

NAME : AVISHKAAR PAWAR

ROLL NO : AD-1224

Course : B.Sc(H) CS

Subject : Programming in Java

Batch : 2021-2024

Submitted to : Mr Mahesh Kumar Bhandari

LAB 0

//Practise Set Number 1

//Run this in terminal , not Code Runner

/\*\* Pass command line arguements in form java <name> < rollno> <course> <college> \*/

public class Student{

public static void main(String args[]){

// Name,RollNo,Course,college

Student stu1 = new Student(args);

stu1.display();

}

public void display(){

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

System.out.println("Roll NO is "+this.rollno);

System.out.println("Marks Scored : "+this.course);

System.out.println("College : "+this.college);

}

public String name;

public String course;

public int rollno;

public String college;

public Student(String args[]){

name=args[0];

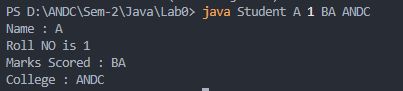
rollno=Integer.parseInt(args[1]);

course=args[2];

college=args[3];

}

}



//Practise Set 2

import java.util.Scanner;

public class Leap\_Year {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter Year : ");

int year = sc.nextInt();

if ((year % 400 != 0) && (year % 100 == 0)) {

System.out.println("Not a leap year ");

} else if (year % 400 == 0) {

System.out.println("Leap Year");

} else if (year % 4 == 0) {

System.out.println("Leap Year");

} else {

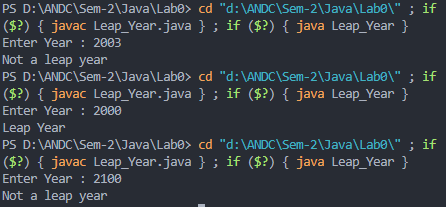
System.out.println("Not a leap year ");

}

sc.close();

}

}



// Practise Set 3

// Sum of N Command Line Arguement

public class SumOfN {

// Command Line Arguement

public static void main(String[] args) {

int sum = 0;

for (int i = 0; i < args.length; i++) {

sum += Integer.parseInt(args[i]);

}

System.out.println("\nThe sum is : " + sum + "\n");

}

}



import java.util.Scanner;

//Practise Set 4

public class Factorial {

public static void main(String [] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter number to find factorial of ");

int num=sc.nextInt();

int fact=1;

for(int i=1;i<=num;i++){

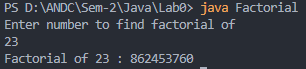
fact\*=i; //fact = fact\*i

}

System.out.println("Factorial of "+num+" : "+fact);

}

}



// Practise Set 5

public class Pattern {

public static void main(String[] args) {

int x = 1;

for (int i = 0; i < 4; i++) {

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

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

x++;

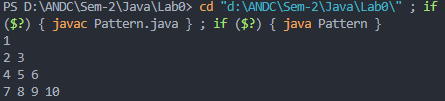
}

System.out.println();

}

}

}



LAB 1

//Lab Exercise No 1

public class ComplexNum {

float real, imag;

// constructors

ComplexNum() {

this.real = 0;

this.imag = 0;

}

ComplexNum(float x, float y) {

this.real = x;

this.imag = y;

}

public ComplexNum add(ComplexNum c2) {

ComplexNum c = new ComplexNum();

c.real = this.real + c2.real;

c.imag = this.imag + c2.imag;

return c;

}

public ComplexNum mult(ComplexNum c2) {

ComplexNum c = new ComplexNum();

c.real = (this.real) \* (c2.real) - (this.imag) \* (c2.imag);

c.imag = (this.real) \* (c2.imag) + (this.imag) \* (c2.real);

return c;

}

public void ToString() {

System.out.print("ComplexNum number is : ");

System.out.println(this.real + " + " + this.imag + "i");

}

public static void main(String[] args) {

ComplexNum c1 = new ComplexNum(3, 4);

c1.ToString();

ComplexNum c2 = new ComplexNum(2, 3);

c2.ToString();

ComplexNum sum1 = new ComplexNum();

sum1 = c1.add(c2);

System.out.print("Sum of ");

sum1.ToString();

ComplexNum prod = new ComplexNum();

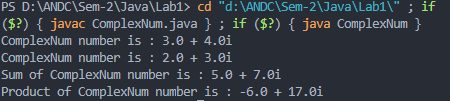
prod = c1.mult(c2);

System.out.print("Product of ");

prod.ToString();

}

}



// Practise set 1

import java.util.Scanner;

public class ScannerExample {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter your name: ");

String name = scan.nextLine();

System.out.println("Enter your gender: ");

char gender = scan.next().charAt(0);

System.out.println("Enter your age: ");

int age = scan.nextInt();

System.out.println("Enter your mobile no: : ");

long mobileNo = scan.nextLong();

System.out.println("Enter your CGPA: ");

double cgpa = scan.nextDouble();

System.out.println("Name: " + name + "\nGender:" + gender + "\nAge: " + age + "\nCGPA: " + cgpa);

}

}

// Practise set 1

import java.util.Scanner;

public class ScannerExample {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter your name: ");

String name = scan.nextLine();

System.out.println("Enter your gender: ");

char gender = scan.next().charAt(0);

System.out.println("Enter your age: ");

int age = scan.nextInt();

System.out.println("Enter your mobile no: : ");

long mobileNo = scan.nextLong();

