

# OBJECT ORIENTED PROGRAMMING LAB

**Submitted By**

**Name-** NIHARIKA AGRAWAL

**Roll. No-** R117219122

**SAP ID-** 500075359

**Branch-** B.Tech CSE AIML (2019-2023)

**Batch-** B4

**SEMESTER-** IV

**Submitted to**

Ahatsham Sir

School of Computer Science

**EXPERIMENT – 1**

**TITLE: -** Introduction to JAVA Environment.

1. Installation of JDK

In a browser, go to the Java SE Development Kit 10 Downloads page and click **Accept License Agreement**. Under the **Download** menu, click the **Download** link that corresponds to the .exe for your version of Windows. Download the file

jdk-10.*interim.update.patch*\_windows-x64\_bin.exe.

1. Setting of path and classpath

Write a Program in NOTEPAD.

Now Click on **File->Save As->Local Disk(D:)->JAVA PROGRAMS-> Test.java**

Now Open **C0MMAND PROMPT**

1. TEMPORARY PATH

**Write the commands:**

#java

#set path= C:\Program Files\Java\jdk1.8.0\_261\bin

#D:

#cd JAVA PROGRAMS

#javac Test.java

#java Test

**RESULT:** Hello JAVA

1. PERMANENT PATH

My PC-> Properties-> Advanced System Setting-> Environment Variables-> System Variables-> New..-> Type **path**-> Variable value-> **C:\Program Files\Java\jdk1.8.0\_261\bin** -> OK-> OK-> OK

And close the window

Now path is set Permanent and Now open **Command Prompt**

**Write the commands:**

#java

#D:

#cd JAVA PROGRAMS

#Javac Test.java

#java Test

**RESULT:** Hello JAVA

1. Introduction to Eclipse

Search for **eclipse.org** and click **Eclipse IDE for JAVA Developers.** Now Install it.

Launch Eclipse.

Go to File-> New..->Java Project-> Create->Project Name->Name

Modifiers as **public**

AND START DOING CODING

**EXPERIMENT - 2,3**

**TITLE: -** Basic Java Programming

1. Write a program to find the largest of 3 numbers.

public class Largest\_No{

public static void main(String[] args) {

int num1 = 8, num2 = 24, num3 = 29;

if( num1 >= num2 && num1 >= num3)

System.out.println(num1+" is the largest Number");

else if (num2 >= num1 && num2 >= num3)

System.out.println(num2+" is the largest Number");

else

System.out.println(num3+" is the largest Number");

}

}

OUTPUT:-



1. Write a program to add two number using command line arguments.

import java.lang.\*;

public class CommandLineArg

{

public static void main(String[] args)

{

int x,y,s=0;

x=Integer.parseInt(args[0]);

y=Integer.parseInt(args[1]);

