**LAB MANUAL**



NAME: S.RaghuRam

Roll No:AV.SC.U4CSE24315

SECTION:CSE-A

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **WEEK** | **TITLE** | **PAGE NO.** |
| 1. | **1** | Installation of JAVA | 4 |
|  |  | Basic Details Of a Student | 7 |
| 2. | **2** | Simple Interest | 9 |
|  |  | Factorial Of a Number | 11 |
|  |  | Celcius to Fahrenheit Temperature | 13 |
|  |  | Fahrenheit to Celcius Temperature | 14 |
|  |  | Area Of Rectangle and Triangle using Herons Formula | 16 |
|  |  | Fibonacii Series | 19 |
| 3. | **3** |  |  |
|  |  |  |  |
|  |  |  |  |

**WEEK-1:**How to install jdk and first program on

printing student details.

**Step-1:** Download JDK-21 from oracle website

**

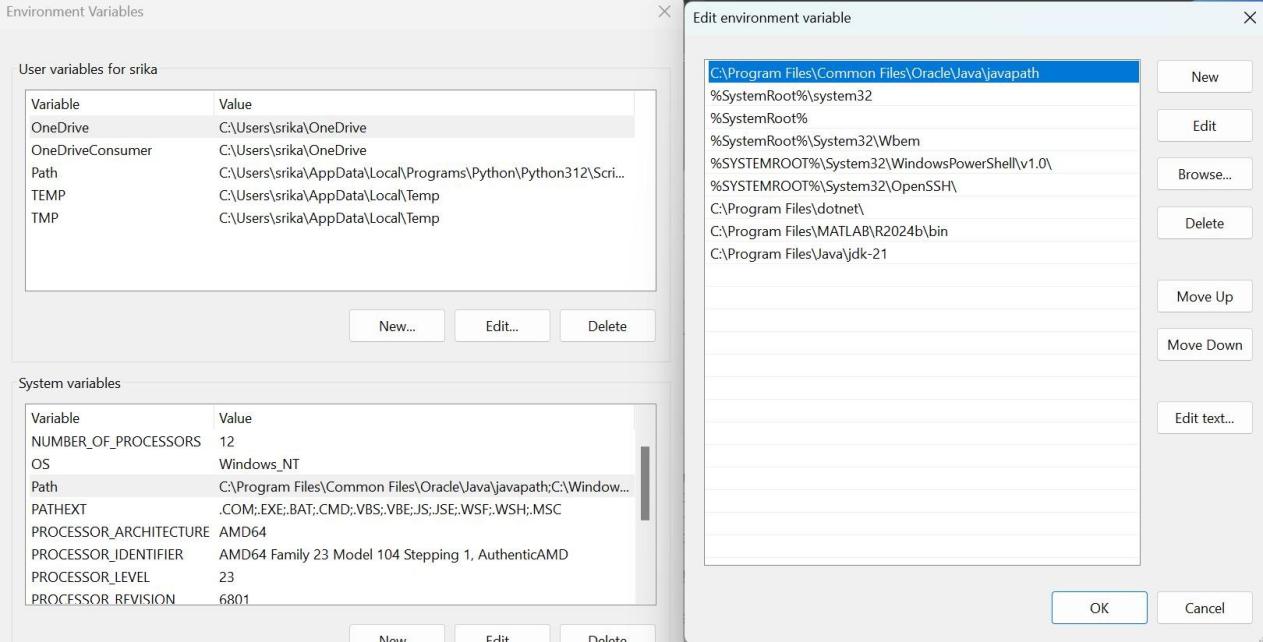
**Step-2:**Install the JDK-21 with accepting terms and

conditions according to the respective windows.

*.*

**

**Step-3:**Setting up environmental variables.

**

\*Windows c -> C-drive -> program files ->Java -

>JDK-21->select bin

\*Select and open environmental variable in search

bar-> either select system variables or user

variables-> select path-> click edit->New-> paste

the bin-> finish the setup(apply the changes).

~for verifying the installed version

Open cmd-> type java --version

~command propt

Javac filename.java ->compiling.

Java filename.java ->displaying

**PROGRAM-1:**Wirte a program in java for displaying

student details.

class  ME

{

    public static void main(String[] args)

    {

        System.out.println("NAME= S.RaghuRam")

        System.out.println("ROLL NO= AV.SC.U4CSE24315");

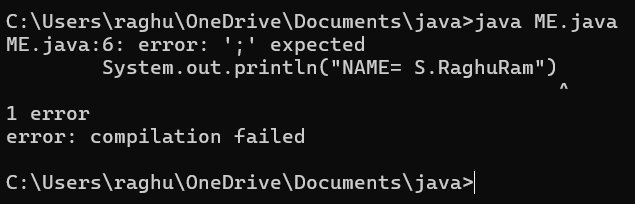
        System.out.println("SEC= A”);

        System.out.println("BRANCH= CSE");

    }

}

**Output:**



|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| System.out.println  ("NAME:  S.RaghuRam") | Semi colon(;) is  missing at the  end. | Add a semi colon(;) at the  end.  System.out.println("NAME:  S.RaghuRam"); |

**PROGRAM-1(Rectified):**

class ME

{

public static void main(String[] args)

{

System.out.println("NAME= S.RaghuRam");

System.out.println("ROLL NO= AV.SC.U4CSE24315");

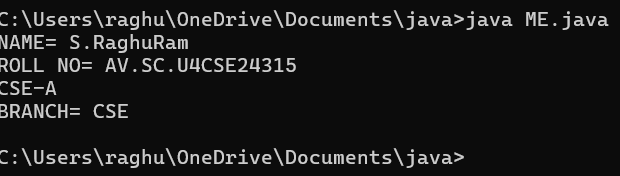
System.out.println("CSE-A");

System.out.println("BRANCH= CSE");

}

}

**OUTPUT:**

****

**WEEK-2:**

**PROGRAM-1:** Write a java program for SI?

import java.util.Scanner;

class SI{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("Enter Principal Amount :");

double p = input.nextDouble();

System.out.println("Enter Time:");

double t=input.nextDouble();

System.out.println("Enter rate of Interest:");

double r=input.nextDouble();

double I=(p\*t\*r)/100;

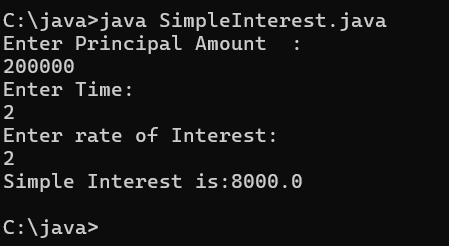
System.out.println("Simple Interest is:"+ I);

input.close();

}

}

**OUTPUT:**

****

**PROGRAM-2:** Write a program in java for area of

rectangle.

import java.util.Scanner;

class Main{

public static void main(String[] args){

Scanner input = new Scanner (System.in);

System.out.println("Enter value of length:");

double L=input.nextDouble();

System.out.println("Enter value of Width:");

double W=input.nextDouble();

double A=L\*W;

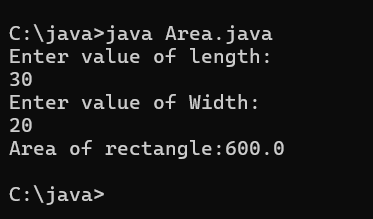
System.out.println("Area of rectangle:"+A);

input.close();

}

}

**OUTPUT:**



**PROGRAM-3:**Write a program in java for area of

triangle using heron’s formula.

import java.util.Scanner;

class Main{

public static void main(String[] args){

Scanner input = new Scanner (System.in);

System.out.println("Enter value of a:");

double a=input.nextDouble();

System.out.println("Enter value of b:");

double b=input.nextDouble();

System.out.println("Enter value of c:");

double c=input.nextDouble();

double s=(a+b+c)/2;

double A=sqrt(s\*(s-a)\*(s-b)\*(s-c));

System.out.println("Area of triangle:"+A);

input.close();

