            case 3:
                    System.out.print("\nMarch");
                    break;
            case 4:
                    System.out.print("\nApril");
                    break;
            case 5:
                    System.out.print("\nMay");
                    break;
            case 6:
                    System.out.print("\nJune");
                    break;
            case 7:
                    System.out.print("\nJuly");
                    break;
            case 8:
                    System.out.print("\nAugust");
                    break;
            case 9:
                    System.out.print("\nSeptember");
                    break;
            case 10:
                    System.out.print("\nOctober");
                    break;
            case 11:
                    System.out.print("\nNovember");
```



```
                    break;
            case 12:
                    System.out.print("\nDecember");
                    break;
            default:
                    System.out.print("\nInvalid Month number\n");
                    break;
        }
    }
}
```

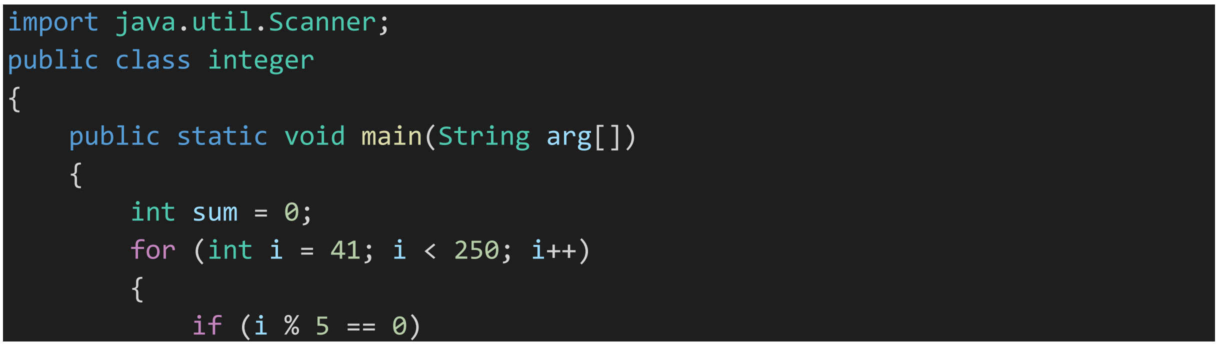
- Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>  & 'c:\Use
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-
61f42e9adce14ec37456e\redhat.java\jdt_ws\Session 2_96bcbbbe\bin' 'month'

Month No: 4

April
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2>  d:; cd 'c
'; & 'c:\Users\HP\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat'
ncoding=UTF-8' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\workspaceStorage\05a45f6529161f4

Month No: 13

Invalid Month number
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2> 
```

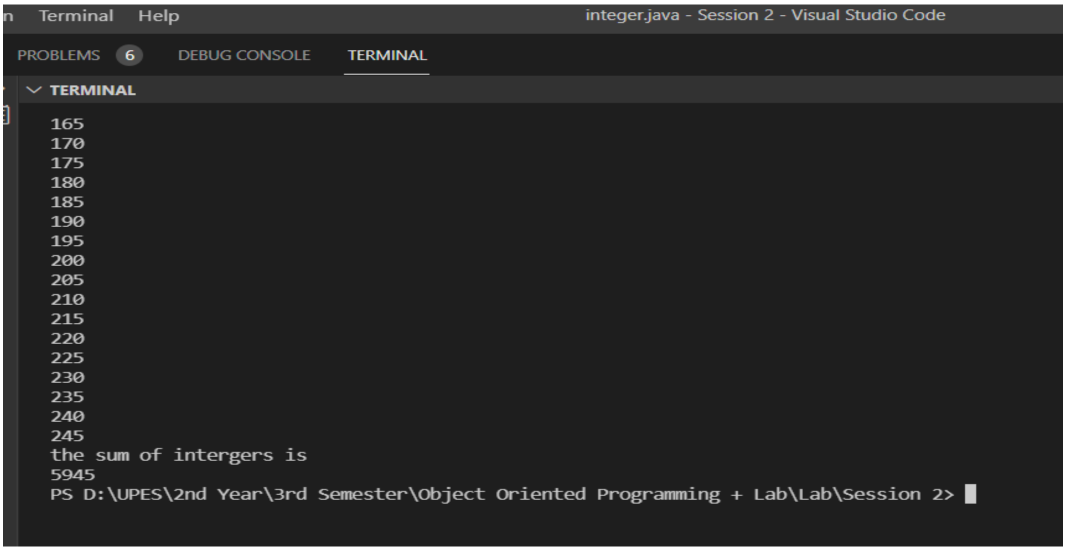
4. Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5.

- Code



```
import java.util.Scanner;
public class integer
{
    public static void main(String arg[])
    {
        int sum = 0;
        for (int i = 41; i < 250; i++)
        {
            if (i % 5 == 0)
            {
                System.out.println(i);
                sum = sum + i;
            }
        }
        System.out.println("the sum of intergers is \n" + sum);
    }
}
```

- Output



```
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
the sum of intergers is
5945
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 2> 
```

❖ Experiment 4

➢ Lab Exercise

1. Write a Java program to show that private member of a super class cannot be accessed from derived classes.

   - Code

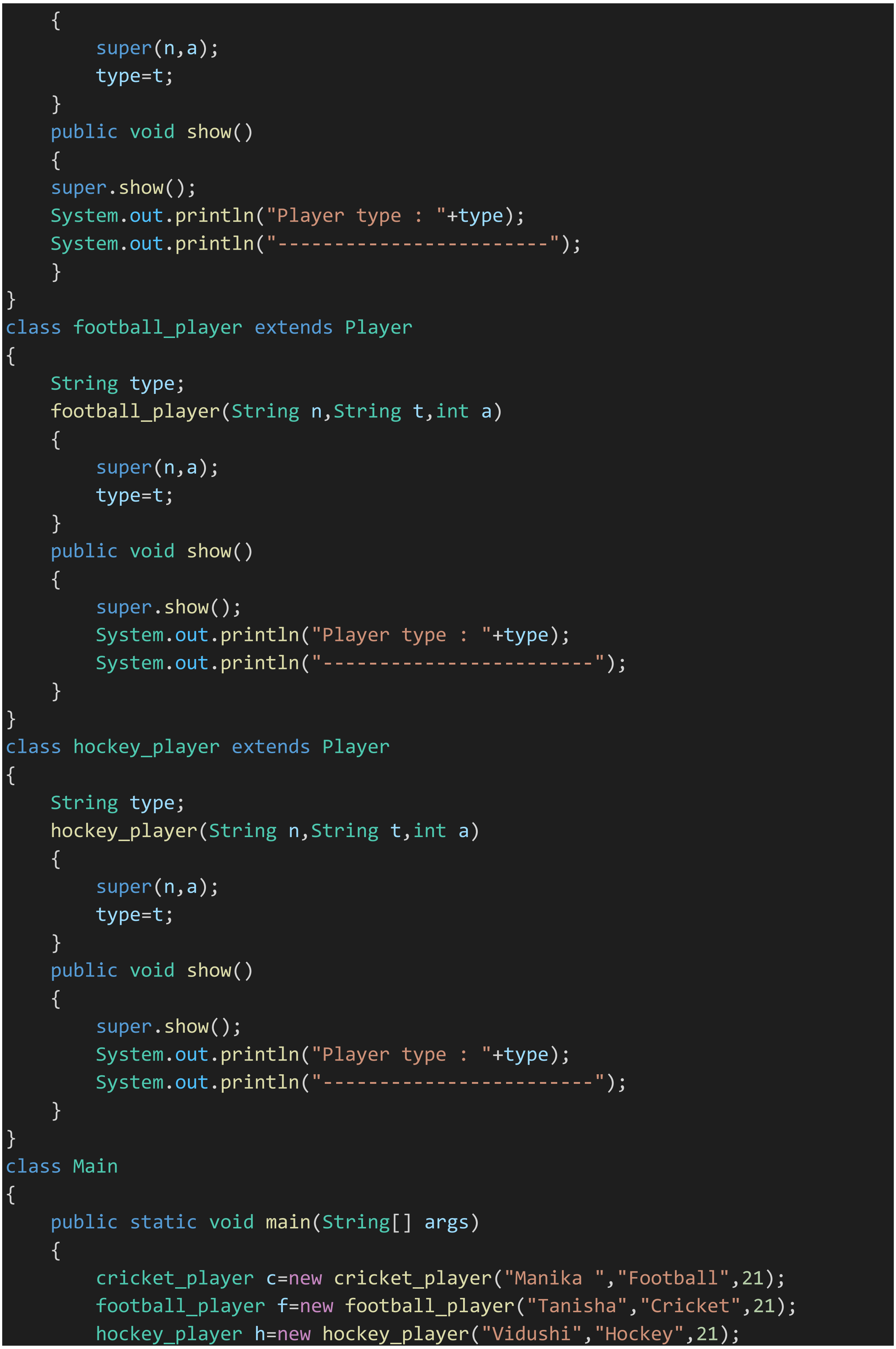
```
class shape
{
    private int l,b;
        int area()
        {return(l*b);}
}
class cuboid extends shape
{
    int h;
    cuboid(int l,int b,int d)
{
    super(l,b);
    h=d;
}
    int volume()
{
    return(area()*h);
}
}
    class lab6
{
    public static void main(String args[])
    {
    cuboid c=new cuboid(16,12,4);
    int ar=c.area();
    int vl=c.volume();
    System.out.println("Area of Room : "+ar);
    System.out.println("Volume of Room : "+vl);
    }
}
```

- Output

2. Write a program in Java to create a Player class. Inherit the classes Cricket _Player, Football _Player and Hockey_ Player from Player class.