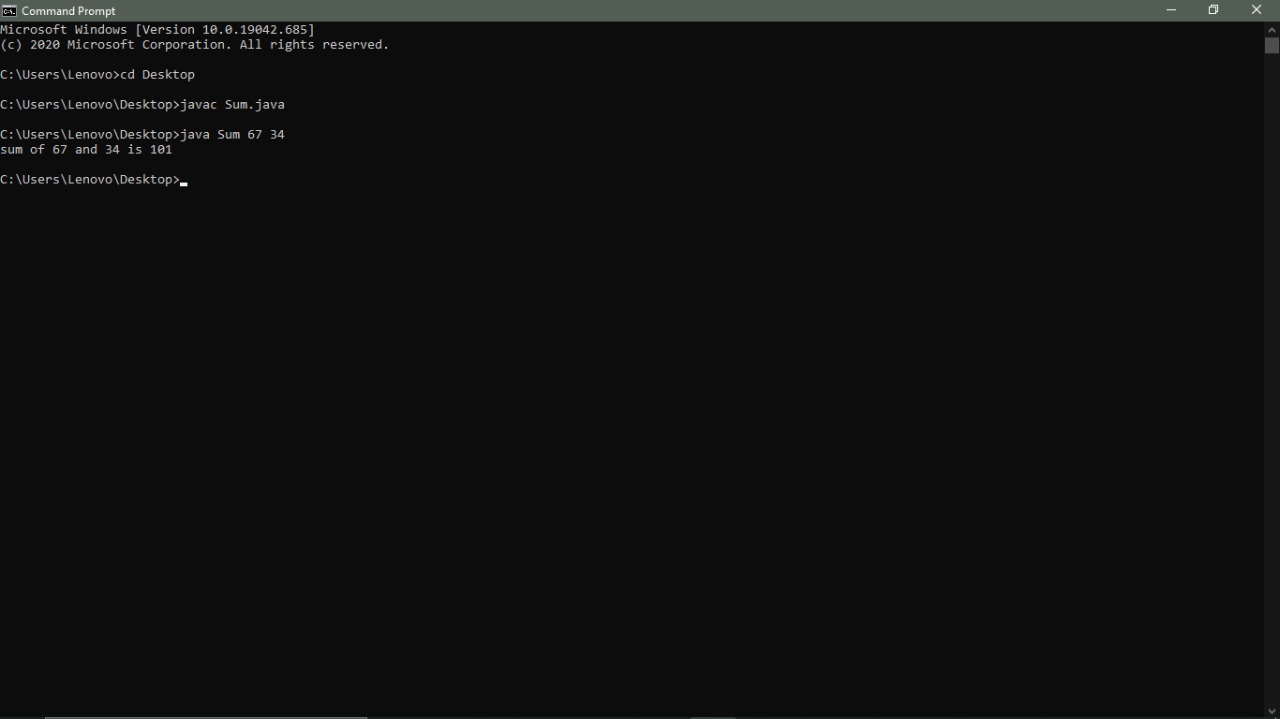
s=x+y;

System.out.println("sum of " + x + " and " + y +" is " +s);

}

}

OUTPUT:-



1. Write a program to print Fibonacci series using loop.

public class Fibonacci {

public static void main(String[] args) {

int count = 15, num1 = 0, num2 = 1;

System.out.print("Fibonacci Series of "+count+" numbers:");

for (int i = 1; i <= count; ++i)

{

System.out.print(num1+" ");

int sumOfPrevTwo = num1 + num2;

num1 = num2;

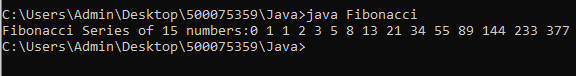
num2 = sumOfPrevTwo;

}

}

}

OUTPUT:-



1. Write a program to implement a command line calculator.

import java.util.Scanner;

public class Calculator {

public static void main(String[] args) {

Scanner reader = new Scanner(System.in);

System.out.print("Enter two numbers: ");

double first = reader.nextDouble();

double second = reader.nextDouble();

System.out.print("Enter an operator (+, -, \*, /): ");

char operator = reader.next().charAt(0);

double result;

switch (operator) {

case '+':

result = first + second;

break;

case '-':

result = first - second;

break;

case '\*':

result = first \* second;

break;

case '/':

result = first / second;

break;

default:

System.out.printf("Error! operator is not correct");

return;

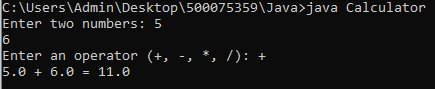
}

System.out.println(first + " " + operator + " " + second + " = " + result);

}

}

OUTPUT:-



1. Write a program using classes and object in java.

import java.util.\*;

public class Classandobject

{

int id;

double marks;

String name;

void student1(int i,double m,String n){

id=i;

marks=m;

name=n;

if(marks>60)

{

System.out.println("name is: "+ name);

System.out.println("Passed");

System.out.println("Student id is "+id);

}

else

{

System.out.println("name is: "+ name);

System.out.println("not Passed");

System.out.println("Student id is "+id);

}

}

public static void main(String[] args)

{

Classandobject obj=new Classandobject();

Scanner sc =new Scanner(System.in);

System.out.println("Enter the student's name");

String name=sc.nextLine();

System.out.println("Enter the student's id");

int id=sc.nextInt();