}

}

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| double A=sqrt(s\*(s-a)\*(s-b)\*(s-c)); | double A=sqrt(s\*(s-a)\*(s-b)\*(s-c));  ^  symbol: method sqrt(double)  location: class Main  1 error  ERROR!  error: compilation failed | Use Math.sqrt statement.  double  A=Math.sqrt(x); |

**PROGRAM-3(Rectified):**

import java.util.Scanner;

class Main{

public static void main(String[] args){

Scanner input = new Scanner (System.in);

System.out.println("Enter value of a:");

double a=input.nextDouble();

System.out.println("Enter value of b:");

double b=input.nextDouble();

System.out.println("Enter value of c:");

double c=input.nextDouble();

double s=(a+b+c)/2;

double A=Math.sqrt(s\*(s-a)\*(s-b)\*(s-c));

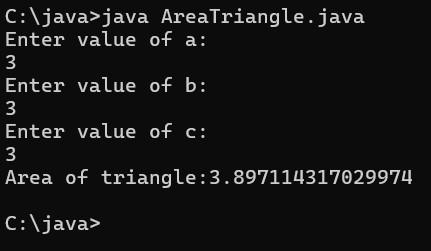
System.out.println("Area of triangle:"+A);

input.close();

}

}

**OUTPUT:**

****

**PROGRAM-4(a):**Write a program in java for converting temperature from celsius to fahrenite.

import java.util.Scanner;

class Main{

public static void main(String[] args){

Scanner input = new Scanner(System.in);

System.out.println("Enter temperature in Celsius:");

double C=input.nextDouble();

double F=(9\*C/5)+32;

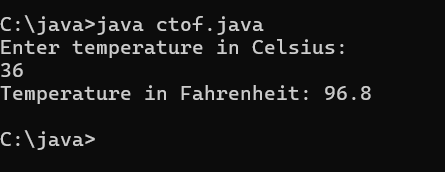
System.out.println("Temperature in Fahrenheit: "+F);

input.close();

}

}

**OUTPUT:**



**PROGRAM-4(b):**Write a program in java for converting temperature from fahrenite to celsius.

import java.util.Scanner;

class Main{

public static void main(String[] args){

Scanner input = new Scanner(System.in);

System.out.println("Enter temperature in Fahrenheit:");

double f=input.nextDouble();

double C=(f-32)\*5/9;

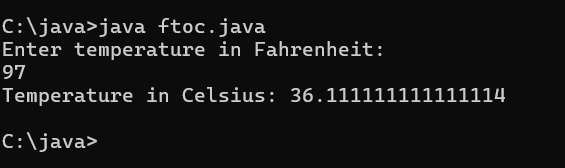
System.out.println("Temperature in Celsius: "+C);

input.close();

}

}

**OUTPUT:**



**PROGRAM-5:**Write a program in java for factorial of a number.

import java.util.Scanner;

class Factorial {

public static void main(String[] args)

{

int num;

int factorial = 1;

int i;

Scanner input = new Scanner(System.in);

System.out.println("Enter a number: ");

num = input.nextInt();

for(i = 1; i > num; i++)

{

factorial \*= i;

}

System.out.println(num+"!"+"="+factorial);

input.close();

}

}

**OUTPUT:**

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| for(i = 1; i > num; i++) | Output is wrong | for(i = 1; i <= num; i++) |

**PROGRAM-5(Rectified):**

import java.util.Scanner;

class Factorial {

public static void main(String[] args)

{

int num;

int factorial = 1;

int i;

Scanner input = new Scanner(System.in);

System.out.println("Enter a number: ");

num = input.nextInt();

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

{

factorial \*= i;

}

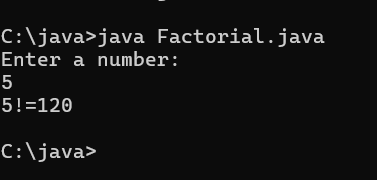
System.out.println(num+"!"+"="+factorial);

input.close();

}

}

**OUTPUT:**

****

**PROGRAM-5:**Write a program in java for fibonacci series.

import java.util.Scanner;

class fibonacci{

public static void main(String[] args){

Scanner input=new Scanner(System.in);

System.out.print("enter number of digits to be displayed :");

int num=input.nextInt();

int first=0;

int second=1;

int third;

System.out.println("fibonaaci series for "+ num +" number of terms :");

System.out.println(first);

System.out.println(second);

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

third=first+second;

first=second;

second=third;

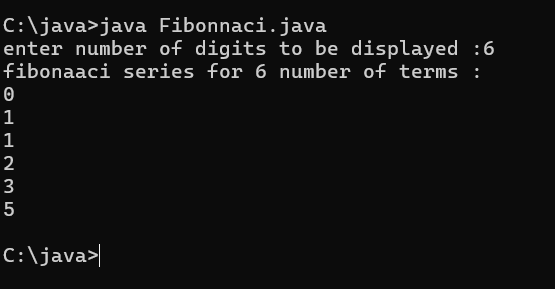
System.out.println(second);

}

}

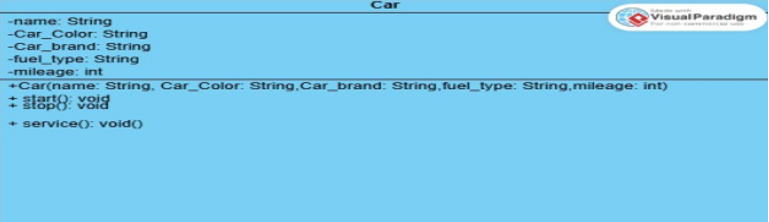
}

**OUTPUT:**



**WEEK-3:**

**CLASS DIAGRAM:**

****

**PROGRAM-1:**Write a program in java with following instructions

1. Create a class with name car.
2. Create 4 attributes named car\_color,car\_brand,fuel\_type,mileage.
3. Create three methods named start(),stop() and service().
4. Create three objects named car1,car2,car3.

class Car{

public String car\_color;

public String car\_brand;

public String fuel\_type;

public String mileage;

public void start(){

System.out.println("Car Started");

}

public void stop(){

System.out.println("Car Stopped");

}

public void service(){

System.out.println("Car has given to service");

}

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println("Class:CSE-A");

System.out.println("Roll No:24315");

System.out.println("");

car car1=new car();

car1.car\_color="Red";

car1.car\_brand="Tata";

car1.fuel\_type="Petrol";

car1.mileage="70kmpl";

System.out.println("Car1 :");

car1.start();

car1.stop();

car1.service();

System.out.println("Car Color is "+car1.car\_color);

System.out.println("Car Brand is "+car1.car\_brand);

System.out.println("Car fuel type is "+ car1.fuel\_type);

System.out.println("Car mileage is "+ car1.mileage);

System.out.println("");

Car car2=new Car();

car2.car\_color="Black";

car2.car\_brand="BMW";

car2.fuel\_type="Diesel";

car2.mileage="80kmpl";

System.out.println("Car2 :");

car2.start();

car2.stop();

car2.service();

System.out.println("Car Color is "+car2.car\_color);

System.out.println("Car Brand is "+car2.car\_brand);

System.out.println("Car fuel type is "+ car2.fuel\_type);

System.out.println("Car mileage is "+ car2.mileage);

System.out.println("");

Car car3=new Car();

car3.car\_color="White";

car3.car\_brand="Audi";

car3.fuel\_type="Diesel";

car3.mileage="75kmpl";

System.out.println("Car3 :");

car3.start();

car3.stop();

car3.service();

System.out.println("Car Color is "+car3.car\_color);

System.out.println("Car Brand is "+car3.car\_brand);

System.out.println("Car fuel type is "+ car3.fuel\_type);

System.out.println("Car mileage is "+ car3.mileage);

System.out.println("");

Car car4=new Car();

car4.car\_color="Green";

car4.car\_brand="Ferrari";

car4.fuel\_type="Petrol";

car4.mileage="90kmpl";

System.out.println("Car4 :");

car4.start();

car4.stop();

car4.service();