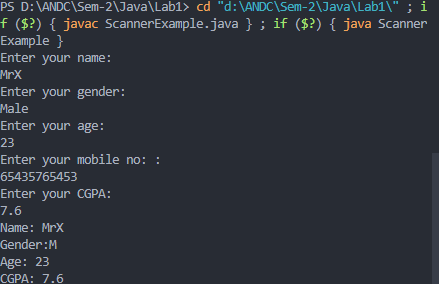
System.out.println("Enter your CGPA: ");

double cgpa = scan.nextDouble();

System.out.println("Name: " + name + "\nGender:" + gender + "\nAge: " + age + "\nCGPA: " + cgpa);

}

}



//Practise Set 2

public class Nth {

static int[] bubbleSort(int[] arr) {

int n = arr.length;

int temp = 0;

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

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

if (arr[j - 1] > arr[j]) {

// swap elements

temp = arr[j - 1];

arr[j - 1] = arr[j];

arr[j] = temp;

}

}

}

return arr;

}

public static void main(String[] args) {

// I will use CLA as size of array

int[] arr = { 4, 5, 1, 2, 8, 6, 7, 9, 88, 92, 3 };

arr = bubbleSort(arr);

int n = Integer.parseInt(args[0]);

System.out.println("Nth smallest Element " + arr[n - 1]);

System.out.println("Nth Largest Element " + arr[arr.length - n]);

// Here I will print element count from 1 , not 0

}

}



// Practise Set 3

public class SumAtOdd {

public static void main(String [] args){

int[] arr={2,3,7,1,86,99,22,75};

int sum=0;

for (int a=0;a<arr.length;a++){

if ((a%2)==0){

if (arr[a]%2!=0){

sum+=arr[a];

}

}

}

System.out.println("Sum of all odd numbers at even index is "+sum);

}

}



LAB 2

//Practical Set 1

public class Mobile {

String name;

int price;

String model;

Mobile() {

name = "DummyName";

price = 0;

model = "Alpha";

}

Mobile(String name, int price, String model) {

this.name = name;

this.price = price;

this.model = model;

}

void setInfo(String name, int price, String model) {

this.name = name;

this.price = price;

this.model = model;

}

void show() {

System.out.println("Name : " + name + "\nPrice : " + price + "\nModel : " + model + "\n");

}

public static void main(String[] args) {

Mobile m1 = new Mobile();

// 1- Using setter method

m1.setInfo("Vivo", 23000, "X23Pro");

// 2- Constructor with No Arguement

Mobile m2 = new Mobile();

// 3- Constructor with values

Mobile m3 = new Mobile("Samsung", 17000, "M30");

Mobile m4 = new Mobile();

// 4- Setting values one by one

m4.name = "Iphone";

m4.price = 130000;

m4.model = "13Pro[256GB]";

m1.show();

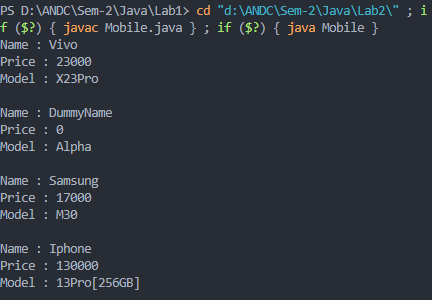
m2.show();

m3.show();

m4.show();

}

}



[//Practise](file:///\\Practise) set 2

public class OverloadingDemo {

void display() {

System.out.println("Hello ");

}

void display(String str) {

System.out.println("Hello " + str);

}

public static void main(String[] args) {

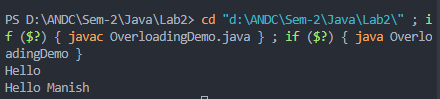
OverloadingDemo dem = new OverloadingDemo();

dem.display();

dem.display("Manish");

}

}



public class TypeConversionAuto {

void show(int a){

System.out.println(a);

}

void show(double a,int b){

System.out.println(a);

}

public static void main(String[] args){

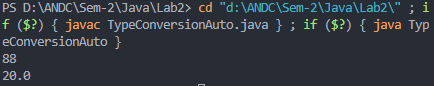
TypeConversionAuto obj=new TypeConversionAuto();

obj.show(88);

obj.show(20,1);//See this this int 20 converts to 20.0

}

}



//Practise Set 4

public class Constructor {

String name;

int age;

Constructor() {

name = "Default";

age = 0;

System.out.println("Default Constructor , Name : " + name + " Age : " + age);

}

Constructor(String name, int age) {

this.name = name;

this.age = age;

System.out.println("Parameterized Constructor , Name : " + name + " Age : " + age);

}

// Copy Constructor

Constructor(Constructor obj) {

this.name= obj.name ;

this.age=obj.age ;

System.out.println("Copy Constructor , Name : " + name + " Age : " + age);

}

public static void main(String[] args) {

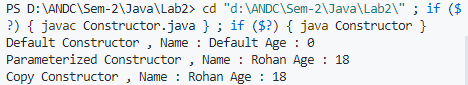
Constructor c1 = new Constructor();

Constructor c2 = new Constructor("Rohan", 18);

Constructor c3 = new Constructor(c2);

}

}



// Practise Set 5

/\* Program to demonstrate the difference between public and private access control in java \*/

class PublicPrivateDemo {

    int a;

    public int b;

    private int c;

    void setc(int x) {

        c = x;