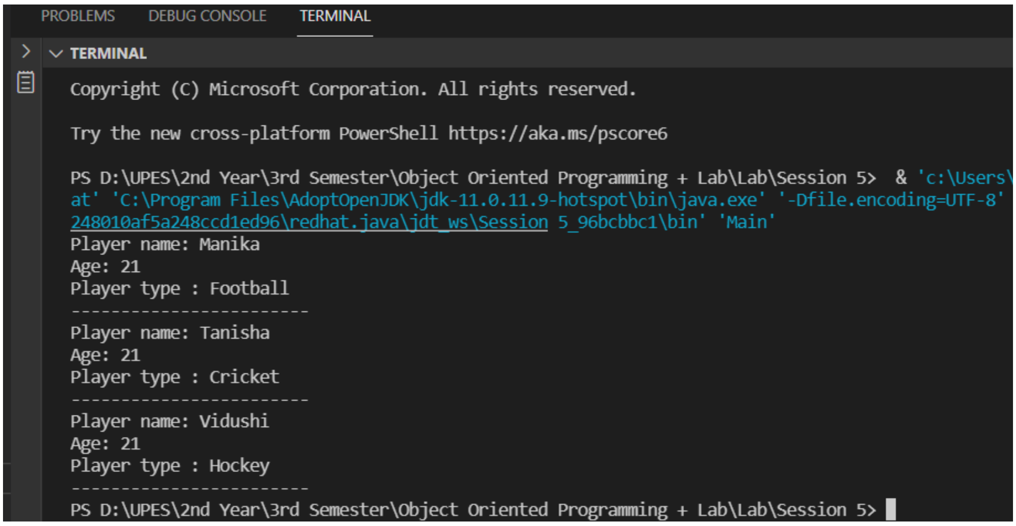
- Code

```
class Player
{
    String name;
    int age;
    Player(String n,int a)
    {
        name=n; age=a;
    }
    void show()
    {
        System.out.println("Player name: "+name);
        System.out.println("Age: "+age);
    }
}
class cricket_player extends Player
{
    String type;
    cricket_player(String n,String t,int a)
```



```
    {
        super(n,a);
        type=t;
    }
    public void show()
    {
    super.show();
    System.out.println("Player type : "+type);
    System.out.println("------------------------");
    }
}
class football_player extends Player
{
    String type;
    football_player(String n,String t,int a)
    {
        super(n,a);
        type=t;
    }
    public void show()
    {
        super.show();
        System.out.println("Player type : "+type);
        System.out.println("------------------------");
    }
}
class hockey_player extends Player
{
    String type;
    hockey_player(String n,String t,int a)
    {
        super(n,a);
        type=t;
    }
    public void show()
    {
        super.show();
        System.out.println("Player type : "+type);
        System.out.println("------------------------");
    }
}
class Main
{
    public static void main(String[] args)
    {
        cricket_player c=new cricket_player("Manika ","Football",21);
        football_player f=new football_player("Tanisha","Cricket",21);
        hockey_player h=new hockey_player("Vidushi","Hockey",21);
        c.show();
        f.show();
        h.show();
    }
}
```

- Output

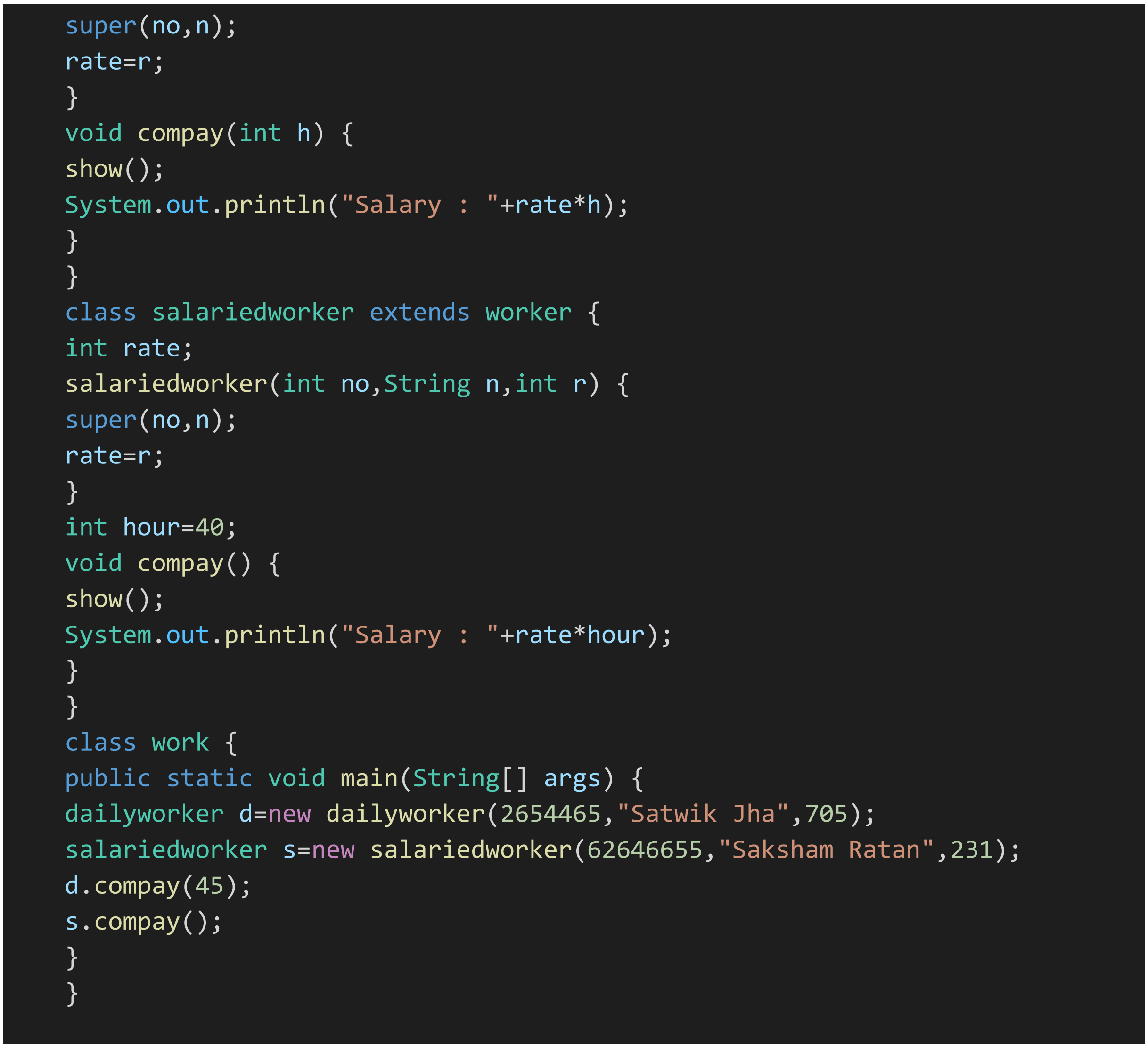


```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 5>  & 'c:\Users\
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8'
248010af5a248ccd1ed96\redhat.java\jdt_ws\Session 5_96bcbbc1\bin' 'Main'
Player name: Manika
Age: 21
Player type : Football
------------------------
Player name: Tanisha
Age: 21
Player type : Cricket
------------------------
Player name: Vidushi
Age: 21
Player type : Hockey
------------------------
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 5> 
```

3. Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days he/she works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

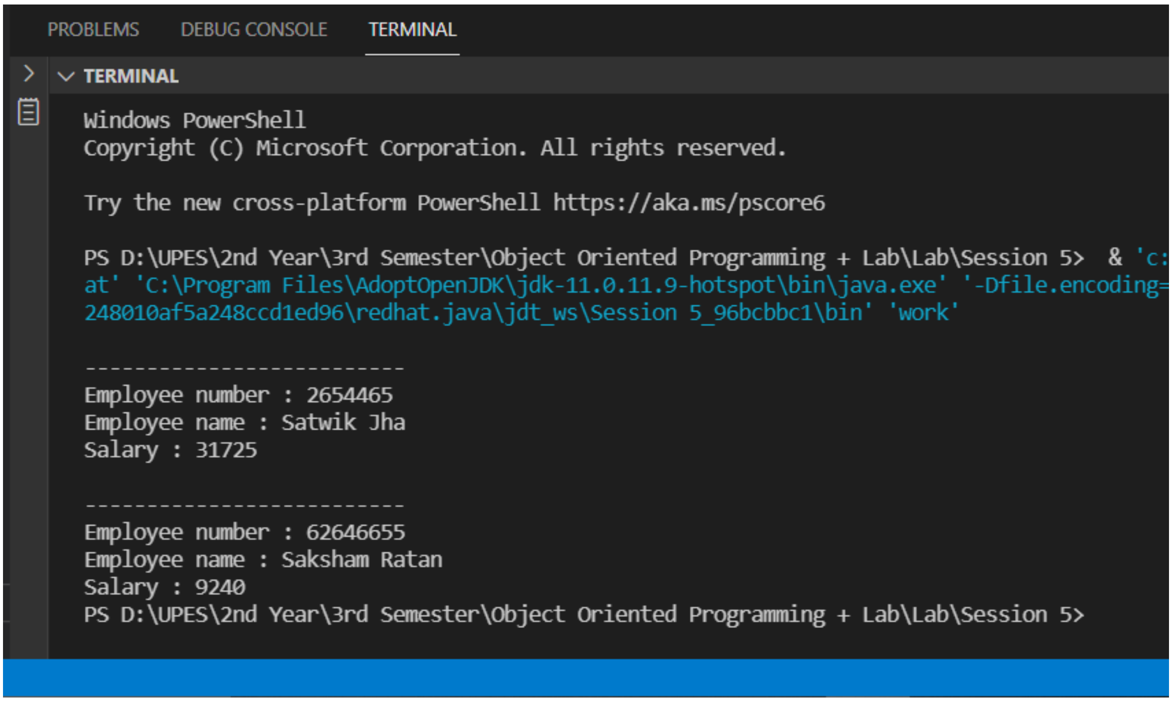
- Code

```
class worker {
    String name;
    int empno;
    worker(int no,String n){
    empno=no; name=n;
    }
    void show() {
    System.out.println("\n-------------------------");
    System.out.println("Employee number : "+empno);
    System.out.println("Employee name : "+name);
    }
    }
    class dailyworker extends worker {
    int rate;
    dailyworker(int no,String n,int r) {
```



```java
super(no,n);
rate=r;
}
void compay(int h) {
show();
System.out.println("Salary : "+rate*h);
}
}
class salariedworker extends worker {
int rate;
salariedworker(int no,String n,int r) {
super(no,n);
rate=r;
}
int hour=40;
void compay() {
show();
System.out.println("Salary : "+rate*hour);
}
}
class work {
public static void main(String[] args) {
dailyworker d=new dailyworker(2654465,"Satwik Jha",705);
salariedworker s=new salariedworker(62646655,"Saksham Ratan",231);
d.compay(45);
s.compay();
}
}
```