System.out.println("Car Color is "+car4.car\_color);

System.out.println("Car Brand is "+car4.car\_brand);

System.out.println("Car fuel type is "+ car4.fuel\_type);

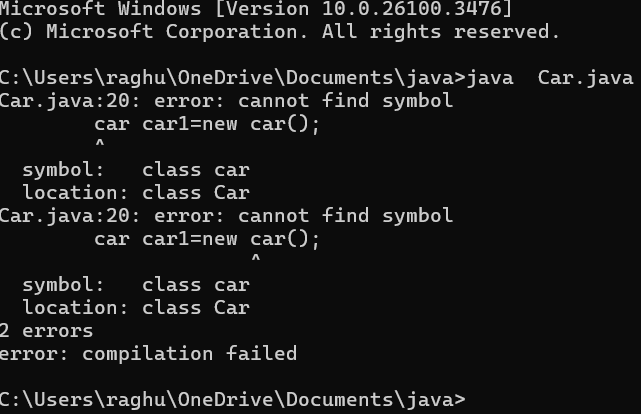
System.out.println("Car mileage is "+ car4.mileage);

System.out.println("");

}

}

**OUTPUT:**

****

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| car car1=new car(); | Capital C used for class name | Car car1=new Car(); |

**PROGRAM-1(RECTIFIED):**

class Car{

public String car\_color;

public String car\_brand;

public String fuel\_type;

public String mileage;

public void start(){

System.out.println("Car Started");

}

public void stop(){

System.out.println("Car Stopped");

}

public void service(){

System.out.println("Car has given to service");

}

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println("Class:CSE-A");

System.out.println("Roll No:24315");

System.out.println("");

Car car1=new Car();

car1.car\_color="Red";

car1.car\_brand="Tata";

car1.fuel\_type="Petrol";

car1.mileage="70kmpl";

System.out.println("Car1 :");

car1.start();

car1.stop();

car1.service();

System.out.println("Car Color is "+car1.car\_color);

System.out.println("Car Brand is "+car1.car\_brand);

System.out.println("Car fuel type is "+ car1.fuel\_type);

System.out.println("Car mileage is "+ car1.mileage);

System.out.println("");

Car car2=new Car();

car2.car\_color="Black";

car2.car\_brand="BMW";

car2.fuel\_type="Diesel";

car2.mileage="80kmpl";

System.out.println("Car2 :");

car2.start();

car2.stop();

car2.service();

System.out.println("Car Color is "+car2.car\_color);

System.out.println("Car Brand is "+car2.car\_brand);

System.out.println("Car fuel type is "+ car2.fuel\_type);

System.out.println("Car mileage is "+ car2.mileage);

System.out.println("");

Car car3=new Car();

car3.car\_color="White";

car3.car\_brand="Audi";

car3.fuel\_type="Diesel";

car3.mileage="75kmpl";

System.out.println("Car3 :");

car3.start();

car3.stop();

car3.service();

System.out.println("Car Color is "+car3.car\_color);

System.out.println("Car Brand is "+car3.car\_brand);

System.out.println("Car fuel type is "+ car3.fuel\_type);

System.out.println("Car mileage is "+ car3.mileage);

System.out.println("");

Car car4=new Car();

car4.car\_color="Green";

car4.car\_brand="Ferrari";

car4.fuel\_type="Petrol";

car4.mileage="90kmpl";

System.out.println("Car4 :");

car4.start();

car4.stop();

car4.service();

System.out.println("Car Color is "+car4.car\_color);

System.out.println("Car Brand is "+car4.car\_brand);

System.out.println("Car fuel type is "+ car4.fuel\_type);

System.out.println("Car mileage is "+ car4.mileage);

System.out.println("");

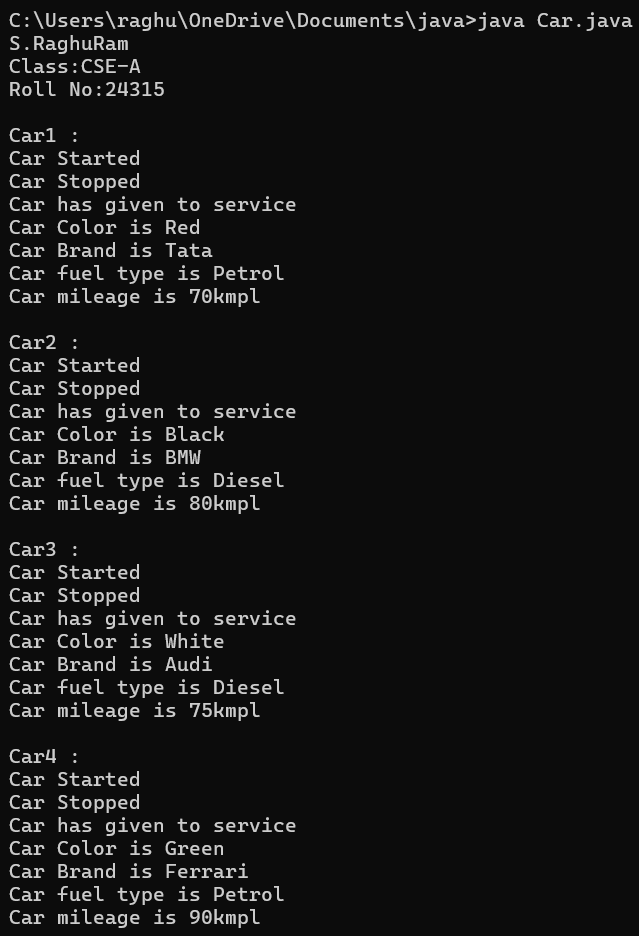
}

}

**Important points:**

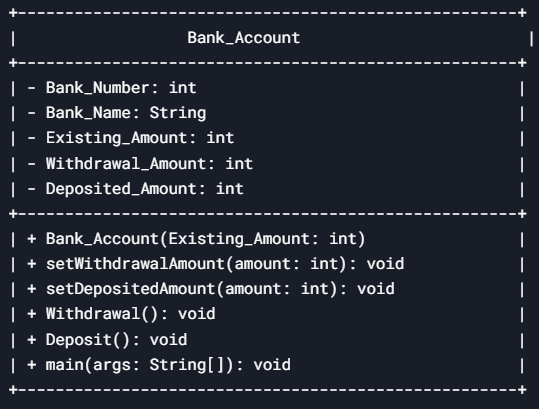
Encapsulation is achieved as methods and variables are existing in the same class.

**OUTPUT:**



**PROGRAM-2:** Write a program in java to create class Bank\_Account with methods deposit and withdrawal.

**CLASS DIAGRAM:**

****

import java.util.Scanner;

class Bank\_Account {

private int Bank\_Number;

public String Bank\_Name;

private int Existing\_Amount;

private int Withdrawal\_Amount;

private int Deposited\_Amount;

public Bank\_Account(int Existing\_Amount) {

this.Existing\_Amount = Existing\_Amount;

}

public void Withdrawal() {

if (Withdrawal\_Amount > this.Existing\_Amount) {

System.out.println("Withdrawal is not possible, ERROR!");

} else {

System.out.println("Withdrawal is possible");

int Total = Existing\_Amount - Withdrawal\_Amount;

System.out.println("Remaining amount is " + Total);

}

}

public void Deposit() {

System.out.println(Deposited\_Amount + " amount is successfully deposited");

int Total\_Amount = Existing\_Amount + Deposited\_Amount;

System.out.println("Total amount is " + Total\_Amount);

}

public static void main(String[] args) {

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println(“Roll:24315”);

System.out.println("");

Scanner input = new Scanner(System.in);

Bank\_Account bank = new Bank\_Account(100000);

bank.Bank\_Number = 1234567;

bank.Bank\_Name = "SBI";

System.out.println("Account Number: "+ bank.Bank\_Number);

System.out.println("Bank Name: "+bank.Bank\_Name);

System.out.println("");