    }

    int getc() {

        return c;

    }

}

public class AccessTest {

    public static void main(String[] args) {

        Access y = new Access();

        y.a = 50;

        y.b = 10;

        y.setc(90);

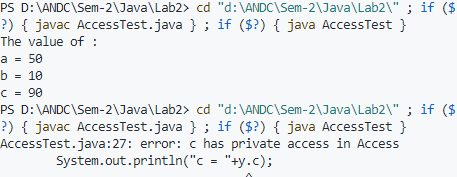
        System.out.println("The value of :\na = " + y.a + "\nb = " + y.b + "\nc = " + y.getc());

       //Comment Next Line

        System.out.println("c = "+y.c);

    }

}



LAB 3

//Practise Set 1 - Lab3

public class StaticDemo{

    public static int objCount=0;

    //Static Block

    static{System.out.println("This line is in static block");}

    StaticDemo(){

        objCount++;

    }

    static void lorem(){

        System.out.println("This is in a static method call.");

    }

    public static void main(String[] args){

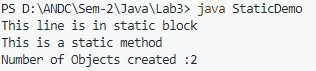
        StaticDemo d1= new StaticDemo();

        StaticDemo d2= new StaticDemo();

        lorem();

        System.out.println("Number of Objects created :"+objCount);

}



//Practise Set 2

public class StringDemo {

public static int getLength(String str){

return str.length();

}

public static void main (String args[]){

String du="Delhi University";

String andc= "Acharya Narendra Dev College" ;

String pushpa="Pushpa";

String kgf="KGF";

String dev="Dev";

String delhi="New Delhi";

System.out.println("Length of string \"Delhi University\" is "+getLength( du));

System.out.println("First occurence of 'U' is at index "+du.indexOf('U'));

System.out.println("Char at index 7 is "+du.charAt(7));

System.out.println( "\"Pushpa\" and \"KGF\" "+pushpa.compareTo(kgf));

System.out.println( "\"Pushpa\" and \"Pushpa\" "+pushpa.compareTo(pushpa));

System.out.println( "\"KGF\" and \"Pushpa\" "+kgf.compareTo(pushpa));

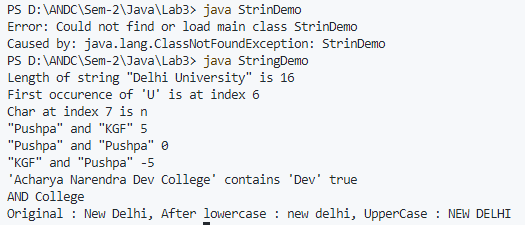
System.out.println("'Acharya Narendra Dev College' contains 'Dev' "+ andc.contains(dev));

System.out.println(andc.replace("Acharya Narendra Dev","AND"));

System.out.print("Original : "+delhi+", After lowercase : "+ delhi.toLowerCase()+", UpperCase : "+delhi.toUpperCase());

}

}



//Practise Set 3 -Lab3

public class VarArgs {

    static int calcSum(int ... arr){

        int sum=0;

        for(int x : arr){

            sum+=x;

        }

        return sum;

    }

    public static void main(String [] args){

        int sum=calcSum(10,20,30);

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

    }

}



LAB 4

//Program 1

class Father {

    String name;

    String surname;

}

public class SingleInheritance {

    int j;

    static void printRelation(Son son, Father dad){

        System.out.println("Child no "+son.childNO+" of "+dad.name+dad.surname +" is "+son.name+" "+son.surname);

    }

public static void main(String [] args){

    Son child1=new Son();

    child1.name="Rahul";

    child1.surname="Sharma";

    child1.childNO=1;

    Father dad=new Father();

    dad.name="Ramprasad ";

    dad.surname="Sharma";

    printRelation(child1,dad );

}

}

class Son extends Father{

    String name;

    int childNO;

}



//Program 2

// WAP to demonstrate Member Access and Inheritance

public class MemberAccess {

    public static void main(String [] args){

        Human.main();

Human.eatFood();

    }

}

class Human{

    public String name;

    protected static void eatFood(){

        System.out.println("Eating Food");

    }

    static void main(){

        System.out.println("This is Human:main");

    }

}

class Student extends Human{

    void study(){

        eatFood();

        System.out.println("Studying passionately ");

    }

}



//Program 3 -Lab 4

class BoxWeight extends Box {

    double weight; // weight of box

    // constructor for BoxWeight

    BoxWeight(double w, double h, double d, double wt) {

        width = w;

        height = h;

        depth = d;

        weight = wt;

    }

}

class Box {

    double width;

    double height;

    double depth;

    // construct clone of an object

    Box(Box ob) {

        width = ob.width;

        height = ob.height;

        depth = ob.depth;

    }

    // constructor used when all dimensions specified

    Box(double w, double h, double d) {

        width = w;

        height = h;

        depth = d;

    }

    Box() {

        width = -1; // use -1 to indicate

        height = -1; // an uninitialized

        depth = -1; // box

    }

    // constructor used when cube is created

    Box(double len) {

        width = height = depth = len;

    }

    // compute and return volume

    double volume() {

        return width \* height \* depth;

    }

}

public class DemoBoxWeight {

    public static void main(String args[]) {

        BoxWeight mybox1 = new BoxWeight(10, 20, 15, 34.3);