System.out.println("Enter the student's marks");

double marks=sc.nextDouble();

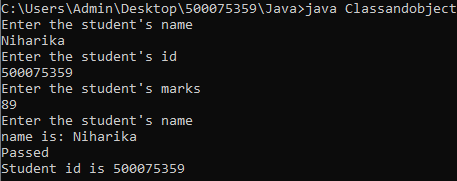
System.out.println("Enter the student's name");

obj.student1(id,marks,name);

}

}

OUTPUT:-



1. 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

import java.util.Scanner;

class Grades

{

public static void main( String [] args )

{

Scanner key = new Scanner(System.in);

System.out.println("Enter the number of students");

int n = key.nextInt();

Student student\_list[] = new Student[ n ];

System.out.println("Enter the name and score of --- ");

for( int i = 0 ; i < n ; i++ )

{

System.out.println("Student - " + (i+1));

String name = key.next();

int score = key.nextInt();

student\_list[i] = new Student( name , score );

}

for( int i = 0 ; i < n - 1 ; i++ )

{

for( int j = 0 ; j < n - i - 1 ; j++ )

{

if( student\_list[j+1].score < student\_list[j].score )

{

Student temp = student\_list[j+1];

student\_list[j+1] = student\_list[j];

student\_list[j] = temp ;

}

}

}

System.out.print("\nSorted List ---> \n" );

for( int i = 0 ; i < n ; i++ )

{

student\_list[i].printInformation();

}

String status[]= { "Pass" , "Merit" , "Distinction" };

int current = 0 ;

System.out.print("\nFail - ");

for( int i = 0 ; i < n ; i++ )

{

if( (student\_list[i].score >=40 && current == 0) || (student\_list[i].score >=51 && current == 1 ) ||

(student\_list[i].score >=75 && current == 2))

System.out.print("\n" + status[current++] + " - ");

System.out.print(" " + student\_list[i].name);

}

System.out.println();

}

}

class Student

{

int score ;

String name;

public Student( String name , int score )

{

this.score = score ;

this.name = name;

}

void printInformation ()

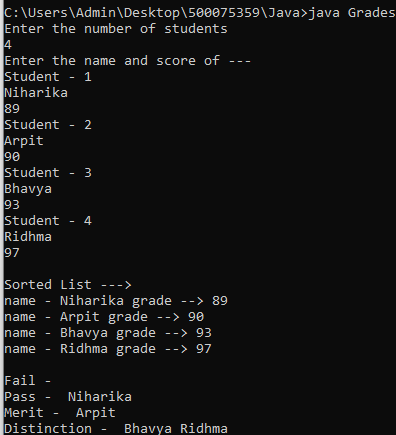
{

System.out.println("name - " + this.name + " grade --> "+ this.score);

}

}

Output:-



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.)

import java.util.\*;

import java.io.\*;

public class Combination {

public static void main(String[] args) {

int arr[]=new int[3];

Scanner sc=new Scanner(System.in);

System.out.println("Enter the 3 numbers of which possible combinations we want to generate");

for(int i=0;i<3;i++) {

arr[i]=sc.nextInt();

}

for(int i=0;i<3;i++) {

for(int j=0;j<3;j++) {

for(int k=0;k<3;k++) {

if(i!=j&&j!=k&&k!=i) {

System.out.println(arr[i]+""+arr[j]+""+arr[k]);

}

}

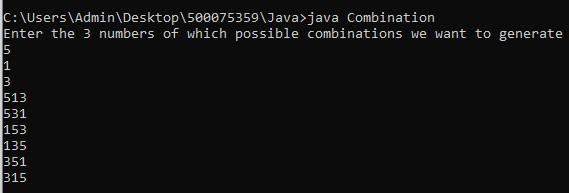
}

}

}

}

OUTPUT:-



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

import java.util.Scanner;

class SumOfSquares

{

public static void main(String[] args)

{

int n=10,sum=0;

Scanner s = new Scanner(System.in);

int a[] = new int[n];

for(int i=0; i <n; i++){

a[i] = s.nextInt();

sum = sum + (a[i] \* a[i]);