System.out.println("Enter Withdrawal Amount:");

int withdrawal = input.nextInt();

bank.setWithdrawalAmount(withdrawal);

System.out.println("Enter Deposit Amount:");

int deposit = input.nextInt();

bank.setDepositedAmount(deposit);

bank.Withdrawal();

bank.Deposit();

}

}

**Output:**

|  |  |  |
| --- | --- | --- |
| code | Error | rectification |
| Not used set method to withdrawal and deposited amount | Wrong Output | Use set method to withdrawal and deposited amount |

**PROGRAM-2(Rectified):**

import java.util.Scanner;

class Bank\_Account {

private int Bank\_Number;

public String Bank\_Name;

private int Existing\_Amount;

private int Withdrawal\_Amount;

private int Deposited\_Amount;

public Bank\_Account(int Existing\_Amount) {

this.Existing\_Amount = Existing\_Amount;

}

public void setWithdrawalAmount(int amount) {

this.Withdrawal\_Amount = amount;

}

public void setDepositedAmount(int amount) {

this.Deposited\_Amount = amount;

}

public void Withdrawal() {

if (Withdrawal\_Amount > this.Existing\_Amount) {

System.out.println("Withdrawal is not possible, ERROR!");

} else {

System.out.println("Withdrawal is possible");

int Total = Existing\_Amount - Withdrawal\_Amount;

System.out.println("Remaining amount is " + Total);

}

}

public void Deposit() {

System.out.println(Deposited\_Amount + " amount is successfully deposited");

int Total\_Amount = Existing\_Amount + Deposited\_Amount;

System.out.println("Total amount is " + Total\_Amount);

}

public static void main(String[] args) {

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

Scanner input = new Scanner(System.in);

Bank\_Account bank = new Bank\_Account(100000);

bank.Bank\_Number = 1234567;

bank.Bank\_Name = "SBI";

System.out.println("Account Number: "+ bank.Bank\_Number);

System.out.println("Bank Name: "+bank.Bank\_Name);

System.out.println("");

System.out.println("Enter Withdrawal Amount:");

int withdrawal = input.nextInt();

bank.setWithdrawalAmount(withdrawal);

System.out.println("Enter Deposit Amount:");

int deposit = input.nextInt();

bank.setDepositedAmount(deposit);

bank.Withdrawal();

bank.Deposit();

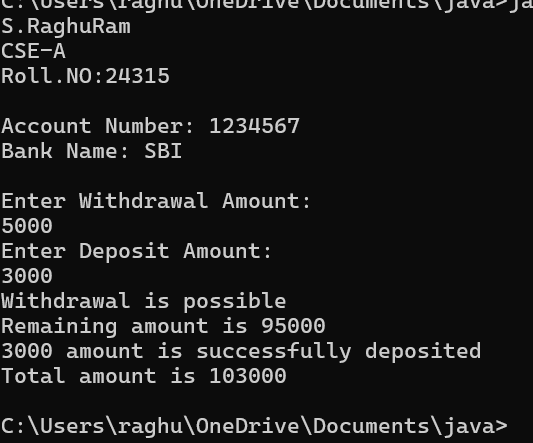
}

}

**Important points:**

1. Encapsulation is achieved as methods and variables are existing in the same class.
2. Constructer is used ***.***
3. Set Method is Used

**OUTPUT:**

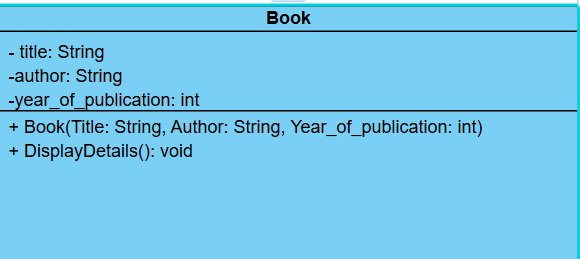


***WEEK-4:***

***PROGRAM-1:*** Write a java program with class named “book”, the class should contain various attributes such as title, author, year of publication it should also contain a constructor with parameters which initializes, title, author, and year of publication.

Create a method which displays the details of the book and display the details of two books

***class diagram:***



class Book{

public String Title;

public String Author;

public int Publish\_date;

Book(String Title,String Author,int Publish\_date){

this.Title=Title;

this.Author=Author;

this.Publish\_date= Publish\_date;

}

public void display(){

System.out.println("The title of the book is "+Title);

System.out.println("The author of the book is "+Author);

System.out.println("The publish date of the book is "+Publish\_date);

}

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

Book book1=new Book("Harry Potter","J.K. Rowling",1997);

book1.display();

System.out.println("");

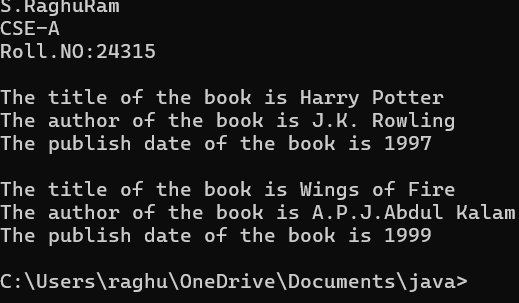
Book book2=new Book("Wings of Fire","A.P.J.Abdul Kalam",1999);

book2.display();

}

}

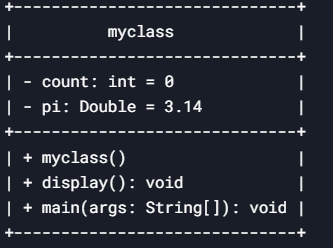
**OUTPUT:**

****

**PROGRAM-2:**

Create a java Program with class named myclass with static variable count of int type, initialized to zero and a constant variable “pi” of type double initialized to 3.14 as attributes of the class, ow define a constructor for “myclass” that increments the count variable each time an object of my class is created (count++), finally print the final values of count and pi variables create three objects.

**class diagram:**



class myclass{

static int count=0;

static Double pi=3.14;

myclass(){

count++;

}

public void display(){

System.out.println("Count value: "+count);

System.out.println("pi value: "+pi);

}

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

myclass hello=new myclass();

hello.display();

System.out.println("");

myclass hi=new myclass();

hi.display();

System.out.println("");

myclass my=new myclass();

my.display();

}

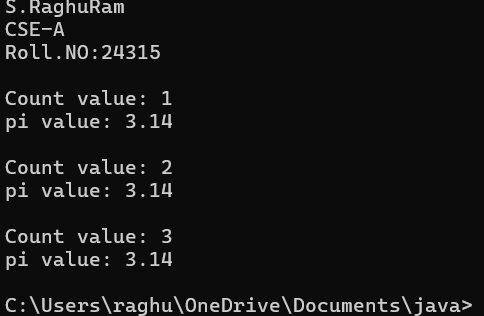
}

**Important points:**

**Static Variable (**count**)**

The variable count is declared as static, meaning it is shared among all instances of the MyCount class.It starts at 0 and increments each time a new object is created.

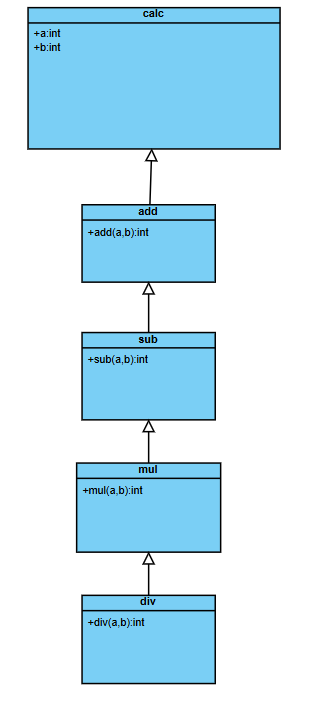
**OUTPUT:**



**WEEK-5:**

**AIM:** Create a calculator using the operations including addition, subtraction, multiplication and division using multi level inheritance and display the desired output.