        BoxWeight mybox2 = new BoxWeight(2, 3, 4, 0.076);

        double vol;

        vol = mybox1.volume();

        System.out.println("Volume of mybox1 is " + vol);

        System.out.println("Weight of mybox1 is " +

                mybox1.weight);

        System.out.println();

        vol = mybox2.volume();

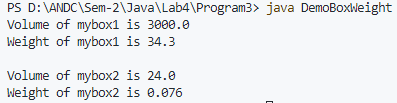
        System.out.println("Volume of mybox2 is " + vol);

        System.out.println("Weight of mybox2 is " +

                mybox2.weight);

    }

}



//Program 4 -Lab4

public class SuperSubReference {

    int i;

    public static void main(String [] args){

        SuperSubReference obj=new SubClass();

        System.out.println(obj.getClass());

    //We can see that following line will give error as,

    // refernce type is SuperSubReference type, and it don't know about SubClass method.

    // But still object is of type SubClass

       // System.out.println(obj.getString());

    }

}

class SubClass extends SuperSubReference{

    int x;

   String getString(){

    return "This is a demo string from SubClass.";

   }

}



//Program 5

public class SuperCallEg {

    int a;

    int b;

    SuperCallEg() {

        a = 0;

        b = 0;

    }

    SuperCallEg(int a, int b) {

        this.a = a;

        this.b = b;

    }

    public static void main(String[] args) {

        SubClass obj = new SubClass(2, 3);

        System.out.printf("a=%d b=%d x=%d y=%d", obj.a, obj.b, obj.x, obj.y);

    }

}

class SubClass extends SuperCallEg {

    int x;

    int y;

    SubClass() {

        super(0, 0);

        x = 0;

        y = 0;

    }

    SubClass(int x, int y) {

        // Calls Super with 1 incremented

        super(x + 1, y + 1);

        this.x = x;

        this.y = y;

    }

}



//Program 6

public class SuperAccess {

    int a;

    int b;

    int getSum(){

        return a+b;

    }

    SuperAccess(){

        a=0;

        b=0;

    }

    SuperAccess(int a,int b){

        this.a=a;

        this.b=b;

    }

    public static void main(String [] args){

        SubClass obj=new SubClass(10,5);

        obj.main();

        //Expected output 17

    }

}

class SubClass extends SuperAccess{

    int x;

    int y;

    SubClass(){

        x=0;

        y=0;

    }

    void main(){

        System.out.println( super.getSum());

    }

    SubClass(int x,int y){

        super(x+1,y+1);

        //Calls Super with 1 incremented

        this.x=x;

        this.y=y;

    }

}



LAB 5

// Lab Exercise 2

package P1;

import java.util.Formatter;

public class TwoDim {

int x;

int y;

public TwoDim(){

x=0;

y=0;

}

public TwoDim(int x,int y){

this.x=x;

this.y=y;

}

public String toString(){

Formatter f=new Formatter();

String res =f.format(("%dx +%dy"),x,y).toString();

return res;

}

}

------------------------------------------------------------

package P2;

import P1.TwoDim;

public class ThreeDim extends TwoDim{

int z;

public ThreeDim(){

z=0;

}

public ThreeDim(int x, int y,int z){

super(x, y);

this.z=z;

}

public String toString(){

String res =super.toString()+"+"+z+"z";

return res;

}

}

------------------------------------------------------------------------------------------

package P;

import P1.TwoDim;

import P2.ThreeDim;

public class Driver {

public static void main(String [] args){

TwoDim obj2d=new TwoDim(3,4);

ThreeDim obj3d=new ThreeDim(3,4,5);

TwoDim obj;

obj=obj3d;

System.out.println( obj.toString());

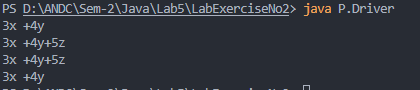
obj=obj2d;

System.out.println( obj.toString());

}

}

OUTPUT



//Practise Set 1 -Lab5

public class Multilevel {

    public static void main(String[] args) {

        XUV500 xuv500 = new XUV500();

        xuv500.model = " EX 1.5 ";

        xuv500.name = "XUV 500";

        System.out.println("You ordered " + xuv500.name + xuv500.model + "\nPrice = " + xuv500.price);

    }

}

class Vehicle {

    String name;

    String model;

}

class SUV extends Vehicle {

    int minWeight = 2000;

}

class XUV500 extends SUV {

    int price = 1\_400\_000;

}



//Practise 2

import java.util.Formatter;

class TwoDim {

    int x;

    int y;

    public TwoDim(){

        x=0;

        y=0;

    }

    public TwoDim(int x,int y){

        this.x=x;

        this.y=y;

    }

    public String toString(){

        Formatter f=new Formatter();

        String res =f.format(("%dx +%dy"),x,y).toString();

        return res;

    }

}

class ThreeDim extends TwoDim{

    int z;

    public ThreeDim(){

        z=0;

    }

    public ThreeDim(int x, int y,int z){

        super(x, y);

        this.z=z;

    }

    public String toString(){

        String res =super.toString()+"+"+z+"z";

        return res;

    }

}

public class DynamicDispatch {

    public static void main(String [] args){

        TwoDim obj ;

        obj=new TwoDim(3,4);

        System.out.println(obj.toString());

        obj=new ThreeDim(3,4,5);