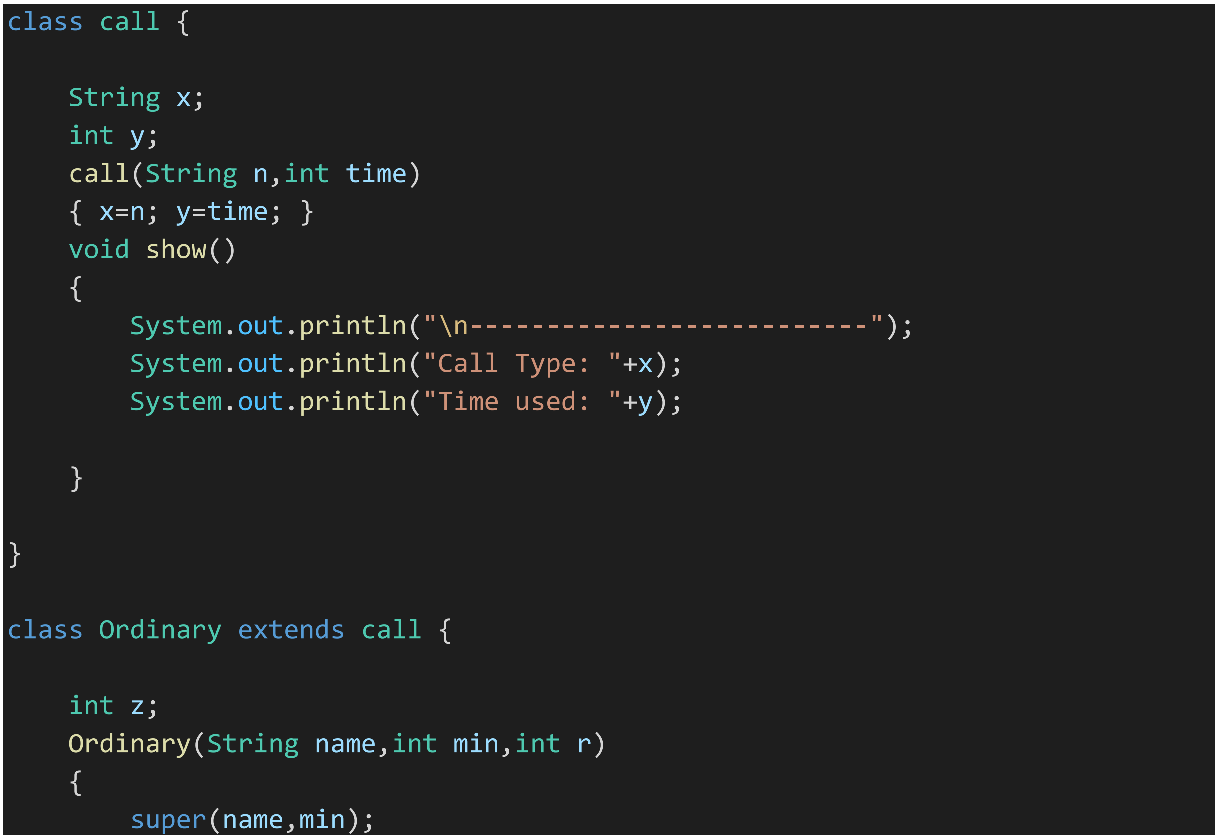
- Output



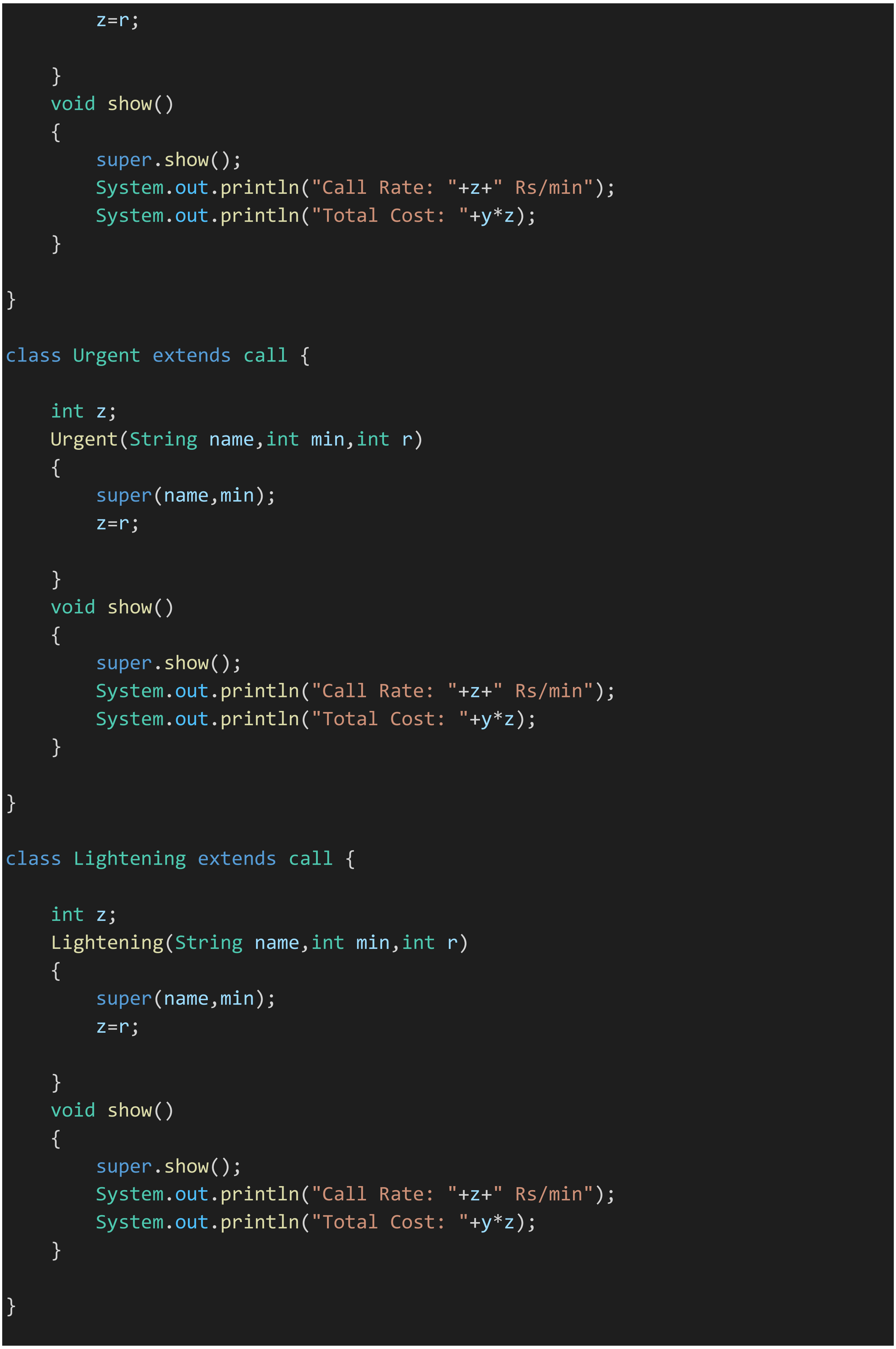
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 5>  & 'c:
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=
248010af5a248ccd1ed96\redhat.java\jdt_ws\Session 5_96bcbbc1\bin' 'work'

--------------------------
Employee number : 2654465
Employee name : Satwik Jha
Salary : 31725

--------------------------
Employee number : 62646655
Employee name : Saksham Ratan
Salary : 9240
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 5>
```

4. Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.

- Code



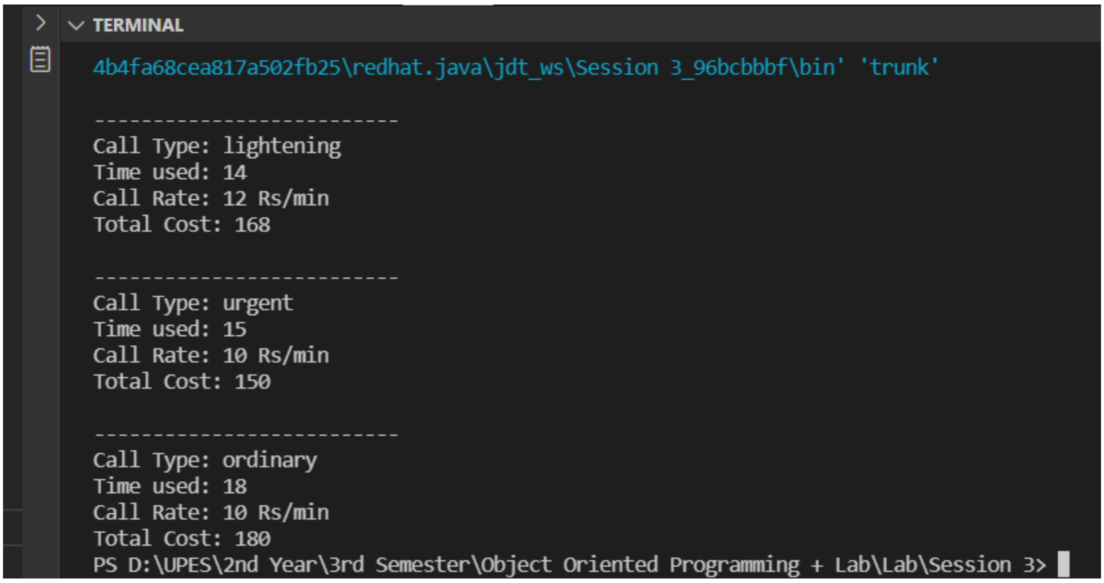
```java
class call {