**class diagram:**



**CODE:**

class calc{

    public int a;

    public int b;

        public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

        add obj1 = new add();

        sub obj2 = new sub();

        mul obj3 = new mul();

        div obj4 = new div();

        System.out.println("addition of the following numbers is :"+obj1.add(10,20));

        System.out.println("subtraction of the follwing numbers is :" +obj2.sub(20, 10));

        System.out.println("multiplication of the follwing number is :" +obj3.mul(10, 20));

        System.out.println("division of the follwing number is :" +obj4.div(20, 10));

        }

    }

class add extends calc{

    public int add(int a, int b){

        int addition = a+b;

        return addition;

    }

}

class sub extends add{

    public int sub(int a, int b){

        int subtraction = a-b;

        return subtraction;

    }

}

class mul extends sub{

    public int mul(int a, int b){

        int multiplication = a\*b;

        return multiplication;

    }

}

class div extends mul{

    public float div(int a, int b){

        if(b==0){

            System.out.println("the value of b is incorrect");

        }

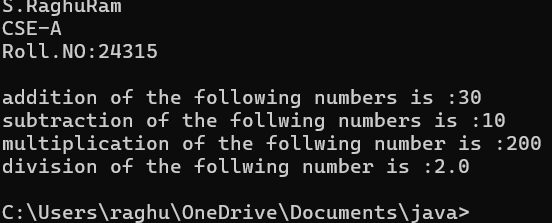
        int division = a/b;

        return division;

    }

}

**OUTPUT:**

****

**Important points:**

1. There are constructors for each and every subclass initializing values for each and every operation.
2. The concept of multilevel inheritance is used here.

**PROGRAM:2**

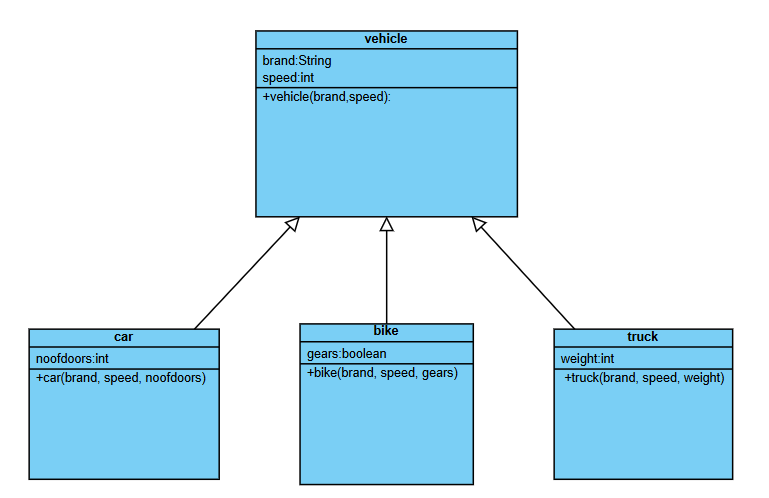
**AIM:**

A vehicle rental company wants to develop a system that maintains information about different types of vehicles available for rent. The company rents out cars , bikes and trucks and they need a program to store details about each vehicle, such as brand and speed.

**CONDITIONS:**

1. Cars should have an additional property such as number\_of\_doors .
2. Bikes should have a property indicating whether they have gears or not.
3. Truck should have a property of their capacity(in tons).
4. Every class should have a constructor.

**class diagram:**



**CODE:**

class vehicle{

String brand;

int speed;

public vehicle(String brand,int speed){

this.brand=brand;

this.speed=speed;

}

public static void main(String[] args) {

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

car obj1=new car("BMW",120,4);

bike obj2=new bike("Honda",100,true);

truck obj3=new truck("Mahindra",80,40);

}

}

class car extends vehicle{

int noofdoors;

public car(String brand, int speed,int noofdoors) {

super(brand, speed);

this.noofdoors=noofdoors;

System.out.println("brand of the car is :"+brand);

System.out.println("speed of the car is :"+speed);

System.out.println("number of doors :"+noofdoors);

}

}

class bike extends vehicle{

boolean gears;

public bike(String brand,int speed,boolean gears){

super(brand, speed);

this.gears=gears;

System.out.println("brand of the bike is :"+brand);

System.out.println("speed of the bike is :"+speed);

System.out.println("gears are present :"+gears);

}

}

class truck extends vehicle{

int weight;

public truck(String brand,int speed,int weight){

super(brand,speed);

this.weight=weight;

System.out.println("brand of the truck is :"+brand);

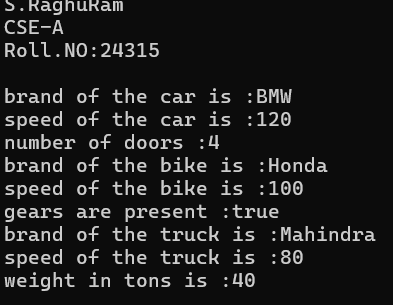
System.out.println("speed of the truck is :"+speed);

System.out.println("weight in tons is :"+weight);

}

}

**OUTPUT:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Name** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

**Important points:**

1. The car, bike, and truck classes inherit from the vehicle. Hierarchical Inheritance is being used here.
2. Each subclass calls the parent constructor using super(brand, speed);, ensuring proper initialization.

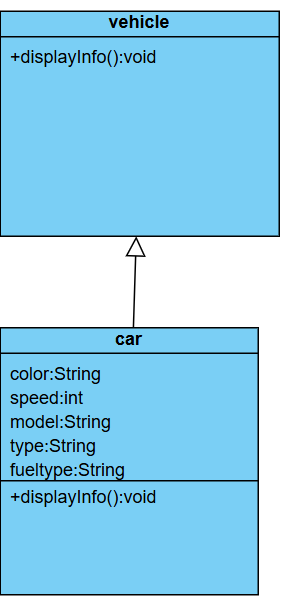
**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Name** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

**WEEK-6**

1. **Write a Java program to create a vehicle class with a method displayInfo(). Override this method in the car subclass to provide specific information about a car, model, fuel type, and color using the constructor**

**CLASS DIAGRAM:**



**CODE:  
class Vehicle {**

**String Brand;**

**String model;**

**String fuel;**

**String color;**

**int capacity;**

**Vehicle(String Brand, String model, String fuel, int capacity, String color) {**

**this.Brand = Brand;**

**this.model = model;**

**this.fuel = fuel;**

**this.capacity = capacity;**

**this.color = color;**

**}**

**void displayInfo(String Brand, String model, String fuel, int capacity, String color) {**

**System.out.println("Vehicle Details: ");**

**System.out.println("Brand: " + Brand);**

**System.out.println("Model: " + model);**

**System.out.println("Fuel: " + fuel);**

**System.out.println("Capacity: " + capacity);**

**System.out.println("Color: " + color);**

**}**

**}**

**class Car extends Vehicle {**

**Car(String Brand, String model, String fuel, int capacity, String color) {**

**super(Brand, model, fuel, capacity, color);**

**}**

**void displayInfo() {**

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

**System.out.println("Car Details: ");**

**System.out.println("Brand: " + Brand);**

**System.out.println("Model: " + model);**

**System.out.println("Fuel: " + fuel);**

**System.out.println("Capacity: " + capacity);**

**System.out.println("Color: " + color);**

**}**

**}**

**class Week6\_1 {**

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

**// Creating an instance of Car**

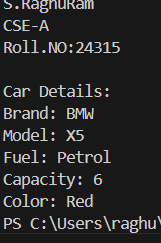
**Car car1 = new Car("BMW", "X5", "Petrol", 6, "Red");**

**car1.displayInfo(); // Display car details**

**}**

**}**

**Output:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Error** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

**IMPORTANT POINTS / EXPLANATION :**

1. Here class vehicle and car share a single level inheritance between them.
2. Here in the above code is an example of method overriding in java where the method names and their signature are same in both super class and sub class.