        System.out.println(obj.toString());

    }

}



//Practise 3

abstract class AbstractDemo {

    int x;

    int y;

   abstract void doFun();

}

public class AbstractRun extends AbstractDemo{

    public AbstractRun(){

        x=0;

        y=0;

    }

    public AbstractRun(int x , int y){

        this.x=x;

        this.y=y;

    }

    void doFun(){

        System.out.println("Doing fun ");

    }

    public String toString(){

        String res=x+"x+"+y+"y";

        return res;

       }

    public static void main(String[] args) {

        AbstractRun obj=new AbstractRun() ;

        obj.doFun();

        System.out.println( obj.toString());

    }

}



LAB 6

//Lab Exercise 3

package P1;

abstract public class Shape{

    public String shape;

    public String **getShape**(){

        return "";

    }

    abstract public double **getArea**();

}

package P2;

import P1.Shape;

import java.util.Scanner;

public class Rectangle extends Shape{

    double length;

    double breadth;

    public **Rectangle**(){

        Scanner sc =new **Scanner**(System.in);

        System.out.**println**("Enter Length");

        length=sc.**nextDouble**();

        System.out.**println**("Enter Breadth");

        breadth=sc.**nextDouble**();

        shape="Rectangle";

    }

    public **Rectangle**(int length,int breadth){

        this.length=length;

        this.breadth=breadth;

        shape="Rectangle";

    }

    public String **getShape**(){

        return shape;

    }

    public double **getArea**(){

        return length\*breadth;

    }

}

package P3;

import java.util.Scanner;

import P1.Shape;

public class Circle extends Shape{

    double radius;

    public **Circle**(){

        Scanner sc=new **Scanner**(System.in);

        System.out.**println**("Enter Radius of circle");

        radius = sc.**nextDouble**();

    }

    public **Circle**(double radius){

        this.radius=radius;

    }

    public double **getArea**(){

        return (3.14159\*radius\*radius);

    }

     public String **getShape**(){

        return shape;

    }

}

import P1.Shape;

import P2.Rectangle;

import P3.Circle;

import java.util.Scanner;

public class Driver1{

    public static void **main**(String [] args){

        Shape s1;

        Scanner scan=new **Scanner**(System.in);

        System.out.**println**("1: Rectangle , 2 Circle");

        int op= scan.**nextInt**();

        choice:{

            switch (op){

                case 1:{

                    s1=new **Rectangle**();

                    System.out.**println**(s1.**getArea**());

                    break;

                }

                case 2:{

                    s1=new **Circle**();

                    System.out.**println**(s1.**getArea**());

                    break;

                }

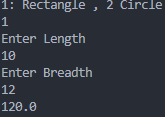
            }

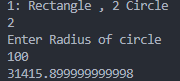
        }

    }

}

OUTPUT





//PRACTISE SET 1

public class InterfaceDemo implements MyMath{

   public void doFun(){

        System.out.println("Doing fun");

    }

    public static void main(String[] args){

        System.out.println(MyMath.PI);

        InterfaceDemo obj=new InterfaceDemo();

        obj.doFun();

    }

}

interface MyMath{

    double PI=3.14159;

    double e=2.71828;

    void doFun();

    default double sq(double x){

        return x\*x;

    }

}



//Practise Set 2 - Lab6

public class MultInterface {

    public static void main(String[] args) {

        College ref;

        ref=new CS();

        ref.show();

        ref=new Physics();

        ref.show();

    }

}

interface College{

    String name="ANDC";

    String address="Govindpuri,Delhi";

    void show();

}

class CS extends MultInterface implements College{

    String department="CS";

    public void show(){

        System.out.println("Department :"+department +" ,College :"+name);

    }

}

class Physics extends MultInterface implements College{

    String department="Physics";

    public void show(){

        System.out.println("Department :"+department +" ,College :"+name);

    }

}



//Practise Set 3

public class ExtendInterfaceDemo implements Car {

    public static void main(String [] args){

        Car obj=new ExtendInterfaceDemo();

        System.out.println( obj.getColor());

        System.out.println(obj.getcc());

        System.out.println(obj.getTyreCount());

    };

}

interface Body {

    String paint\_color="Black";

    default String getColor(){

        return paint\_color;

    }

}

interface Engine{

    int cc=1200;

    default int getcc(){

        return cc;

    }

}

interface Car extends Body,Engine{

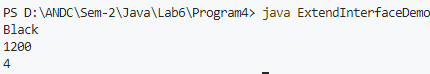
 int tyre\_count=4;

 default int getTyreCount(){

    return tyre\_count;

 }

}



//Practise 4 -Lab6

public class DefaultInterfaceDemo implements Engine{

    public static void main(String [] args){

        Engine obj=new DefaultInterfaceDemo();

        System.out.println(obj.getcc());

    }

}

interface Engine{

    int cc=1200;

    default int getcc(){

        return cc;

    }

}



//Practise 5

public class MultipleInheritance {

    public static void main(String [] args){

    Marksheet obj=new Marksheet("Mohan Das");

        System.out.println(obj.name+" have opted following subjects in 10th ");

        for (String string : obj.subjects10) {

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

        }

    }

}

interface Class10{

    String[] subjects10={"Hindi","English","Science","Mathematics","Social Science"};

}

interface Class12 {

    String stream="Science";

    String[] subjects12={"Physics","Chemistry","Mathematics ","English","CS"};