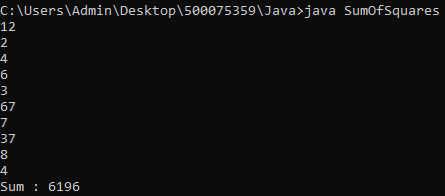
}

System.out.println("Sum : "+sum);

}

}

OUTPUT:-



1. 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.)

import java.util.\*;

import java.io.\*;

public class Month {

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

String month[]= {"January","February","March","April","May","June","July","August","September","October","November","December"};

System.out.println("Enter the month number:");

int num=sc.nextInt();

if(num>=1&&num<=12) {

System.out.println(num+" is "+month[num-1]);}

else {

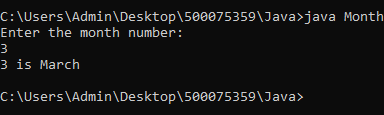
System.out.println("Entered a wrong number to print the equivalent month");

}

}

}

OUTPUT:-



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

import java.util.\*;

import java.io.\*;

public class Sum {

public static void main(String[] args) {

int sum=0;

for(int i=41;i<250;i++)

{

if(i%5==0)

sum+=i;

}

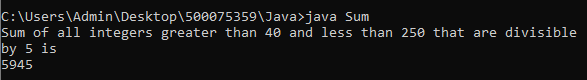
System.out.println("Sum of all integers greater than 40 and less than 250 that are divisible\r\n" +

"by 5 is\n"+sum);

}

}

OUTPUT:-



**EXPERIMENT – 4**

**TITLE: -** Inheritance

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

public class R

{

public static void main(String args[])

{

class\_room cr=new class\_room(10,20,15);

int a1=cr.area();

int v1=cr.volume();

System.out.println("Area of Room : "+a1);

System.out.println("Volume of Room : "+v1);

}

}

class room

{

private int l,b;

room(int x,int y)

{ l=x; b=y;}

int area()

{return(l\*b);

}

}

class class\_room extends room

{

int h;

class\_room(int x,int y,int z)

{

super(x,y);

h=z;

}

int volume()

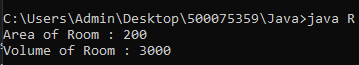
{

return(area()\*h);

}

}

OUTPUT:-



1. Write a program in Java to create a Player class. Inherit the classes Cricket \_Player, Football \_Player and Hockey\_ Player from Player class.

public class c {

public static void main(String args[])

{

criket\_player c=new criket\_player("MS Dhoni","criket",39);

football\_player f=new football\_player("Ronaldo","foot ball",36);

hockey\_player h=new hockey\_player("Manpreet","hockey",28);

c.show();

f.show();

h.show();

}

}

class Player

{

String name;

int age;

Player(String n,int a)

{ name=n; age=a; }

void show()

{

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

System.out.println("Player name : "+name);

System.out.println("Age : "+age);

}

}

class criket\_player extends Player

{

String type;

criket\_player(String n,String t,int a)

{

super(n,a);

type=t;

}

public void show()

{

super.show();

System.out.println("Player type : "+type);

}

}

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);

}

}

class hockey\_player extends Player

{

String type;

hockey\_player(String n,String t,int a)

{

super(n,a);

type=t;

}

public void show()

{

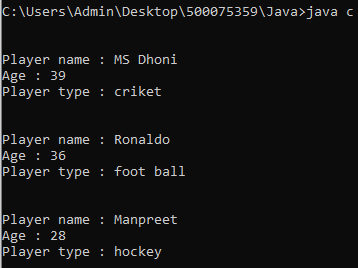
super.show();

System.out.println("Player type : "+type);

}

}

OUTPUT:-



1. 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.

class W

{

public static void main(String args[])

{

dailyworker d=new dailyworker(250,"Ram",75);

salariedworker s=new salariedworker(666,"shyam",100);

d.compay(45);

s.compay();

}

}

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)

{

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()

{

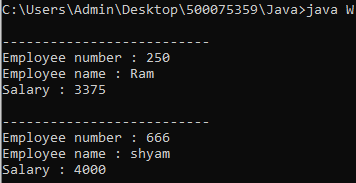
show();

System.out.println("Salary : "+rate\*hour);

}

}

OUTPUT:-



1. 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.

import java.util.Scanner;

class Telephone{

int callnumber;

String calltype;

Telephone(int c,String s){

callnumber = c;

calltype = s;

}

void show() {

System.out.println("call number"+" "+callnumber);

System.out.println("call type"+" "+calltype);

}

}

class Ordinary extends Telephone{

float cost;

Ordinary(int c,String s,float co){

super(c,s);

this.cost = co;

}

void charge(double time) {

super.show();//calling show() of parent class

System.out.println("call charges"+" "+cost\*time);

}

}

class Urgent extends Telephone{

float cost;

Urgent(int c,String s,float co){

super(c,s);

this.cost = co;

}

void charge(double time) {

super.show();

System.out.println("call charges"+" "+cost\*time);

}

}

class Lightening extends Telephone{

float cost;

Lightening(int c,String s,float co){

super(c,s);

this.cost = co;

}

void charge(double time) {

super.show();

System.out.println("call charges"+" "+cost\*time);

}

}

public class Exp\_4\_4 {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

int a,no;

double t;

System.out.println("Enter 1 for ordinary call,2 for urgent call or 3 for lightening call");

a=sc.nextInt();

switch(a) {

case 1:

System.out.println("Enter the number to which call is to be made");

no=sc.nextInt();

Ordinary obj1 = new Ordinary(no,"ordinary call",3);

System.out.println("Enter the duration of the call");

t=sc.nextDouble();

obj1.charge(t);

break;

case 2:

System.out.println("Enter the number");

no=sc.nextInt();

Urgent obj2=new Urgent(no,"Urgent call",4);

System.out.println("Enter the duration of the call");

t=sc.nextDouble();

obj2.charge(t);

break;

case 3:

System.out.println("Enter the number");

no=sc.nextInt();

Lightening obj3 = new Lightening(no,"Lightening call",5);

System.out.println("Enter the duration of the call");

t=sc.nextDouble();

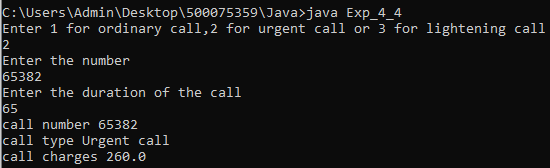
obj3.charge(t);

}

}

}

OUTPUT:-



1. 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.

import java.util.Scanner;

class Employee{

int empid;

String name;

double salary;

Employee(){

System.out.println("This is employee default constructor");

}

Employee(int emid,String name,double salary){

this.empid = emid;

this.name = name;

this.salary = salary;

}

String name()

{

return name;

}

double salary()

{

return salary;

}

void increase\_salary(int i) {

double x;

x=((salary\*i)/100)+salary;

System.out.print("Increased salary ="+" "+x);

}

}

class Manager extends Employee{

String department;

Manager(int e,String n,double sal,String d){

super(e,n,sal);

department = d;

}

void display() {

Scanner sc=new Scanner(System.in);

System.out.println("Name of the employee"+" "+super.name());

System.out.println("salary of a employee"+" "+super.salary());

System.out.print("department is"+ " "+ department+"\n");

System.out.println("Enter the percentage by which a salary is to be increased");

int ps=sc.nextInt();

super.increase\_salary(ps);

}

}

public class Exp\_4\_5 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the name of the employee");

String name=sc.nextLine();

System.out.println("Enter the id of the employee");

int id=sc.nextInt();

System.out.println("Enter the salary of the employee");

double sal=sc.nextDouble();

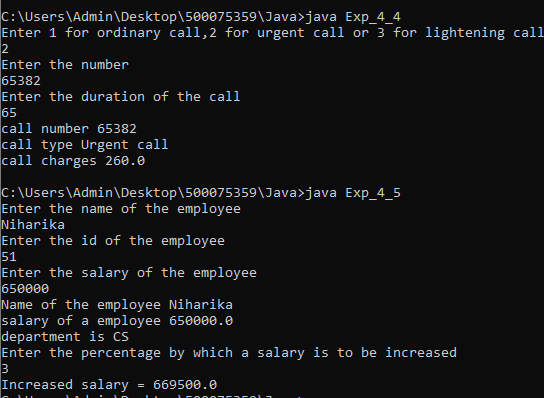
Manager M = new Manager(id,name,sal,"CS");

M.display();

}

}

OUTPUT:-



**EXPERIMENT – 5**

**TITLE: -** Interface

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.

interface test

{

int square();

}

class arithmetic implements test

{

int b;

arithmetic(int x)

{

b = x;

}

public int square()

{

return (b\*b);

}

}

class ToTestInt

{

public int return\_ans(int x)

{

arithmetic a=new arithmetic(x);

return a.square();

}

}

class main\_class

{

public static void main(String []args)

{

ToTestInt x= new ToTestInt();

System.out.println("\nThe square of 64 is "+x.return\_ans(64));

}

}

OUTPUT:-



1. 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.

interface A

{

void meth1();

void meth2();

}

interface B extends A

{

void meth3();

}

class MyClass implements B

{

public void meth1 ( )

{

System.out.println("Implement meth1().");

}

public void meth2()

{

System.out.println ("Implement meth2().");

}

public void meth3()

{

System.out.println ("Implement meth()." );

}

}

class IFExtend

{

public static void main(String arg[])

{

MyClass ob = new MyClass();

ob.meth1();

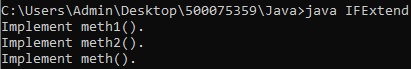
ob.meth2();

ob.meth3();

}

}

OUTPUT:-



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

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 Inter

{

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(10,20));

ar=cr;

System.out.println("Area of the circle= "+ar.compute(10,0));

}

}

OUTPUT:-



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

interface course

{

void division(int a);

void modules(int b);

}

class stud implements course

{

String name;

int div,mod;

void name(String n)

{

name=n;

}

public void division(int a)

{

div=a;

}

public void modules(int b)

{

mod=b;

}

void disp()

{

System.out.println("Name :"+name);

System.out.println("Division :"+div);

System.out.println("Modules :"+mod);

}

}

class Division

{

public static void main(String args[])

{

stud s=new stud();

s.name("Arun");

s.division(5);

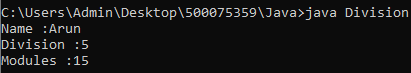
s.modules(15);

s.disp();

}

}

OUTPUT:-



**EXPERIMENT – 6**

**TITLE: -** Package

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

* **SIMPLE CLASS**

package mypack ;

public class Simple {

public static void demo()

{

System.out.println("Method of Simple class") ;

}

public static void main(String[] args)

{

System.out.println("simple class");

}

}

* **MyClass CLASS**

import mypack.Simple

public class MyClass {

public static void main(String[] args) {

System.out.println("Welcome to package");

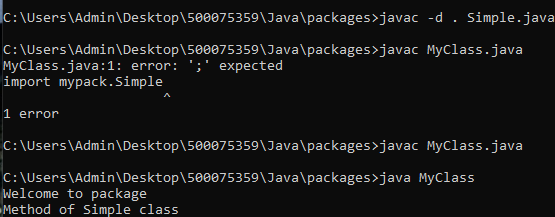
Simple s = new Simple() ;

s.demo() ;

}

}

OUTPUT:-



1. 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.

* **ACCOUNT CLASS**

package balance;

public class Account{

double balance = 200.45 ;

public void Display\_Balance()

{

System.out.println("The balance is : " + balance);

}

public static void main(String args[])

{

System.out.println("Account Class");

}

}

* **BALANCE CLASS**

import balance.Account;

class Balance

{

public static void main(String args[])

{

System.out.println("Balance Class");

Account obj = new Account();

obj.Display\_Balance();

}

}

OUTPUT:-