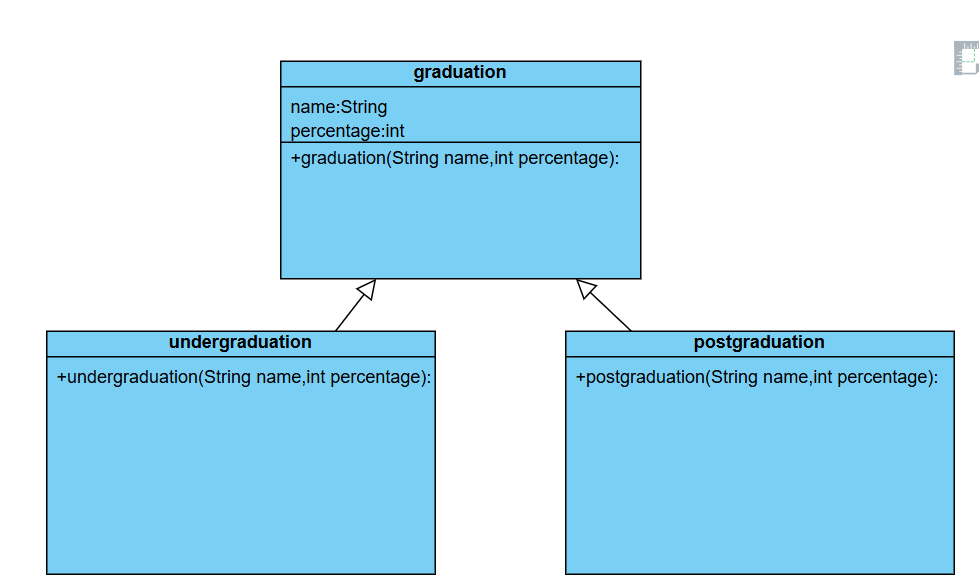
**2) Create a Java program for the scenario.**

**A college is developing an automated admission system that verifies student eligibility for undergraduate (UG) and postgraduate(PG) programs. Each program has different eligibility criteria based on the student's percentage in their previous qualification.**

**i) UG admissions require a minimum of 60%**

**ii) PG admissions require a minimum of 70%**

**CLASS DIAGRAM:**

****

**CODE:**

**class College{**

**String name;**

**int percentage;**

**void geteligibility(String name,int percentage){**

**this.name=name;**

**this.percentage=percentage;**

**}**

**}**

**class UG extends College{**

**void geteligibility(String name,int percentage){**

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

**if (percentage>=60){**

**System.out.println(name+" is eligible");**

**}**

**else{**

**System.out.println(name+" is not eligible");**

**}**

**}**

**}**

**class PG extends College{**

**void geteligibility(String name,int percentage){**

**if (percentage>=70){**

**System.out.println(name+" is eligible");**

**}**

**else{**

**System.out.println(name+" is not eligible");**

**}**

**}**

**}**

**class week6\_2{**

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

**UG ug=new UG();**

**ug.geteligibility("Person-1",40);**

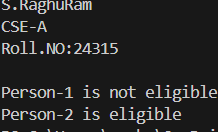
**PG pg=new PG();**

**pg.geteligibility("Person-2",80);**

**}**

**}**

**OUTPUT:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Error** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

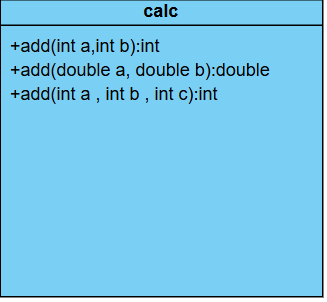
**3) Write a Java Program to create a Calculator class with overloaded methods to perform addition: Take the integer values a and b from the user.**

**i) Add two integers**

**ii) Add two doubles**

**iii) Add three integers**

**CLASS DIAGRAM:**

****

**CODE:**

**class Calc{**

**public int add(int a,int b){**

**return a+b;**

**}**

**public double add(double a,double b){**

**return a+b;**

**}**

**public int add(int a,int b,int c){**

**return a+b+c;**

**}**

**}**

**class week6\_3{**

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

**Calc C1=new Calc();**

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

**System.out.println("Sum of 6 and 9 is: "+C1.add(6,9));**

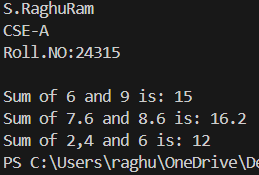
**System.out.println("Sum of 7.6 and 8.6 is: "+C1.add(7.6,8.6));**

**System.out.println("Sum of 2,4 and 6 is: "+C1.add(2,4,6));**

**}**

**}**

**Output:**

****

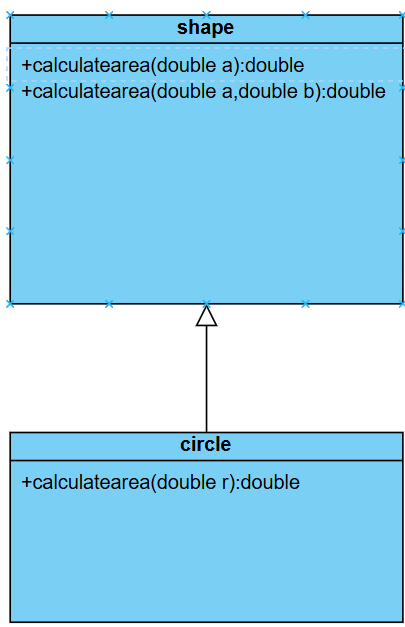
**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Error** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

**IMPORTANT POINTS:**

1. All the methods have same name add but each and every method have different signature.
2. Method overloading is taking place.
3. **Write a Java Program to create a shape class with a method calculateArea() that is overloaded for different shapes(e.g., Square, Rectangle ). Then create a subclass Circle that overrides the calculateArea() method for a circle.**

**ClassDiagram:**

****

**CODE:**

**class Shape {**

**double calculateArea(double side) {**

**return side \* side;**

**}**

**double calculateArea(double width, double height) {**

**return width \* height;**

**}**

**}**

**class Circle extends Shape {**

**double calculateArea(double radius) {**

**return 3.14 \* radius \* radius;**

**}**

**}**

**class Week6\_4 {**

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

**Shape S1 = new Shape();**

System.out.println("S.RaghuRam");

System.out.println("CSE-A");

System.out.println("Roll.NO:24315");

System.out.println("");

**System.out.println("Area of square: " + S1.calculateArea(5));**

**System.out.println("Area of rectangle: " + S1.calculateArea(2, 5));**

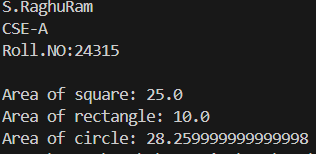
**Circle C1 = new Circle();**

**System.out.println("Area of circle: " + C1.calculateArea(3));**

**}**

**}**

**Output:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Error** | **Rectification** |
| **1** | **Syntax/ Compilation Error** | **Absence of Semicolon** |
| **2** | **Closing Brackets** | **Need to Close the brackets** |
| **3** | **Class Name Error** | **Give the class name correctly** |
| **4** | **Constructor Calling** | **Call the constructor correctly** |

**IMPORTANT POINTS:**

1. We have used both overloading and overloaded methods in these program
2. Shape class with a method calculateArea() that is overloaded for square and triangle subclasses.
3. A subclass circle that overrides the calculateArea() method

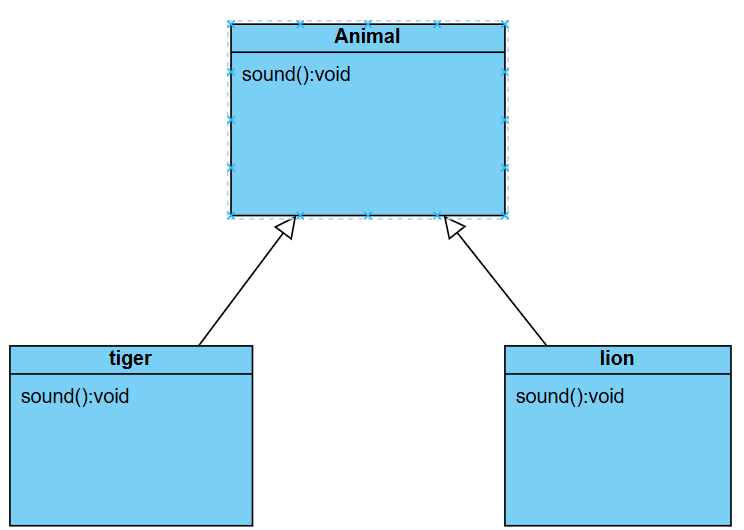
**WEEK-7**

**PROGRAM-1:**

**AIM:**

Write a java program to create an abstract class Animal with an abstract method called sound().Create subclass lion and tiger that extend the animal class and implement extend the Animal class and implement the sound() method to make a specific sound for each animal.

**Class diagram:**

******

**CODE:**

abstract class Animal{

abstract void sound();

public void sounds(){

System.out.println("Sounds of Lions and Tigers");

}

}

class Lion extends Animal{

public void sound(){

System.out.println("Lions roars");

}

}

class Tigers extends Animal{

public void sound(){

System.out.println("Tiger growls");

}

}

class Main {

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println(24315);

System.out.println("CSE-A");

System.out.println("");

Lion L=new Lion();

L.sounds();

L.sound();

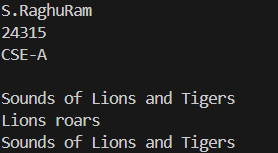
Tigers T=new Tigers();

T.sounds();

}

}

**OUTPUT:**



**IMPORTANT POINTS:**

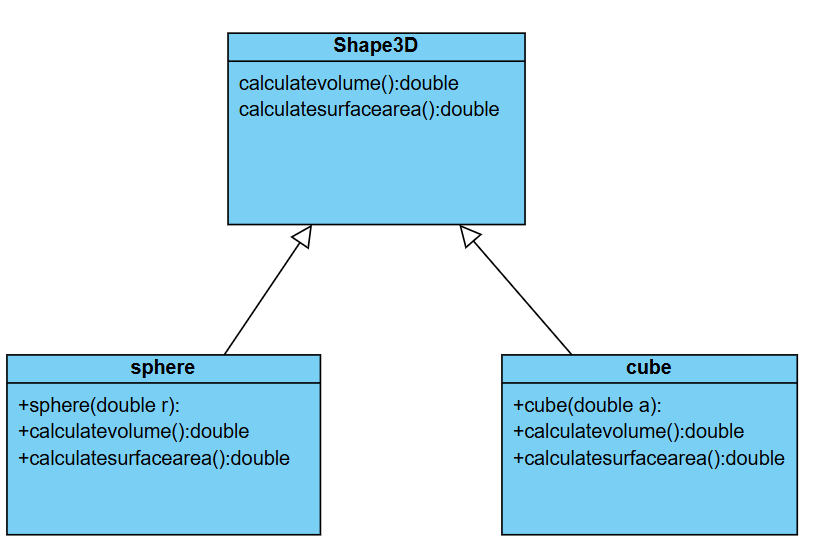
1. Abstract Animal class is used
2. There is a abstract method sound() in it.
3. There are two subclasses tiger and lion which are subclasses and implementing the method sound().

**PROGRAM-2:**

**AIM:**

Write a java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses sphere and cube that extend the Shape3d class and implement the respective methods to calculate volume and surface area of each shape.

**CLASS DIAGRAM:**



**CODE:**

abstract class Shape3D{

abstract void calculateVolume();

abstract void SurfaceArea();

public void display(){

System.out.println("Volume and Surface Area of Sphere and Cube :");

}

}

class Sphere extends Shape3D {

public int r;

public Sphere(int r){

this.r=r;

}

public void calculateVolume(){

double VS=1.3\*3.14\*r\*r\*r;

System.out.println("Volume of a sphere is "+VS);

}

public void SurfaceArea(){

double AS=4\*3.14\*r\*r;

System.out.println("Area of a sphere is "+AS);

}

}

class Cube extends Shape3D {

public int a;

public Cube(int a){

this.a=a;

}

public void calculateVolume(){

double VC=a\*a\*a;

System.out.println("Volume of a sphere is "+VC);

}

public void SurfaceArea(){

double AC=6\*a\*a;

System.out.println("Area of a sphere is "+AC);

}

}

class Main {

public static void main(String[] args){

System.out.println("K.Eswar Balaji");

System.out.println(24202);

System.out.println("CSE-C");

System.out.println("");

Sphere s=new Sphere(4);

s.display();

System.out.println("");

s.calculateVolume();

s.SurfaceArea();

Cube c=new Cube(3);

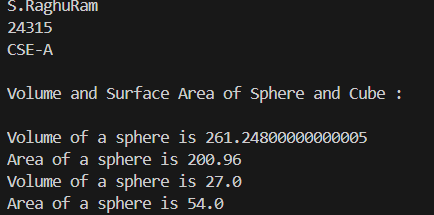
c.calculateVolume();

c.SurfaceArea();

}

}

**OUTPUT:**

****

**ERROR TABLE:**

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| Not using public static void main(String[] args) | error: <identifier> expected. | using public static void main(String[] args) |

**IMPORTANT POINTS / EXPLANATION:**

1. Shape3D is an abstract class.
2. It contains two abstract methods: calculateVolume() and calculateSurfaceArea()
3. Sphere and Cube classes extend Shape3D and override both methods.
4. Constructor is used.

**PROGRAM-3:**

**AIM:**

Write a java program using an abstract class to define a method for printing patterns.

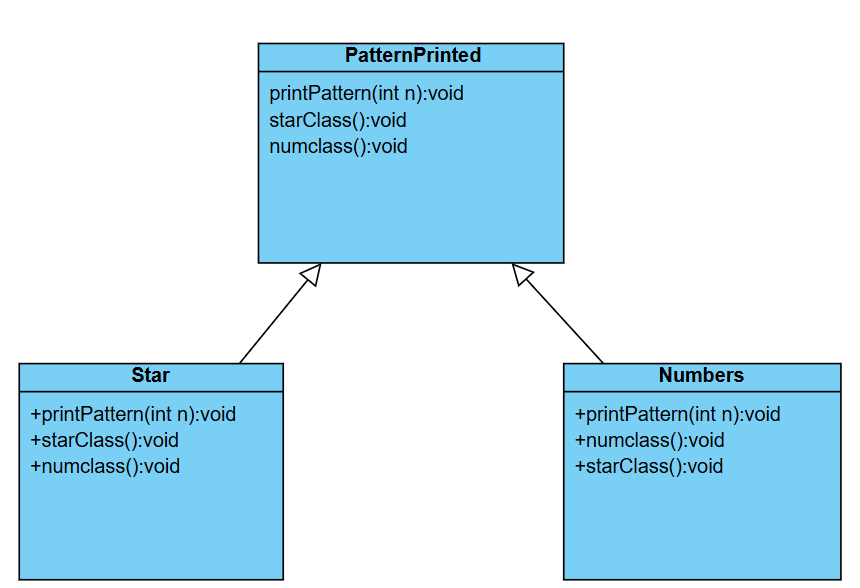
Create an abstract class named PatternPrinted with an abstract method print pattern(int n) and a create a method to display the patterntitle.

Implement two subclasses:

1. star pattern-right angled.

2.numberpatterns-prints a right-angled triangle of increasing numbers.