}

class Marksheet implements Class10,Class12{

    String name;

    Marksheet(String name){

        this.name=name;

    }

}



LAB 7

//Lab7 -Practise Set 1

import java.io.\*;

public class CatchExample {

    public static void main(String [] args){

        try{

            FileReader f=new FileReader("myFile.txt");

            //Un-comment next line to see a file not found exception block

            // f=new FileReader("text.txt");

            BufferedReader br=new BufferedReader(f);

            int temp=br.read();

            while(temp!=-1){

                System.out.print((char)temp);

                temp=br.read();

            }

            f.close();

        }

        catch(IndexOutOfBoundsException e){

            System.out.println("Index Out of Bound Exception block ");

        }

        catch(FileNotFoundException e){

            System.out.println("File not found Exception");

        }

        catch(Exception e){

            System.out.println("This is general exception ");

        }

        // Uncomment next lines to see unreachable error

        // catch(IOException e){

        //     System.out.println("This line won't be reached ");

        // }

    }

}



//Lab 7 - Practise 2

public class Try {

    public static void main(String [] args){

        try{

            int salary=100\_000;

            System.out.println(salary);

            try {

                int arr[]={10,20,30};

                System.out.println(arr[4]);

            }

            /\*This catch statement will be skipped as wrong Exception is there,

             the outer catch will be reached \*/

            catch(EventException e){

                System.out.println("Inner catch");

            }

        }

        catch(IndexOutOfBoundsException e){

            System.out.println("Outer catch");

        }

    }

}



//Lab Exercise No 5

import java.util.Scanner;

import java.lang.Exception;

class stackException extends Exception{

public String overflow()

{

return ("Stack Overflow:Could not add more");

}

public String empty()

{

return ("Stack Underflow:No element in stack");

}

}

class StackD{

int arr[];

int t=-1;

int size;

Scanner sc=new Scanner(System.in);

public StackD(int size){

this.size=size;

arr=new int[size];

}

public void push(int x) throws stackException{

if(t==size-1){

throw new stackException();

}

else

{

t++;

arr[t]=x;

}

}

public int pop() throws stackException{

if(t==-1)

{

throw new stackException();

}

else

{

return t--;

}

}

public void Display(){

int i;

if(t==-1)

System.out.println("Stack is Empty");

for(i=t;i>=0;i--){

System.out.println("Stack [" +i+"] = "+arr[i]+" ");

}

}

}

public class StackExample{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int e, size;

System.out.print("Enter the size of the Stack : ");

size=sc.nextInt();

StackD s=new StackD(size);

int opt;

do{

System.out.print("\n1.Push\t2.Pop\t3.Display\nEnter The Choice : ");

opt=sc.nextInt();

switch(opt){

case 1:

try{

System.out.print("\nEnter the Elements : ");

e=sc.nextInt();

s.push(e);

}

catch(stackException x){

System.out.println(x.overflow());

}

break;

case 2:

try{

s.pop();

}

catch(stackException x){

System.out.println(x.empty());

}

break;

case 3:

s.Display();

break;

default:

System.out.print("Wrong Choice");

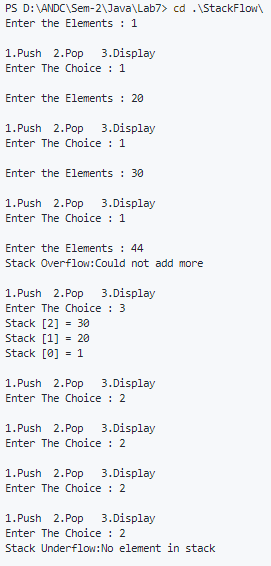
break;

}

}while(true );

}

}



//Lab Exercise No 4

import java.util.Scanner;

class UnderAge extends Exception{

    int age;

    UnderAge(int age){

        this.age=age;

    }

    public String toString(){

        String temp="Under Age: "+age;

        return temp;

    }

}

public class UnderAgeDemo{

    static void test(int age){

        try {

            if (age<18){

                    throw new UnderAge(age);

                }

            else{

                System.out.println("age is above 18");

            }

        }

        catch (UnderAge a){

                System.out.println(a.toString());

        }

        }

    public static void main(String []args ){

        Scanner sc=new Scanner(System.in);

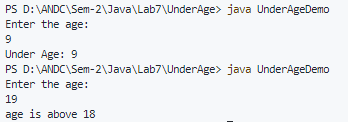
        System.out.println("Enter the age: ");

        int age=sc.nextInt();

        test(age);

    }

}



LAB 8

import java.io.\*;

public class CopyFileExample {

public static void main(String [] args){

int i;

FileInputStream fin;

FileOutputStream fout;

if(args.length==2){

System.out.println("Input Filename: "+args[0]);

System.out.println("Output Filename: "+args[1]);

}

try {

fin=new FileInputStream(args[0]);

fout=new FileOutputStream(args[1]);

do{

i=fin.read();

if(i!=-1){

fout.write(i);

}

}while(i!=-1);

}

catch(FileNotFoundException e){

System.out.println("File Not Found");

}

catch(IOException e){

System.out.println("Reading or Writing not possible");