    String x;
    int y;
    call(String n,int time)
    { x=n; y=time; }
    void show()
    {
        System.out.println("\n-------------------------");
        System.out.println("Call Type: "+x);
        System.out.println("Time used: "+y);

    }

}

class Ordinary extends call {

    int z;
    Ordinary(String name,int min,int r)
    {
        super(name,min);
```



```
        z=r;

    }
    void show()
    {
        super.show();
        System.out.println("Call Rate: "+z+" Rs/min");
        System.out.println("Total Cost: "+y*z);
    }

}

class Urgent extends call {

    int z;
    Urgent(String name,int min,int r)
    {
        super(name,min);
        z=r;

    }
    void show()
    {
        super.show();
        System.out.println("Call Rate: "+z+" Rs/min");
        System.out.println("Total Cost: "+y*z);
    }

}

class Lightening extends call {

    int z;
    Lightening(String name,int min,int r)
    {
        super(name,min);
        z=r;

    }
    void show()
    {
        super.show();
        System.out.println("Call Rate: "+z+" Rs/min");
        System.out.println("Total Cost: "+y*z);
    }

}
```

```java
public class trunk {
    public static void main(String[] args) {
        Lightening lc=new Lightening("lightening",14, 12);
        Urgent uc=new Urgent("urgent", 15, 10);
        Ordinary oc= new Ordinary("ordinary", 18,10);
        lc.show();
        uc.show();
        oc.show();
    }

}
```

- Output

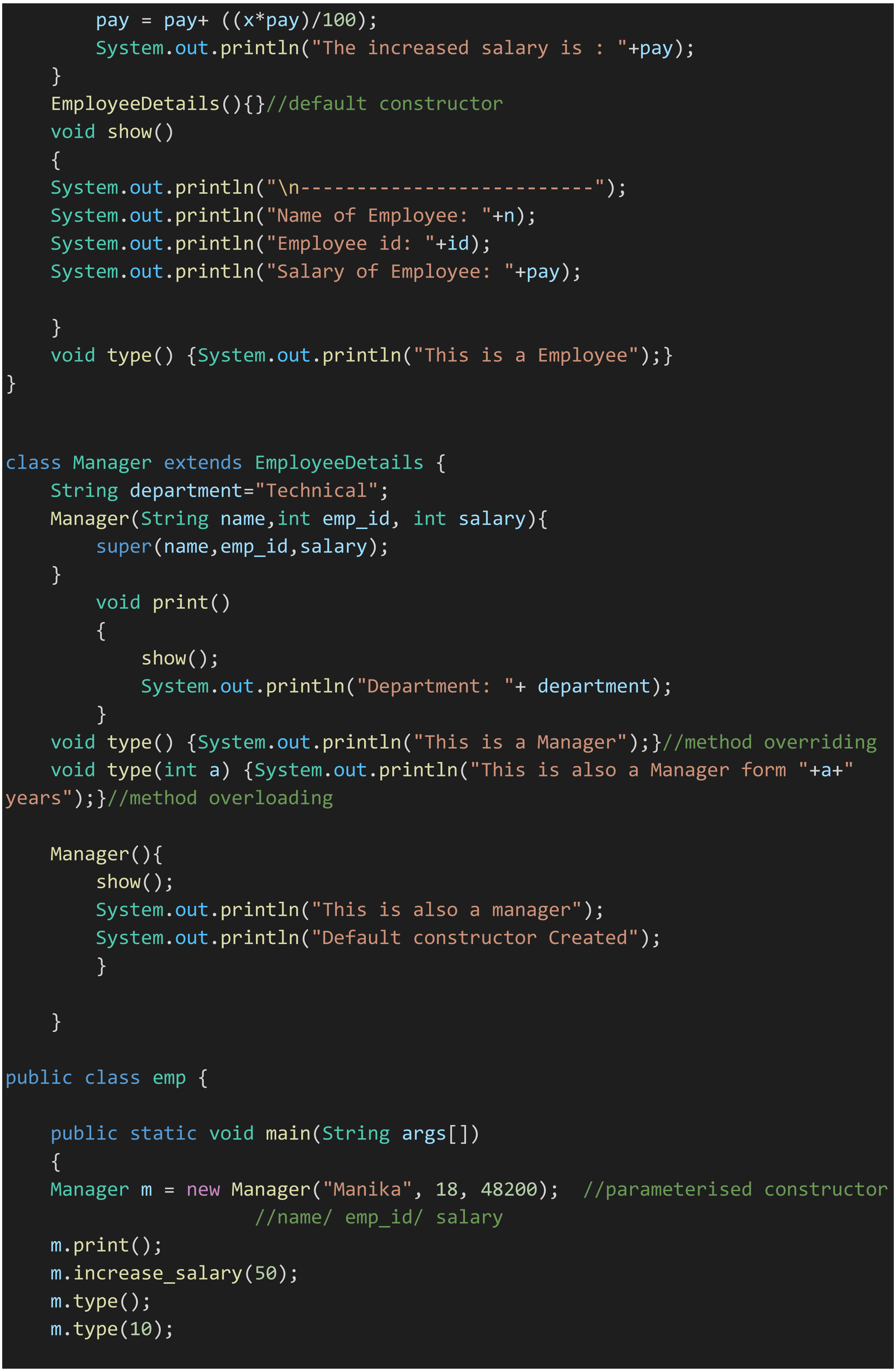


```
4b4fa68cea817a502fb25\redhat.java\jdt_ws\Session 3_96bcbbbf\bin' 'trunk'

-------------------------
Call Type: lightening
Time used: 14
Call Rate: 12 Rs/min
Total Cost: 168

-------------------------
Call Type: urgent
Time used: 15
Call Rate: 10 Rs/min
Total Cost: 150

-------------------------
Call Type: ordinary
Time used: 18
Call Rate: 10 Rs/min
Total Cost: 180
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>
```

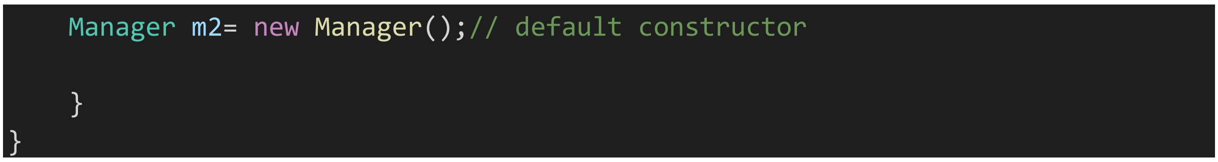
5. Design a class employee of an organization. An employee has a name, empid, and salary. Write the default constructor, a constructor with parameters (name, empid, and salary) and methods to return name and salary. Also write a method increaseSalary that raises the employee's salary by a certain user specified percentage. Derive a subclass Manager from employee. Add an instance variable named department to the manager class. Supply a test program that uses theses classes and methods.

- Code

```java
class EmployeeDetails {
    String n;
    int id;
    int pay;
    EmployeeDetails(String name,int emp_id, int salary)//parameterised
constructor
    { n=name; id=emp_id;pay=salary; }
    void increase_salary(int x) {
```

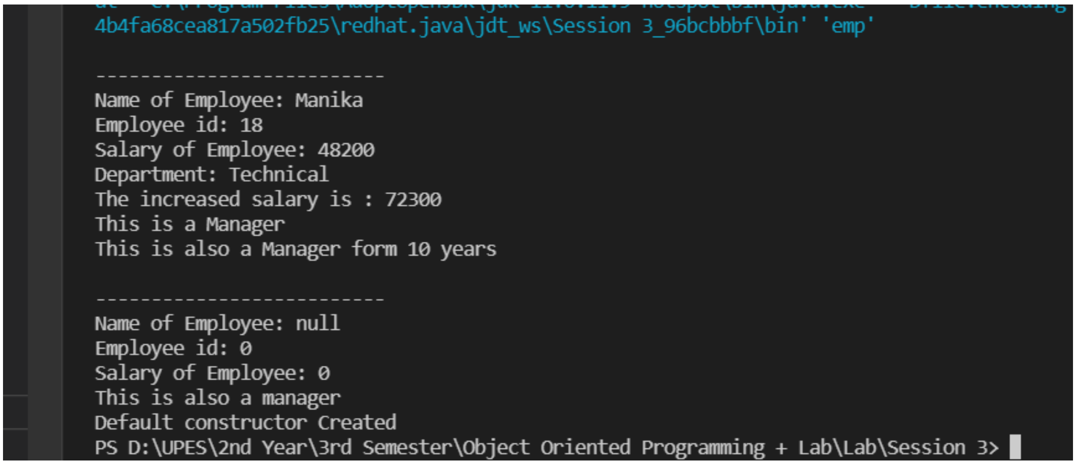


```
        pay = pay+ ((x*pay)/100);
        System.out.println("The increased salary is : "+pay);
    }
    EmployeeDetails(){}//default constructor
    void show()
    {
    System.out.println("\n--------------------------");
    System.out.println("Name of Employee: "+n);
    System.out.println("Employee id: "+id);
    System.out.println("Salary of Employee: "+pay);

    }
    void type() {System.out.println("This is a Employee");}
}


class Manager extends EmployeeDetails {
    String department="Technical";
    Manager(String name,int emp_id, int salary){
        super(name,emp_id,salary);
    }
        void print()
        {
            show();
            System.out.println("Department: "+ department);
        }
    void type() {System.out.println("This is a Manager");}//method overriding
    void type(int a) {System.out.println("This is also a Manager form "+a+"
years");}//method overloading

    Manager(){
        show();
        System.out.println("This is also a manager");
        System.out.println("Default constructor Created");
        }

    }

public class emp {

    public static void main(String args[])
    {
    Manager m = new Manager("Manika", 18, 48200);  //parameterised constructor
                      //name/ emp_id/ salary
    m.print();
    m.increase_salary(50);
    m.type();
    m.type(10);
```



```
    Manager m2= new Manager();// default constructor

    }
}
```

- Output



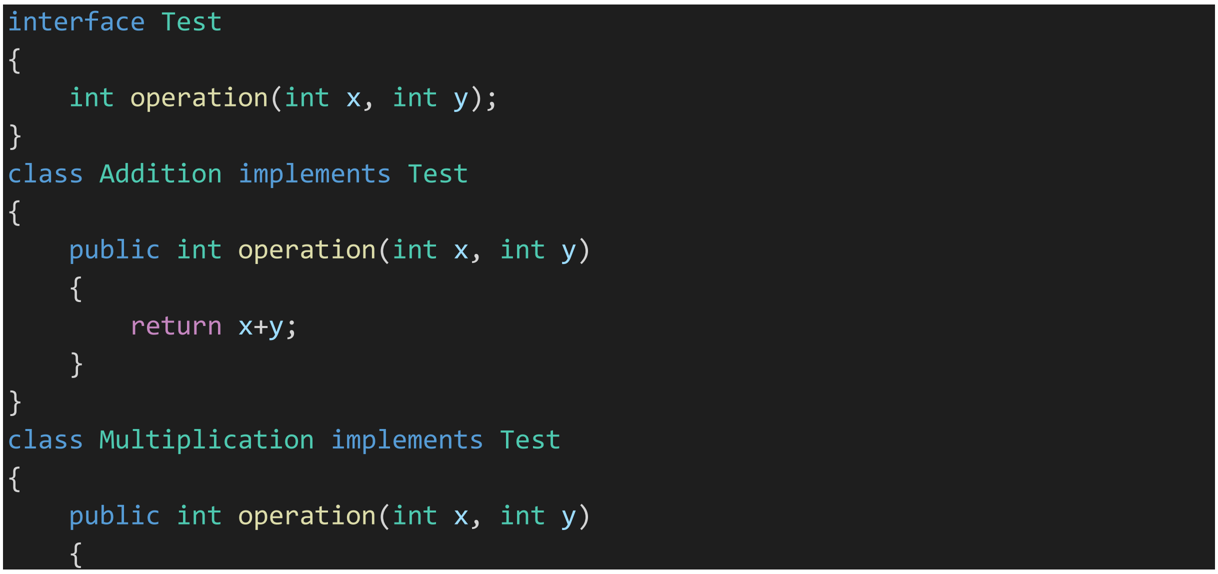
```
4b4fa68cea817a502fb25\redhat.java\jdt_ws\Session 3_96bcbbbf\bin' 'emp'

-------------------------
Name of Employee: Manika
Employee id: 18
Salary of Employee: 48200
Department: Technical
The increased salary is : 72300
This is a Manager
This is also a Manager form 10 years

-------------------------
Name of Employee: null
Employee id: 0
Salary of Employee: 0
This is also a manager
Default constructor Created
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3> ▌
```

❖ Experiment 5

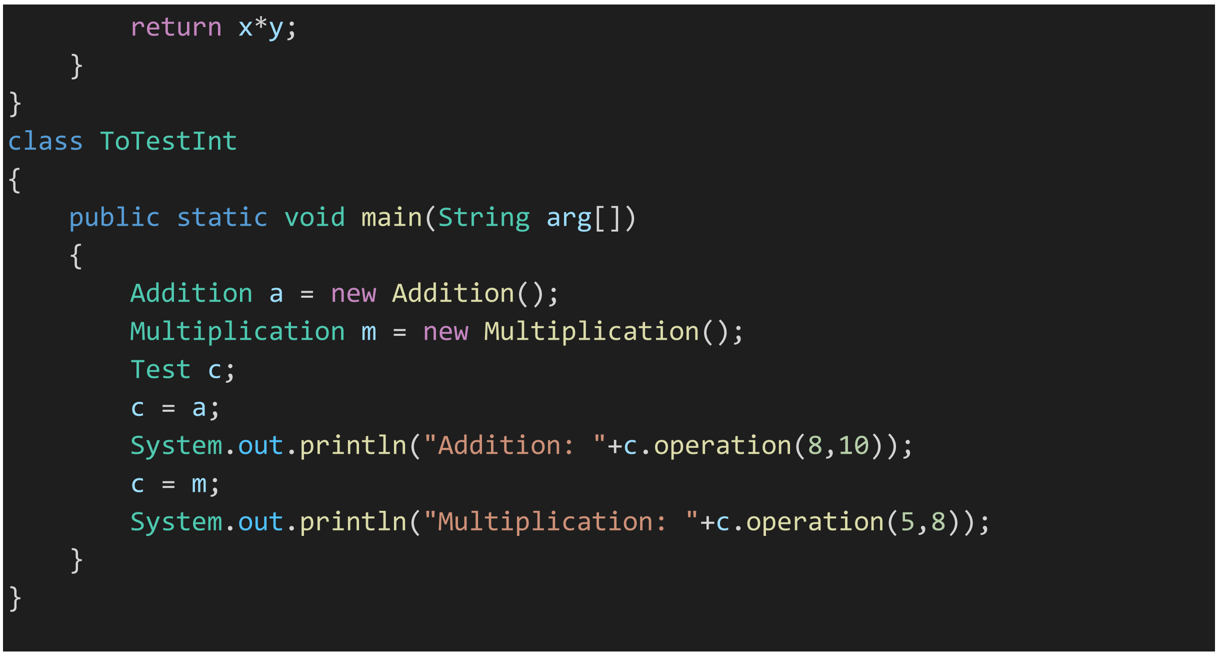
➢ Lab Exercise

1.Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt. In this class use the object of arithmetic class.

- Code

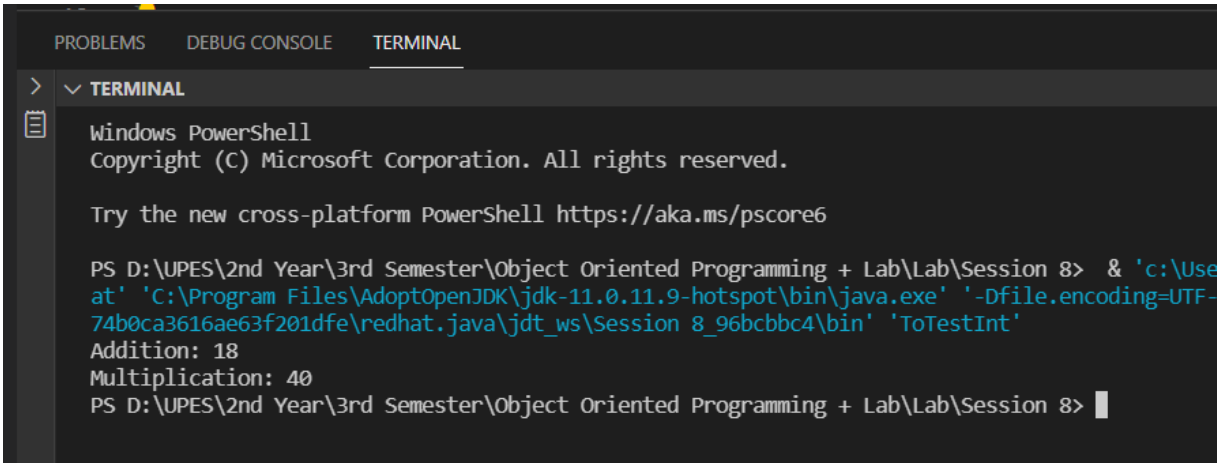


```
interface Test
{
    int operation(int x, int y);
}
class Addition implements Test
{
    public int operation(int x, int y)
    {
        return x+y;
    }
}
class Multiplication implements Test
{
    public int operation(int x, int y)
    {
```



```
        return x*y;
    }
}
class ToTestInt
{
    public static void main(String arg[])
    {
        Addition a = new Addition();
        Multiplication m = new Multiplication();
        Test c;
        c = a;
        System.out.println("Addition: "+c.operation(8,10));
        c = m;
        System.out.println("Multiplication: "+c.operation(5,8));
    }
}
```

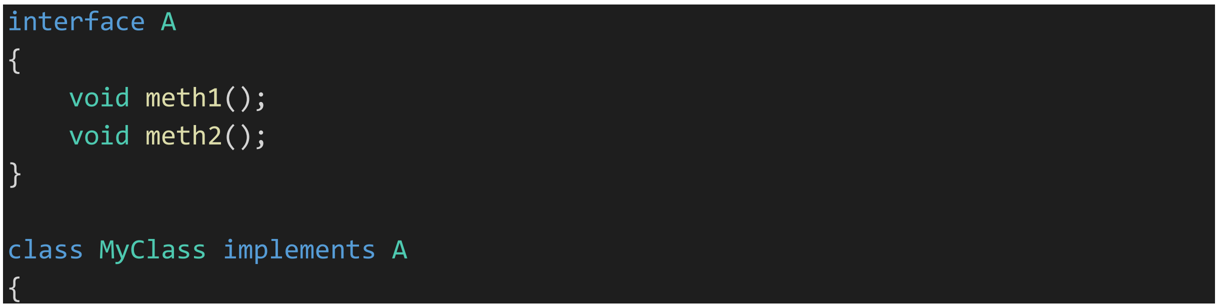
- Output



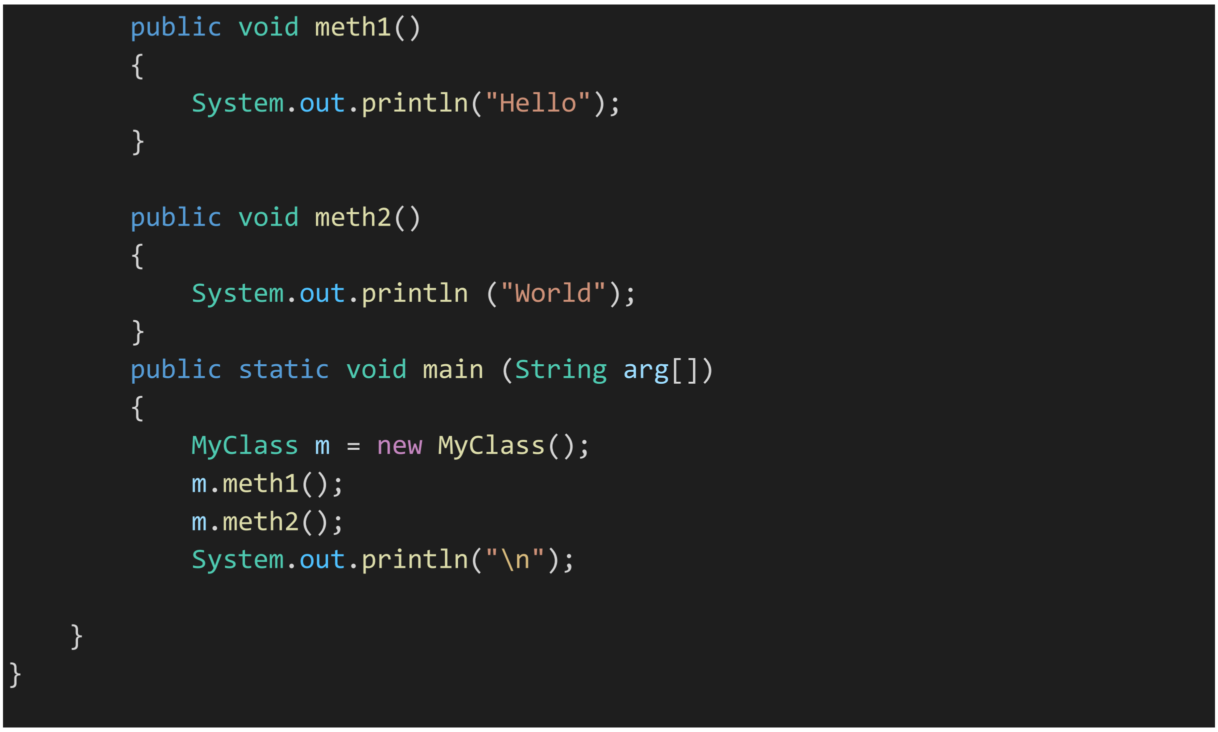
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8>  & 'c:\Use
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-
74b0ca3616ae63f201dfe\redhat.java\jdt_ws\Session 8_96bcbbc4\bin' 'ToTestInt'
Addition: 18
Multiplication: 40
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8> 
```

2. Write a program to create interface A, in this interface we have two method meth1 and meth2. Implements this interface in another class named MyClass.

- Code

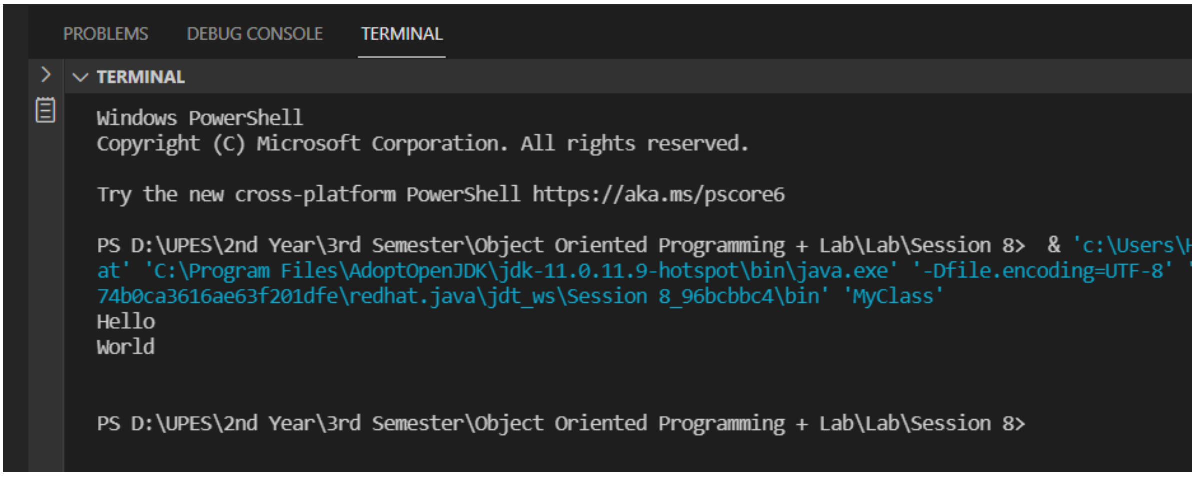


```
interface A
{
    void meth1();
    void meth2();
}

class MyClass implements A
{
```



```
        public void meth1()
        {
            System.out.println("Hello");
        }

        public void meth2()
        {
            System.out.println ("World");
        }
        public static void main (String arg[])
        {
            MyClass m = new MyClass();
            m.meth1();
            m.meth2();
            System.out.println("\n");

    }
}
```

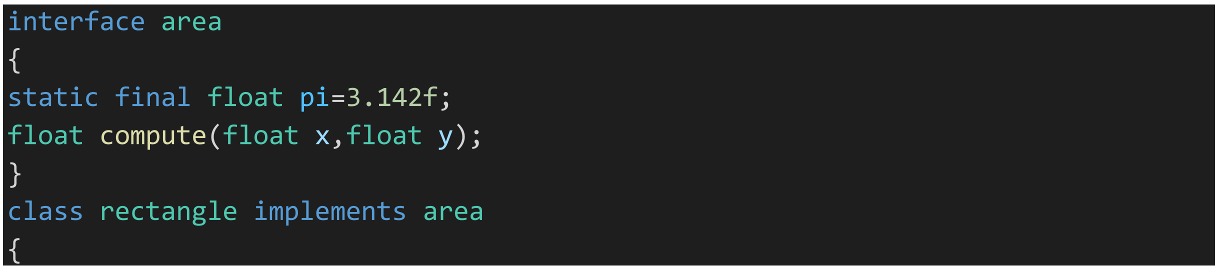
- Output



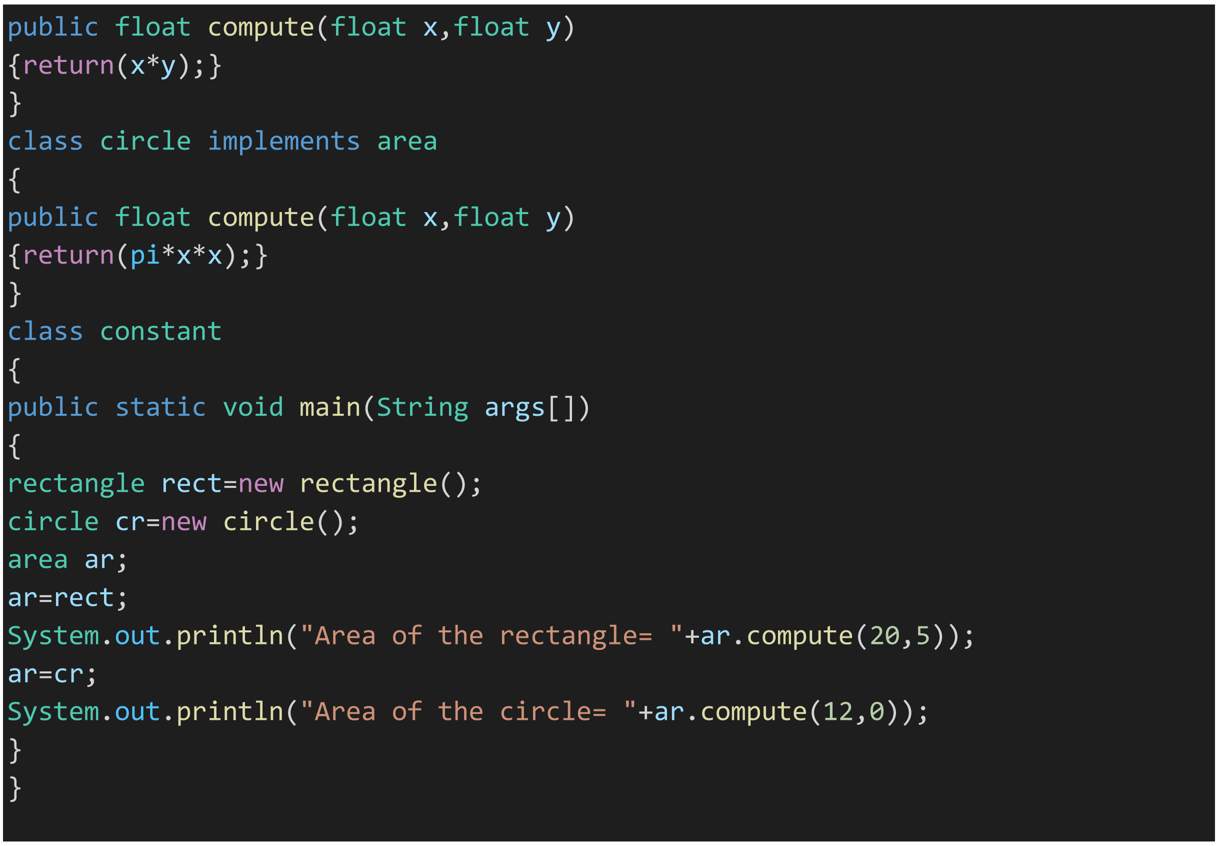
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8>  & 'c:\Users\
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF-8' '
74b0ca3616ae63f201dfe\redhat.java\jdt_ws\Session 8_96bcbbc4\bin' 'MyClass'
Hello
World


PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8>
```

3. Write a program in Java to show the usefulness of Interfaces as a place to keep constant value of the program.

- Code

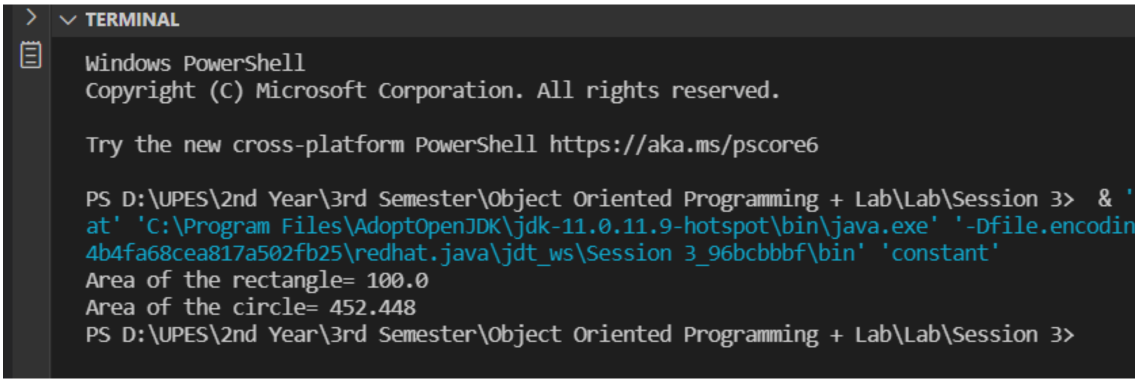


```
interface area
{
static final float pi=3.142f;
float compute(float x,float y);
}
class rectangle implements area
{
```



```
public float compute(float x,float y)
{return(x*y);}
}
class circle implements area
{
public float compute(float x,float y)
{return(pi*x*x);}
}
class constant
{
public static void main(String args[])
{
rectangle rect=new rectangle();
circle cr=new circle();
area ar;
ar=rect;
System.out.println("Area of the rectangle= "+ar.compute(20,5));
ar=cr;
System.out.println("Area of the circle= "+ar.compute(12,0));
}
}
```

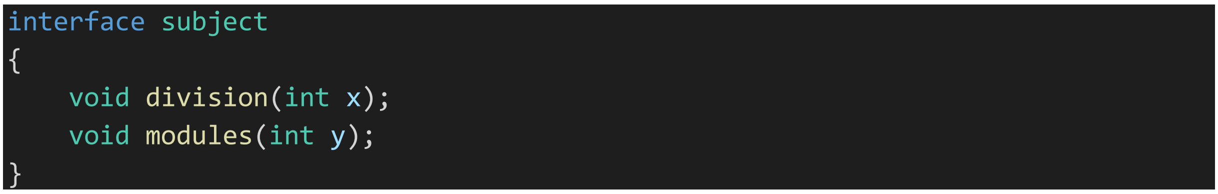
- Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>  & '
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encodin
4b4fa68cea817a502fb25\redhat.java\jdt_ws\Session 3_96bcbbbf\bin' 'constant'
Area of the rectangle= 100.0
Area of the circle= 452.448
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 3>
```

4.Write a program to create an Interface having two methods division and modules. Create a class, which overrides these methods.

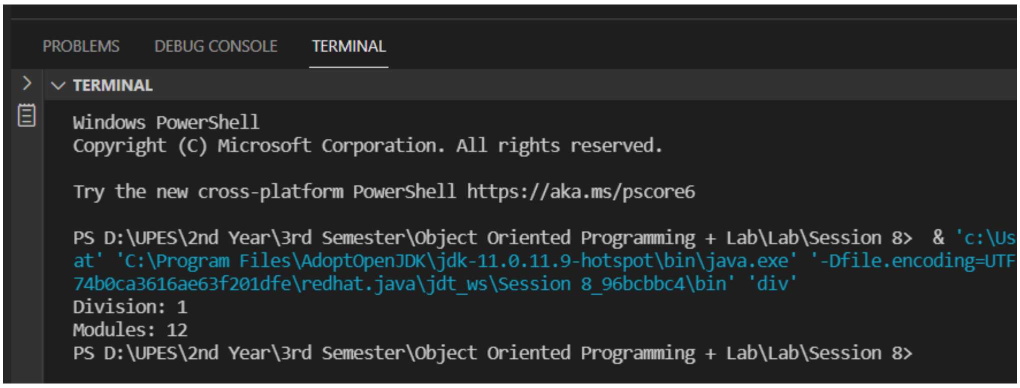
- Code



```
interface subject
{
    void division(int x);
    void modules(int y);
}
```

```
class sub implements subject
{
    int div, mod;
    public void division(int x)
    {
        div = x;
    }
    public void modules(int y)
    {
        mod = y;
    }
    void print(){
        System.out.println("Division: "+div);
        System.out.println("Modules: "+mod);
    }
}
class div
{
    public static void main(String arg[])
    {
        sub s = new sub();
        s.division(1);
        s.modules(12);
        s.print();
    }
}
```

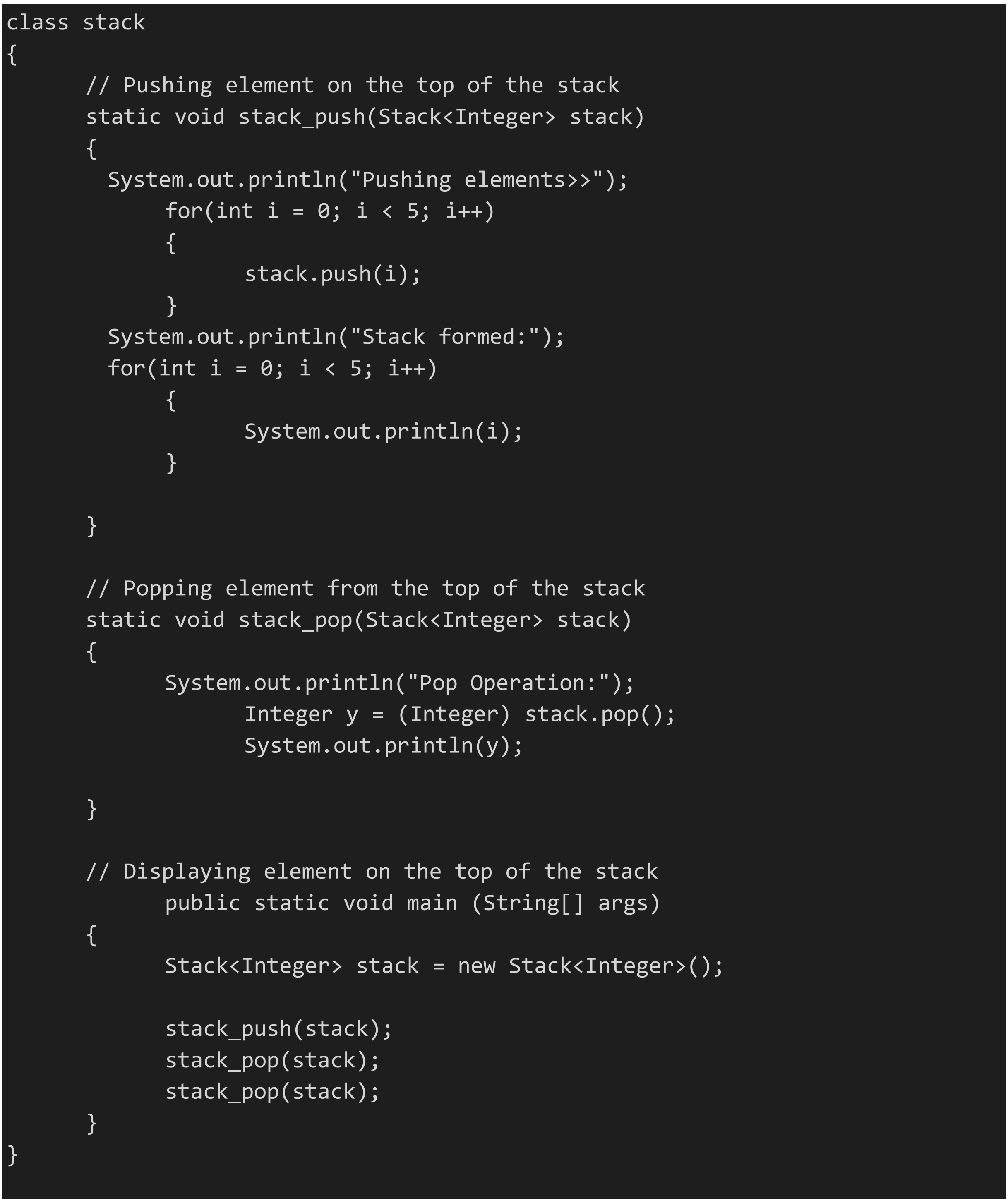
- Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8>  & 'c:\Us
at' 'C:\Program Files\AdoptOpenJDK\jdk-11.0.11.9-hotspot\bin\java.exe' '-Dfile.encoding=UTF
74b0ca3616ae63f201dfe\redhat.java\jdt_ws\Session 8_96bcbbc4\bin' 'div'
Division: 1
Modules: 12
PS D:\UPES\2nd Year\3rd Semester\Object Oriented Programming + Lab\Lab\Session 8>
```

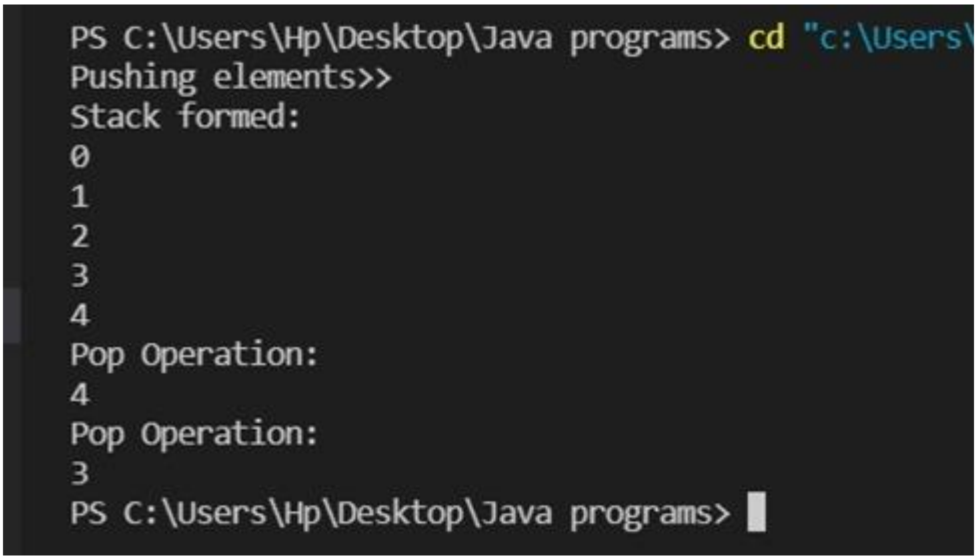
6. Write program to create an interface StackInterface having methods push (), pop () and display (). StackClass implements StackInterface. Class StackClass contains the main method which is having a switch case for selecting the particular operation of the stack.
    - Code

```
interface StackInterface
```



```
class stack
{
      // Pushing element on the top of the stack
      static void stack_push(Stack<Integer> stack)
      {
        System.out.println("Pushing elements>>");
            for(int i = 0; i < 5; i++)
            {
                  stack.push(i);
            }
        System.out.println("Stack formed:");
        for(int i = 0; i < 5; i++)
            {
                  System.out.println(i);
            }

      }

      // Popping element from the top of the stack
      static void stack_pop(Stack<Integer> stack)
      {
            System.out.println("Pop Operation:");
                  Integer y = (Integer) stack.pop();
                  System.out.println(y);

      }

      // Displaying element on the top of the stack
            public static void main (String[] args)
      {
            Stack<Integer> stack = new Stack<Integer>();

            stack_push(stack);
            stack_pop(stack);
            stack_pop(stack);
      }
}
```

- Output



```
PS C:\Users\Hp\Desktop\Java programs> cd "c:\Users\
Pushing elements>>
Stack formed:
0
1
2
3
4
Pop Operation:
4
Pop Operation:
3
PS C:\Users\Hp\Desktop\Java programs> █
```

❖ Experiment 6

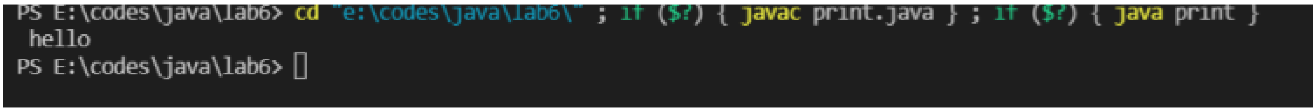
➢ Lab Exercise

1. Write a Java program to implement the concept of importing classes from user defined package and created packages.

- Code

```
package pack;
public class A
{
    public void msg()
    {
        System.out.println("Hello");
    }
}
import pack.*;
class B
{
    public static void main(String args[])
    {
        A obj = new A();
        obj.msg();
    }
}
```

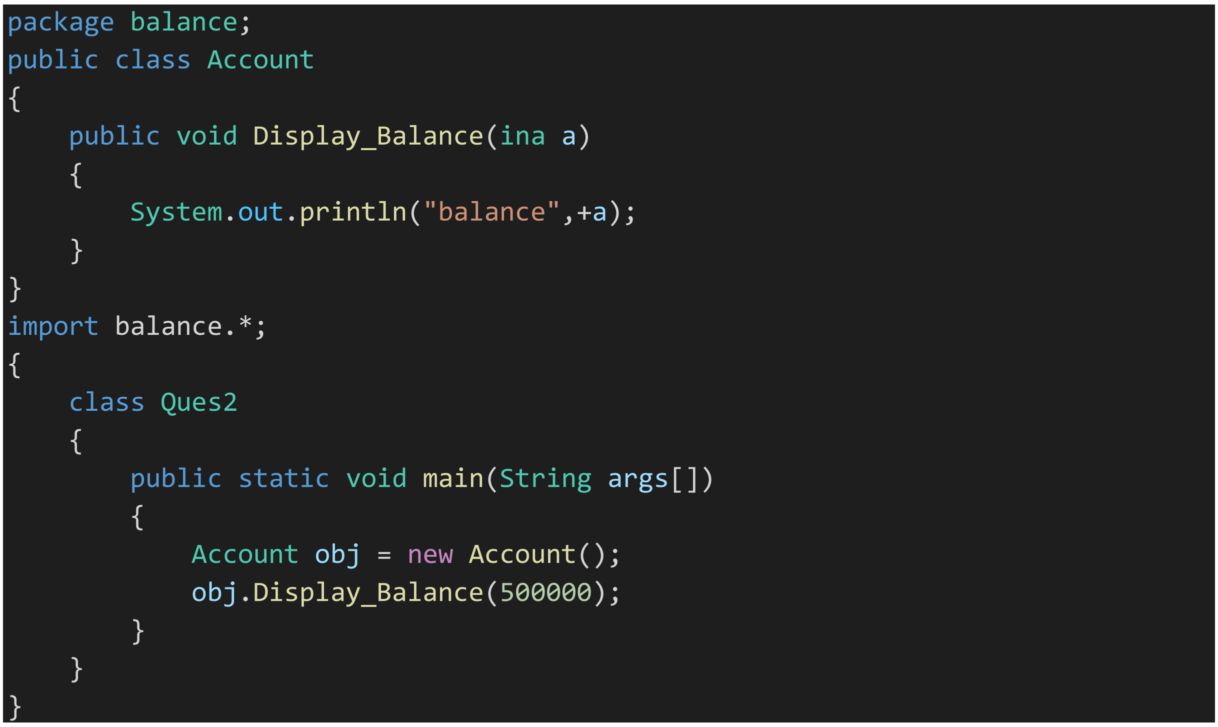
- Output



```
PS E:\codes\java\lab6> cd "e:\codes\java\lab6\" ; if ($?) { javac print.java } ; if ($?) { java print }
 hello
PS E:\codes\java\lab6> 
```

2. Write a program to make a package Balance. This has an Account class with Display_Balance method. Import Balance package in another program to access Display_Balance method of Account class.

    - Code



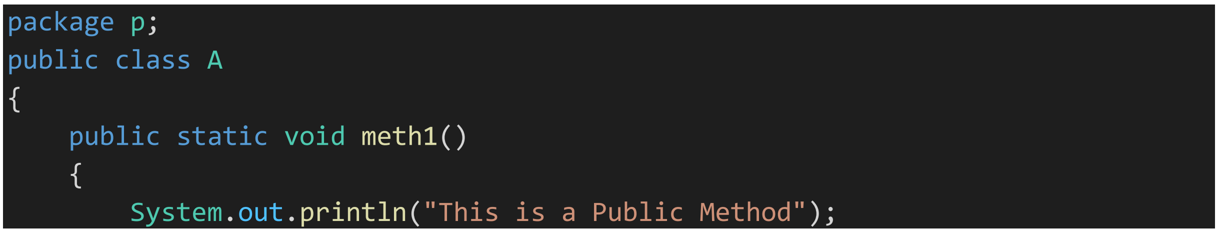
```
package balance;
public class Account
{
    public void Display_Balance(ina a)
    {
        System.out.println("balance",+a);
    }
}
import balance.*;
{
    class Ques2
    {
        public static void main(String args[])
        {
            Account obj = new Account();
            obj.Display_Balance(500000);
        }
    }
}
```

    - Output

```
 Account balance : 500000
PS E:\codes\java>
```

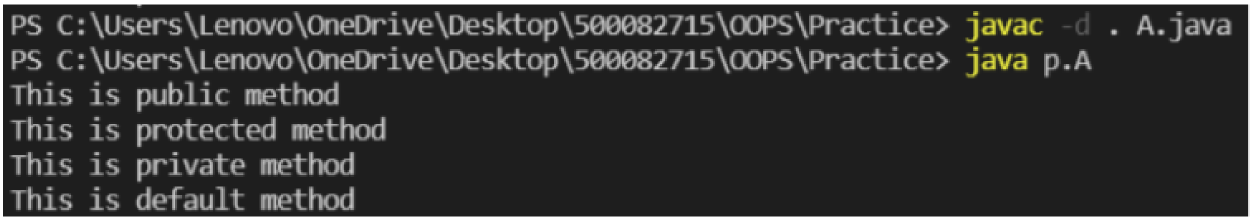
3. WAP to create a package p with class A with 4 types of access protected methods. How we will use these methods in different packages class i.e. there is main() in class B in package Q and 4 methods are in Class A in package p.

    - Code

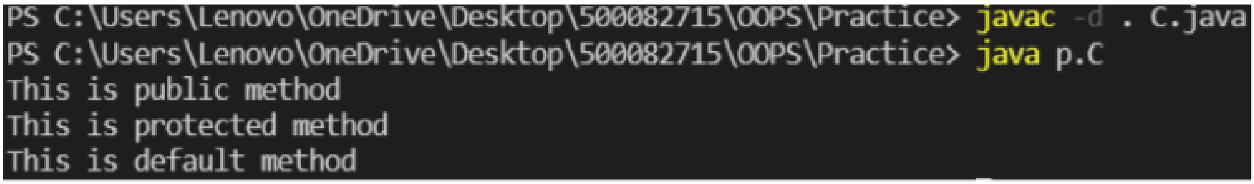


```
package p;
public class A
{
    public static void meth1()
    {
        System.out.println("This is a Public Method");
    }
    protected static void meth2()
    {
        System.out.println("This is a Protected Method");
    }
    private static void meth3()
    {
        System.out.println("This is a private Method");
    }
    static void meth4()
    {
        System.out.println("this is default method");
    }
    public static void main(String[] args)
    {
        A a = new A();
        A.meth1();
        A.meth2();
        A.meth3();
        A.meth4();
    }
}

public class B extends A
{
    public static void main(String[] args)
    {
        B b = new B();
        B.meth1();
        B.meth2();
        //B.meth3(); cannot access as private
        B.meth4();
    }
}
```

- Output



```
PS C:\Users\Lenovo\OneDrive\Desktop\500082715\OOPS\Practice> javac -d . A.java
PS C:\Users\Lenovo\OneDrive\Desktop\500082715\OOPS\Practice> java p.A
This is public method
This is protected method
This is private method
This is default method
```



```
PS C:\Users\Lenovo\OneDrive\Desktop\500082715\OOPS\Practice> javac -d . C.java
PS C:\Users\Lenovo\OneDrive\Desktop\500082715\OOPS\Practice> java p.C
This is public method
This is protected method
This is default method
```