**CLASS DIAGRAM:**



**CODE:**

abstract class PatternPrinter{

abstract void printpattern(int n);

public void display(){

System.out.println("This method is used to print title name");

}

}

class StarPattern extends PatternPrinter{

void printpattern(int n){

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

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

System.out.print("\* ");

}

System.out.println();

}

}

public void display(){

System.out.println("Star Pattern :");

}

}

class NumberPattern extends PatternPrinter{

void printpattern(int n){

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

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

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

}

System.out.println();

}

}

public void display(){

System.out.println("Number Pattern :");

}

}

class Main {

public static void main(String[] args) {

System.out.println("K.Eswar Balaji");

System.out.println(24202);

System.out.println("CSE-C");

System.out.println("");

StarPattern s = new StarPattern();

s.display();

s.printpattern(5);

System.out.println("");

NumberPattern n = new NumberPattern();

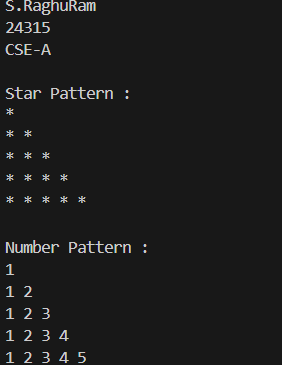
n.display();

n.printpattern(5);

}

}

**OUTPUT:**

****

**ERROR TABLE:**

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| for(int j=1;j<=n;j++) | Incorrect output | for(int j=1;j<=i;j++) |
| Not using System.out.println(); | Incorrect output | Using System.out.println(); |

**IMPORTANT POINTS / EXPLANATION:**

1. In above program an interface Shape is created and it consists of a method getperimeter() and three classes Rectangle, circle and triangle implement the interface shape and provide the functionality of the method getperimeter().
2. Constructers are used in each class for getting values for their respective shapes to calculate perimeter.
3. When object of certain class is created and the method is called it overrides the method and prints it functionality.

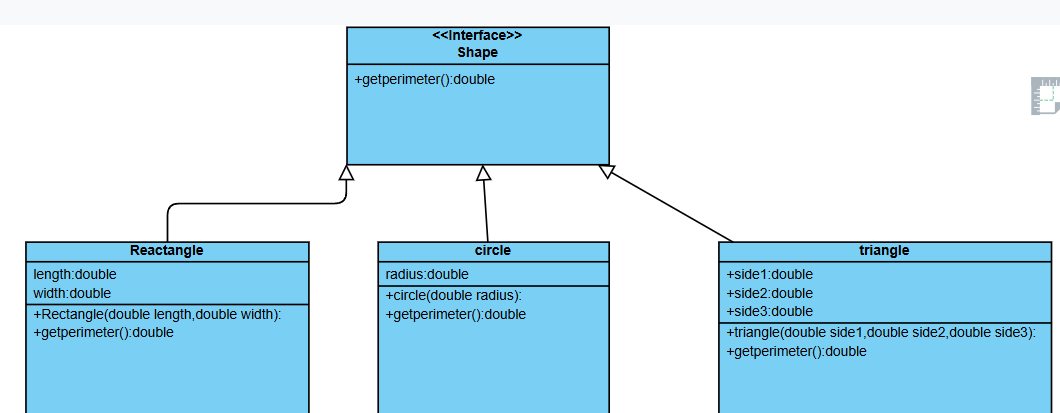
**WEEK-8**

**PROGRAM-1:**

**AIM:**

Write a java program to create an interface Shape with the getPerimeter() method. Create three classes Rectangle, circle and triangle that implements the shape interface. Implement the gerPerimeter() method for each of the three classes.

**CLASS DIAGRAM:**



**CODE:**

import java.util.Scanner;

interface Shape{

public void getPerimeter();

}

class Rectangle implements Shape{

public void getPerimeter(){

Scanner input= new Scanner(System.in);

System.out.println("Enter a number for length of rectangle");

int l=input.nextInt();

System.out.println("Enter a number for width of rectangle");

int w=input.nextInt();

double PR=2\*(l+w);

System.out.println("Perimeter of a rectangle is "+PR);

}

}

class Circle implements Shape{

public void getPerimeter(){

Scanner input= new Scanner(System.in);

System.out.println("Enter a number for radius of circle");

int R=input.nextInt();

double PC=2\*Math.PI\*R;

System.out.println("Perimeter of a circle is "+PC);

}

}

class Triangle implements Shape{

public void getPerimeter(){

Scanner input= new Scanner(System.in);

System.out.println("Enter a value for one side of triangle");

int a=input.nextInt();

System.out.println("Enter a value for another side of triangle");

int b=input.nextInt();

System.out.println("Enter a value for third side of triangle");

int c=input.nextInt();

double PT=a+b+c;

System.out.println("Perimeter of a triangle is "+PT);

}

}

class Main {

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println(24315);

System.out.println("CSE-A");

System.out.println("");

Rectangle R= new Rectangle();

R.getPerimeter();

System.out.println("");

Circle C=new Circle();

C.getPerimeter();

System.out.println("");

Triangle T=new Triangle();

T.getPerimeter();

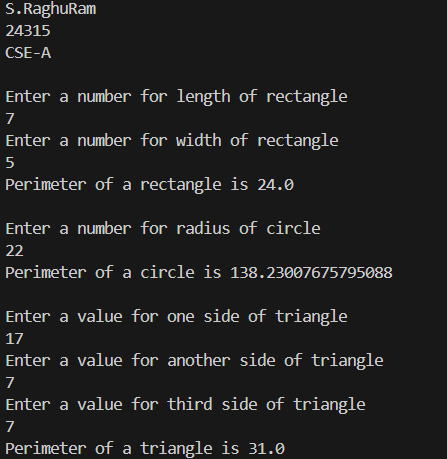
}

}

**ERROR TABLE:**

|  |  |  |
| --- | --- | --- |
| Code | Error | Rectification |
| Math.Pi | error: cannot find symbol  double PC=2\*Math.Pi\*R; | using Math.PI |

**OUTPUT:**

****

**IMPORTANT POINTS / EXPLANATION:**

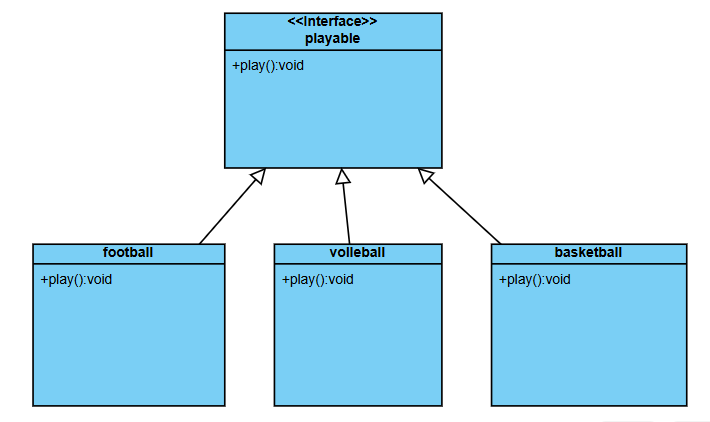
1. Shape is an interface with the method void getPerimeter();
2. All shapes (Rectangle, Circle, Triangle) implement the Shape interface.
3. Each class overrides getPerimeter() method from the Shape interface.
4. Encapsulation: Each class handles its own data and behaviour.
5. Abstraction: The Shape interface hides the specific implementation.

**PROGRAM-2:**

**AIM:**

Write a java program to create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football,Volleyball and Basketball that implement the Playable interface and override the play() method to play the respective sports

**CLASS DIAGRAM:**



**CODE:**

interface playable{

void play();

public static void main(String[] args){

System.out.println("S.RaghuRam");

System.out.println(24315);

System.out.println("CSE-A");

System.out.println("");

Football F=new Football();

F.play();

Volleyball V=new Volleyball();

V.play();

Basketball B=new Basketball();

B.play();

}

}

class Football implements playable{

public void play(){

System.out.println("Playing Football");

}

}

class Volleyball implements playable{

public void play(){

System.out.println("Playing Volleyball");

}

}

class Basketball implements playable{

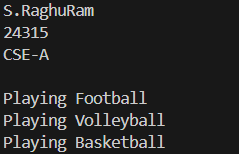
public void play(){

System.out.println("Playing Basketball");

}

}

**OUTPUT:**

****

**IMPORTANT POINTS / EXPLANATION:**

1. Created an interface Playable with a method play().
2. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method