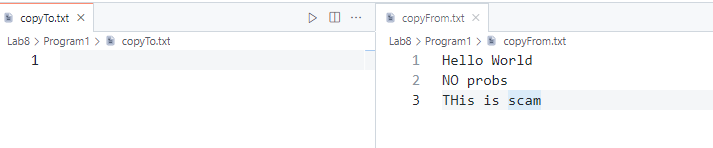
}

System.out.println("File Copy Successful");

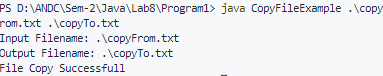
}

}

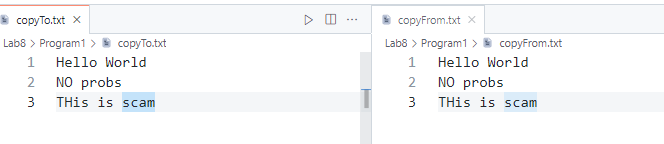
**BEFORE**



OUTPUT



**AFTER**



LAB EXERCISE 7

import java.io.\*;

public class SpecificLine {

public static void main(String[] args) {

String str;

int i;

if (args.length == 1) {

System.out.println("Input Filename: " + "Text.txt");

}

try {

FileReader fr = new FileReader("Text.txt");

BufferedReader br = new BufferedReader(fr);

str = br.readLine();

while (str != null) {

if ((str.charAt(0) == '/') && (str.charAt(1) == '/')) {

System.out.println(str.substring(2,str.length()));

}

str = br.readLine();

}

} catch (FileNotFoundException e) {

System.out.println("File Not Found");

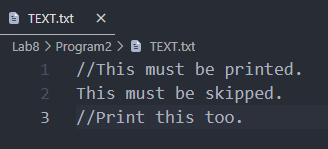
} catch (IOException e) {

System.out.println("Reading not possible");

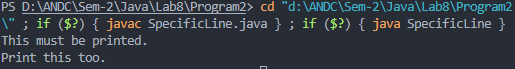
}

}

}



OUTPUT



Lab EXERCISE 8

// Write a program to handle mouse events(Clicked, Entered, Exited, Presses, and Released).

// Lab Exercise 8

import java.awt.\*;

import java.awt.event.\*;

public class MouseExample extends Frame implements MouseListener{

Label l;

MouseExample(){

addMouseListener(this);

l=new Label();

l.setBounds(20,50,100,20);

add(l);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse Clicked");

}

public void mouseEntered(MouseEvent e) {

l.setText("Mouse Entered");

}

public void mouseExited(MouseEvent e) {

l.setText("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

l.setText("Mouse Pressed");

}

public void mouseReleased(MouseEvent e) {

l.setText("Mouse Released");

}

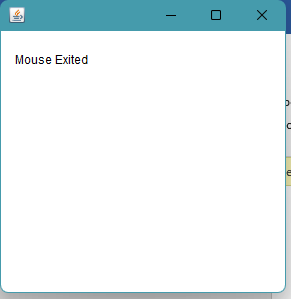
public static void main(String[] args) {

new MouseExample();

}

}

OUTPUT



LAB EXERCISE 9

import java.awt.\*;

import java.awt.event.\*;

public class KeyExample extends Frame implements KeyListener {

Label l;

TextArea area;

KeyExample() {

l = new Label();

l.setBounds(20, 50, 100, 20);

area = new TextArea();

area.setBounds(20, 80, 300, 300);

area.addKeyListener(this);

add(l);

add(area);

setSize(400, 400);

setLayout(null);

setVisible(true);

}

public void keyPressed(KeyEvent e) {

l.setText("Key Pressed");

}

public void keyReleased(KeyEvent e) {

l.setText("Key Released");

}

public void keyTyped(KeyEvent e) {

l.setText("Key Typed");

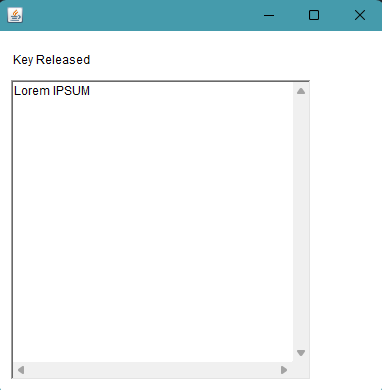
}

public static void main(String[] args) {

new KeyExample();

}

}



LAB 9

LAB EXERCISE 10

import java.awt.\*;

import java.awt.event.\*;

public class Pink extends Frame {

    Label l;

**Pink**() {

        super("Pink Screen Example");

        l = new **Label**("Hello World");

        l.**setBounds**(25, 50, 250, 30);

        l.**setAlignment**(Label.CENTER);

        this.**setBackground**(Color.PINK);

        this.**add**(l);

        this.**setSize**(300, 400);

        this.**setVisible**(true);

    }

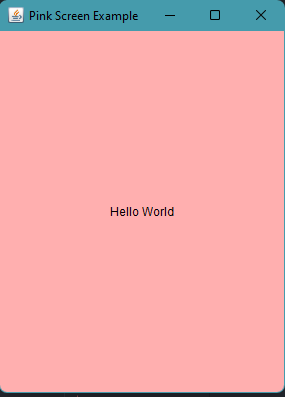
    public static void **main**(String[] args) {

        new **Pink**();

    }

}

OUTPUT



LAB EXERCISE 11

import java.awt.\*;

 import java.awt.event.\*;

 public class Main extends Frame implements ActionListener {

    Button btnRed, btnBlue;

**Main**() {

        super("AWT Buttons");

        btnRed = new **Button**("Red");

        btnRed.**setBounds**(25, 50, 250, 30);

        btnRed.**addActionListener**(this);

        this.**add**(btnRed);

        btnBlue = new **Button**("Blue");

        btnBlue.**setBounds**(25, 100, 250, 30);

        btnBlue.**addActionListener**(this);

        this.**add**(btnBlue);

        this.**setSize**(300, 160);

        this.**setLayout**(null);

        this.**setVisible**(true);

        this.**addWindowListener**(new **WindowAdapter**() {

            public void **windowClosing**(WindowEvent e) {

**dispose**();

            }

        });

    }

    public static void **main**(String[] args) {

                 new **Main**();

    }

    @Override public void **actionPerformed**(ActionEvent e) {

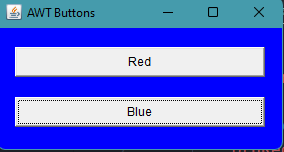
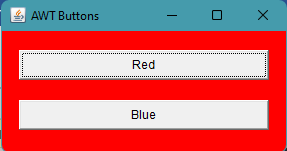
        if (e.**getSource**() == btnRed) { this.**setBackground**(Color.RED); }

        else if (e.**getSource**() == btnBlue) { this.**setBackground**(Color.BLUE); }

    }

}

OUTPUT

LAB EXERCISE 12

import java.awt.\*;

import java.awt.event.\*;

class KbdAdapter extends KeyAdapter {

    Label l;

**KbdAdapter**(Label l) {

         this.l = l;

    }

    @Override public void **keyTyped**(KeyEvent e) {

        l.**setText**("Typed character is: " + e.**getKeyChar**());

    }

    @Override public void **keyPressed**(KeyEvent e) {

        System.out.**println**("Pressed character is: " + e.**getKeyChar**());

    } @Override public void **keyReleased**(KeyEvent e) {

        System.out.**println**("Released character is: " + e.**getKeyChar**());

    }

}

public class Main extends Frame {

        Label l;

**Main**() {

        super("AWT Keyboard"); l = new **Label**("");

        l.**setBounds**(25, 50, 250, 30);

        l.**setAlignment**(Label.CENTER);

        this.**addKeyListener**(new **KbdAdapter**(l));

        this.**add**(l); this.**setSize**(300, 110);

        this.**setLayout**(null);

        this.**setVisible**(true);

        this.**addWindowListener**(new **WindowAdapter**() {

        public void **windowClosing**(WindowEvent e) {

**dispose**(); }

        }

        );

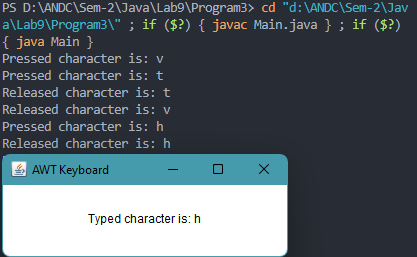
    }

    public static void **main**(String[] args) {

        new **Main**();

    }

}



LAB EXERCISE 13

import java.awt.\*;

import java.awt.event.\*;

 public class Main extends Frame implements ActionListener {

    Button btnA, btnB;

    Label la,lb;

**Main**() {

        super("Student Data");

        la=new **Label**("Name:James, Course :B.A. , Roll No : 1234 , College : Alien College ");

        lb= new **Label**("CGPA : 7.4");

        la.**setBounds**(25,250 ,1000,100);

        lb.**setBounds**(25,350,100,100);

        btnA = new **Button**("A");

        btnA.**setBounds**(25, 50, 100, 100);

        btnA.**addActionListener**(this);

        this.**add**(btnA);

        btnB = new **Button**("B");

        btnB.**setBounds**(25, 150, 100, 100);

        btnB.**addActionListener**(this);

        this.**add**(btnB);

        this.**setSize**(500, 500);

        this.**setLayout**(null);

        this.**addWindowListener**(new **WindowAdapter**() {

            public void **windowClosing**(WindowEvent e) {

**dispose**();

            }

        });

        this.**setVisible**(true);

    }

    public static void **main**(String[] args) {

                 new **Main**();

    }

    @Override public void **actionPerformed**(ActionEvent e) {

        if (e.**getSource**() == btnA) { this.**add**(la); }

        else if (e.**getSource**() == btnB) {

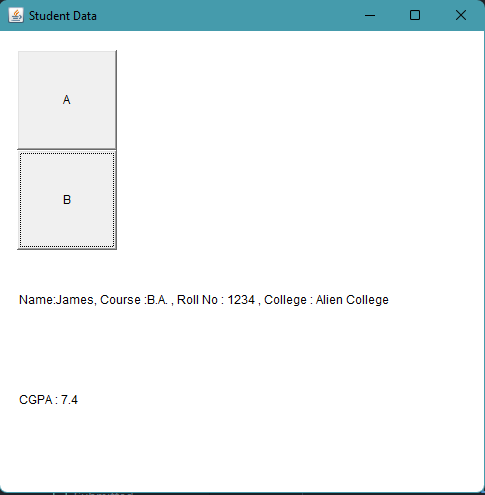
            this.**add**(lb);

            //this.setVisible(true);

        }

    }

}



XXXXXXXXXXXXXXXXXXXXXXXXXXX FINISHED 😉 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX