**23CSE111**

**LAB MANUAL**



**Department of CSE**

**Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Verified By :- Name: B.Syam Sunder**

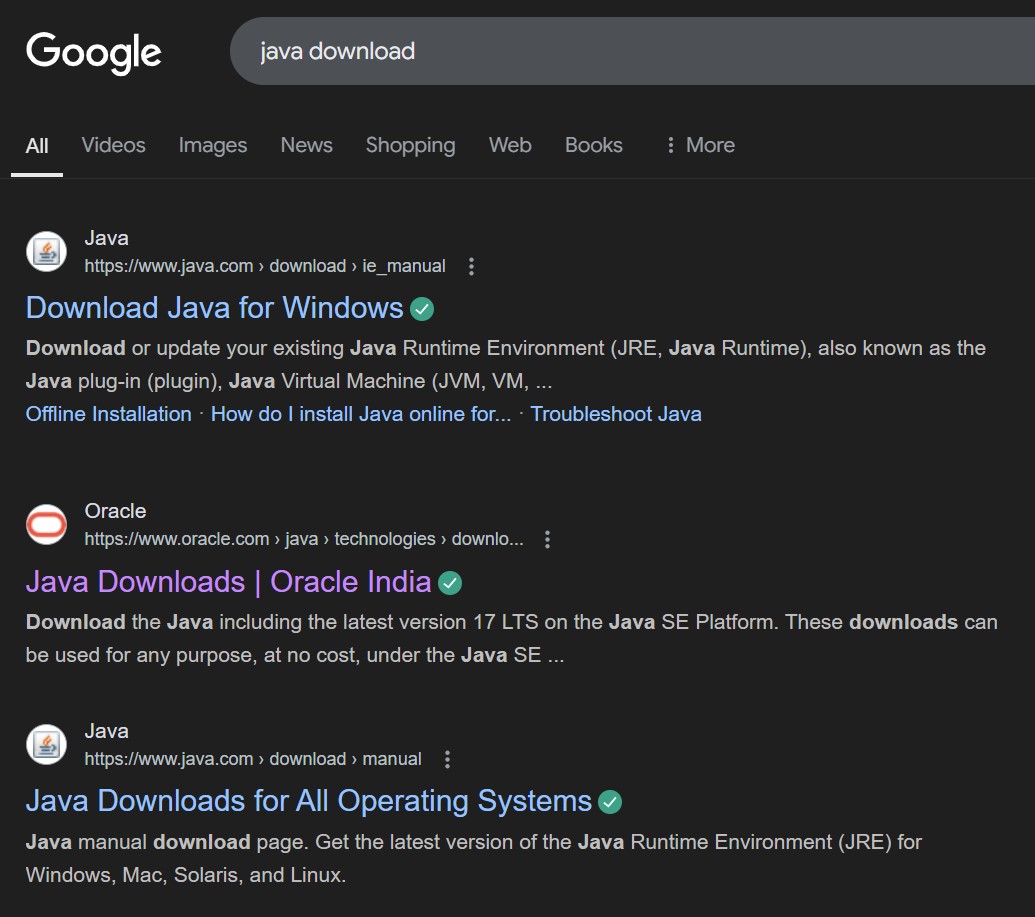
**Roll No: AV.SC.U4CSE24023**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | Programs | Date | Pg:No | Signature |
| 1 | 1. Download and Install Java Software. 2. Write a java program to print message “Welcome to java programming”. 3. Write a java program that prints name,roll number,section of a student. |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

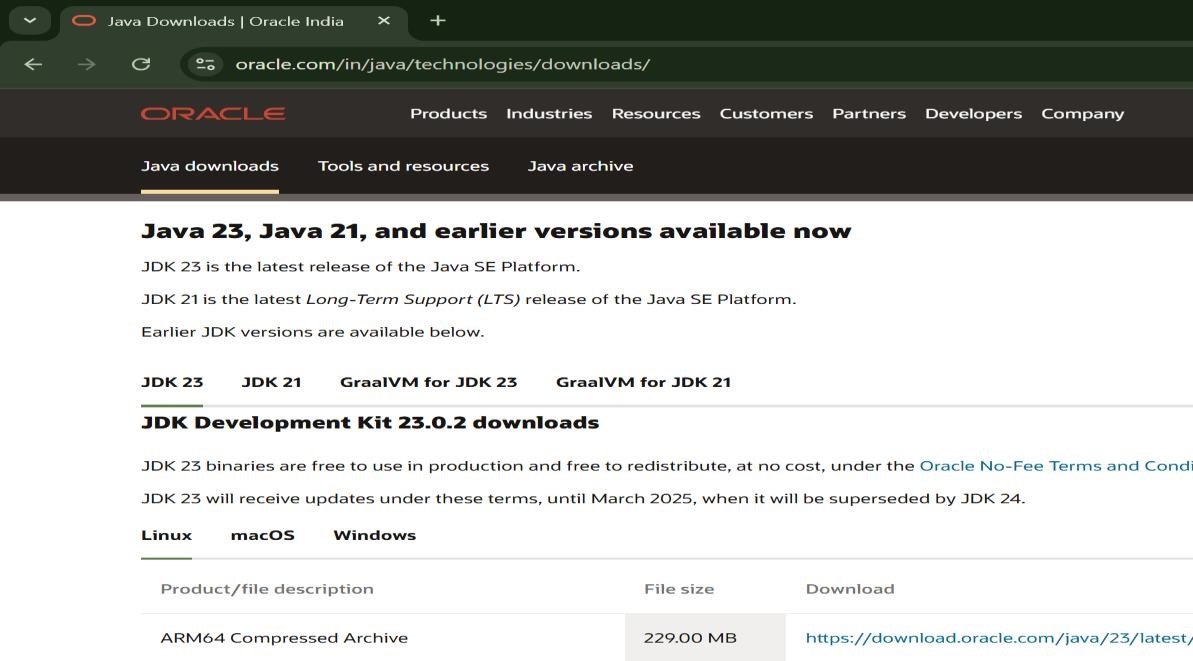
Week 1:-

# Program-1:-

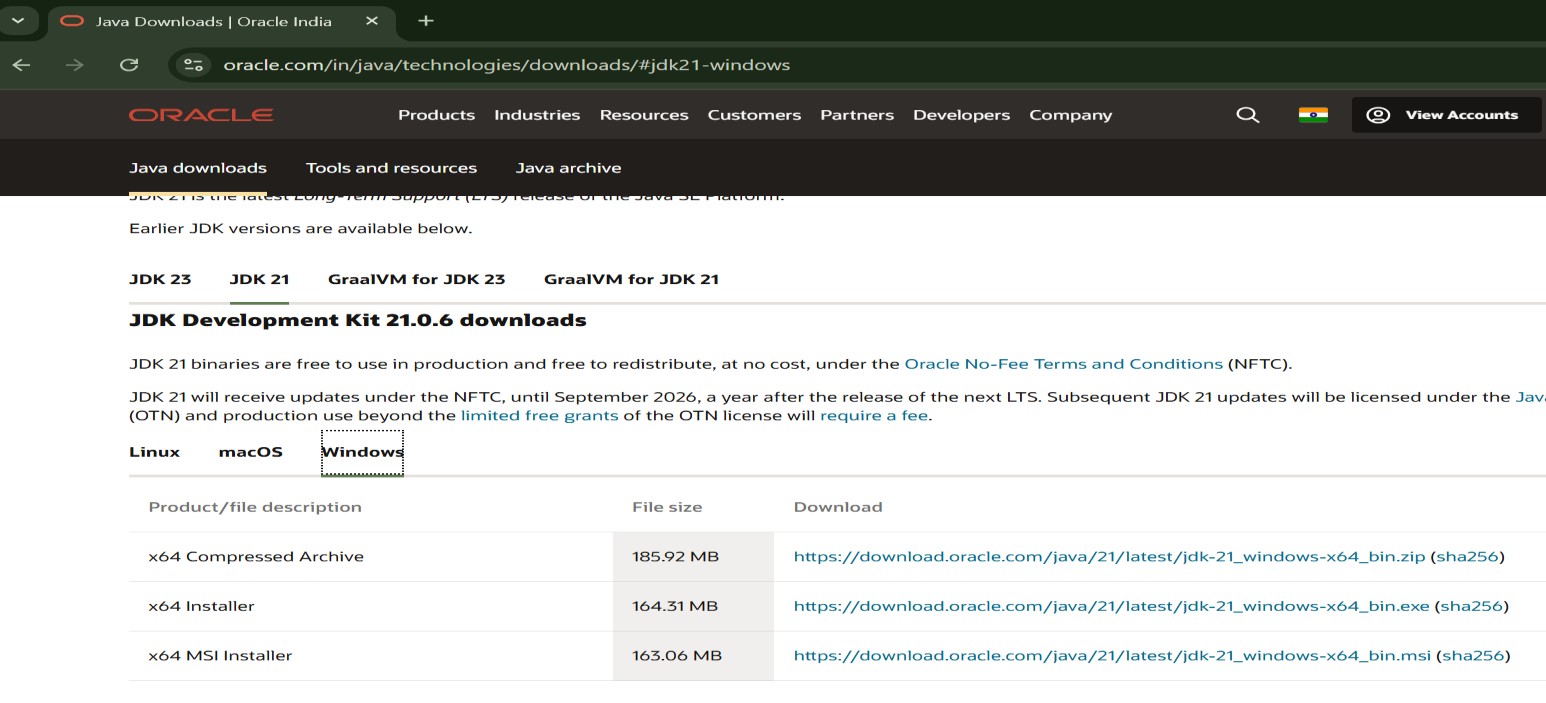
## Aim:-Download and Instal the Java Software Procedure Step-1:- Type Java download in search



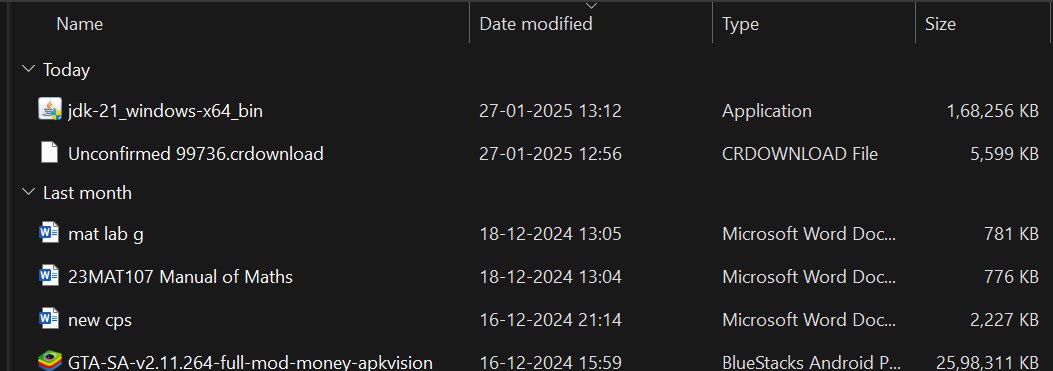
## Step-2:-click on oracle java download and enter into oracle website



## Step-3:-click on JDK21 and click on windows and later click on x64 instalier link to download

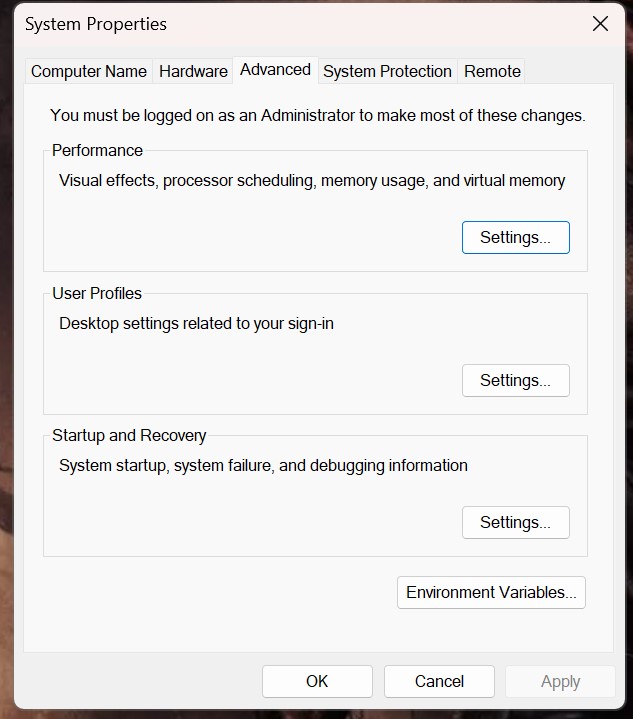


## Step-4:-After completing download click on it’s file and then give permission to install



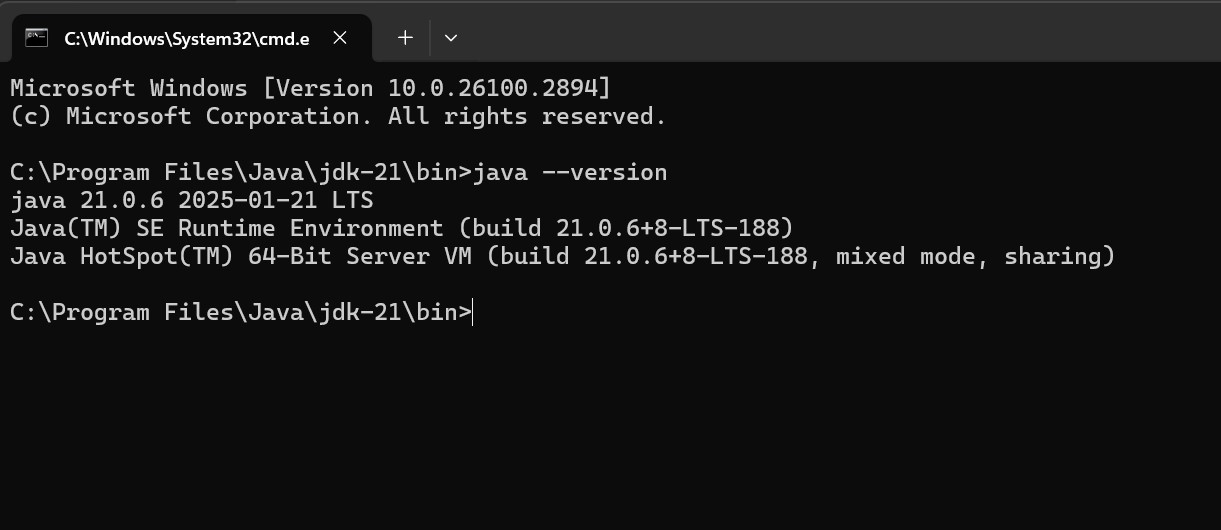
Step-5:-Then go to (This pc) in that click (windows{c}) in that click (Program files) in that click (Java) in that click (jdk-21) in that click (bin)

## Step-6:-Select and copy path of opening the file and then press windows and search System Environmental



Step-7:-After opening Environment variables then past path of opening file in user variable and click on ok

## Step-8:-To verify version open CMD and type java --version



**Program : 2**

## Aim:-write a java program to print[welcome to java programming Input:-

class ex\_1{

public static void main(String[] args){

System.out.println("welcome to java programming");

}

}

## Output:-



### Program : 3

Aim:-write a java program that prints name, roll no, section of the student Input:-

class ex\_2{

public static void main(String[] args){

System.out.println("Name: B.Syam Sunder");

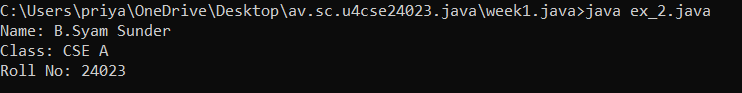
System.out.println("Class: CSE A");

System.out.println("Roll No: 24023");

}

}

### Output:-

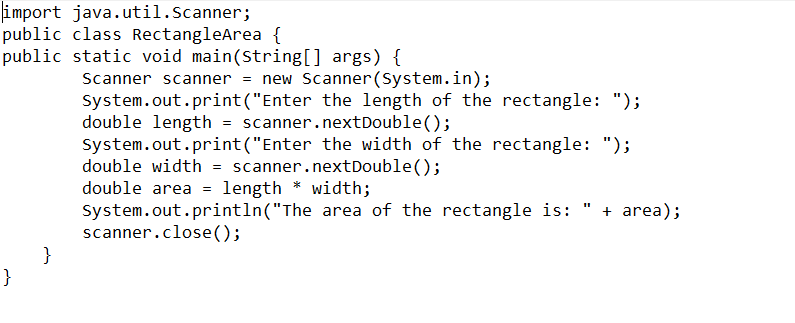


***Week-2***

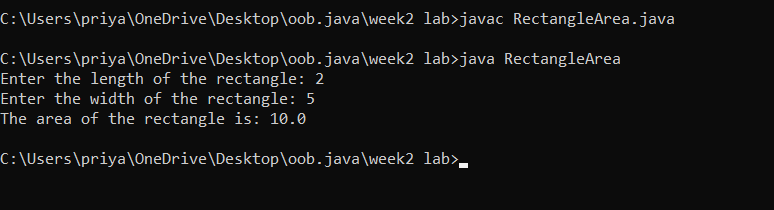
Program-1:

Aim: to write a java program to find area of rectangle

Input:



Output:



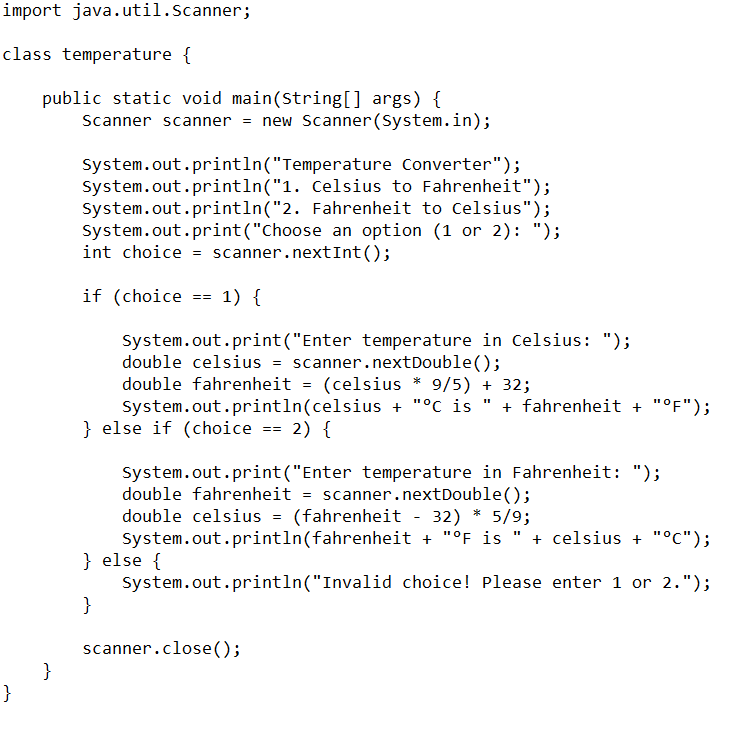
Errors:

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Syntax Error** | Unclosed string literal(“ missing) | Ensure all strings are properly enclosed in double quotes (") |
| **Runtime Error** | Dividing by zero when calculating an aspect ratio | Check for zero before division (if (width != 0) { ... }) |

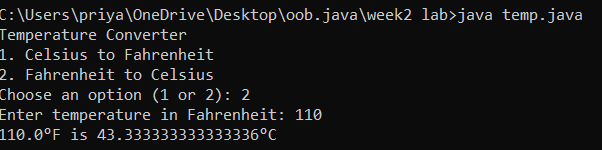
Program-2

2.write a java program to convert the temperature from celcius to farhienheat:

Program:



Output:



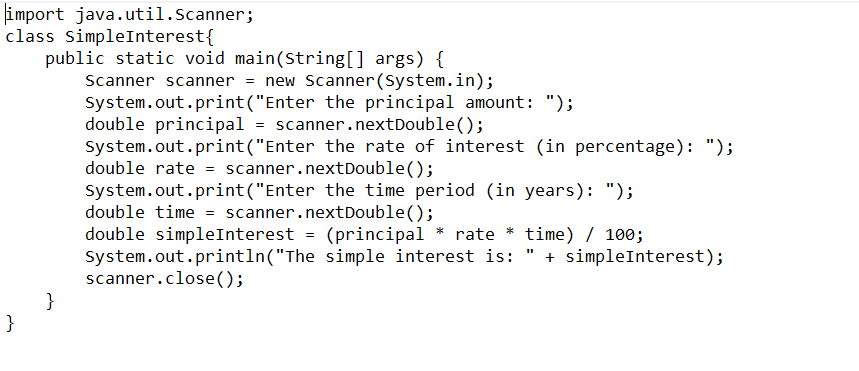
Error:

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Syntax Error** | Forgot to write ; (semicolon) | Ensure every statement ends with a semicolon (;) |
| **Runtime Error** | Dividing by zero when calculating an aspect ratio | Check for zero before division (if (width != 0) { ... }) |
| **Variable Declaration Error** | Wrong variable declaration | Use correct data types and proper syntax |

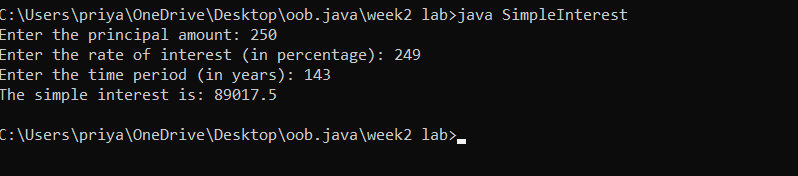
Program-3

3.write a java program to cacluate the simple intrest

Program:



Output:



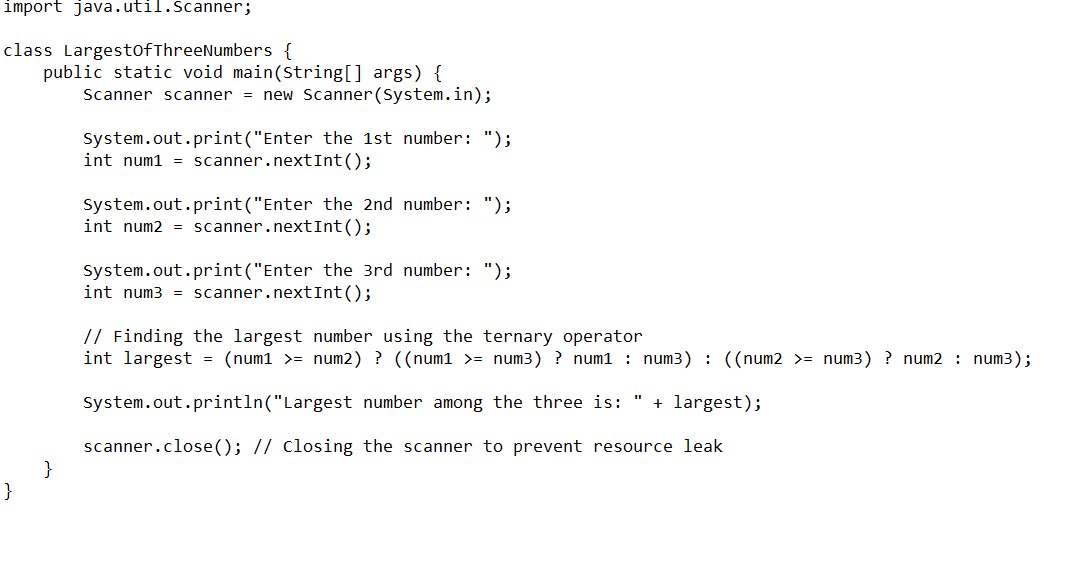
Error:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
| **Error Type** | **Description** | | **Correction** | |
| **Syntax Error** | Missing semicolon (;) after System.out.println() | | Add ; at the end of System.out.println() statements | |
| **Data Type Error** | int used instead of double for time (t) | | Change int t = read.nextInt(); to double t = read.nextDouble(); | |
| **Type Mismatch** | int r = read.nextDouble(); (assigning double to int) | | Change int r to double r for correct data type | |

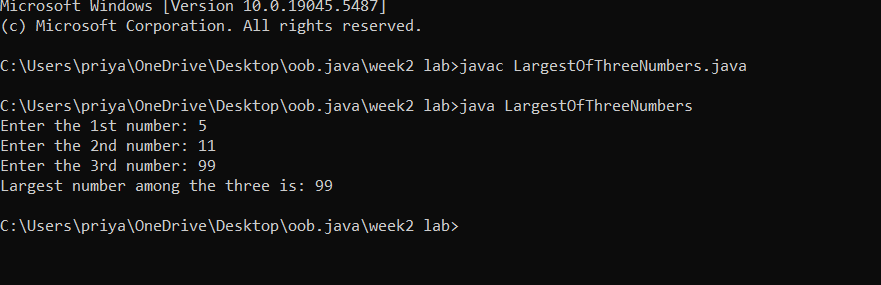
Program-4

Write a java program to find the largest of three numbers using terinary operator

Program:



Output:



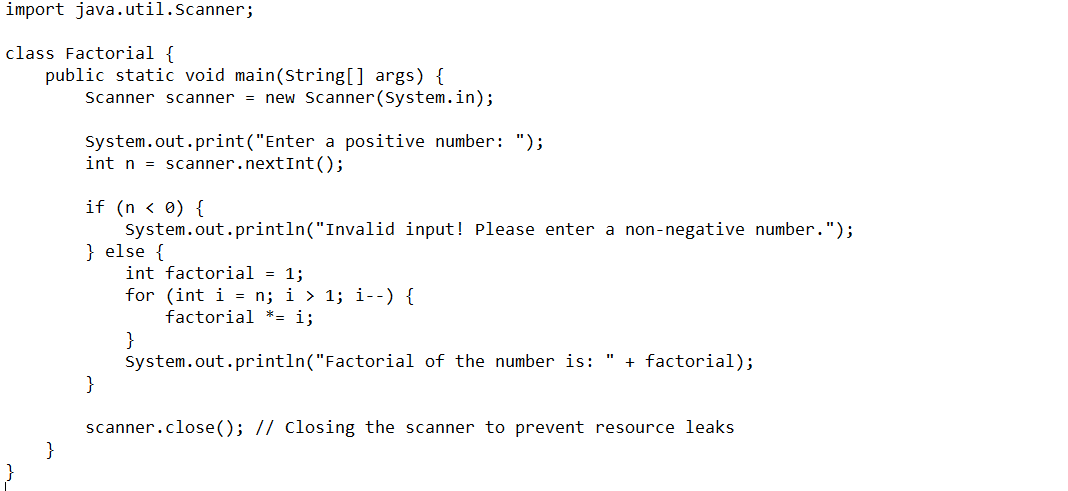
Error:

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Syntax Error** | Missing space in output: "Largest Number of 3 numbers is" + largest | Change to "Largest Number of 3 numbers is " + largest (add space before largest) |
| **Logical Error** | No read.close(); to free resources | Add read.close(); at the end of the program |
| **Input Handling Issue** | No prompt for invalid input (e.g., non-integer values) | Add input validation using if (read.hasNextInt()) before reading values |

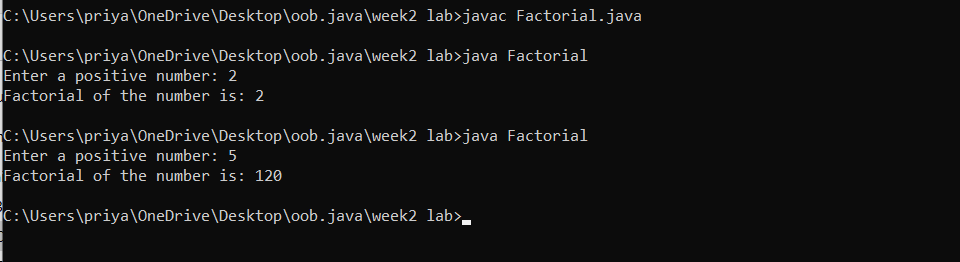
Program-5

5.Write a java program to find the factorial of a number

Program:



Output:



Error:

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Syntax Error** | do keyword mistakenly placed before for loop | Remove do before for(int i=n; i>=1; i--) |
| **Logical Error** | if(n<0) check comes after the factorial calculation | Move if(n<0) check before the loop to prevent calculation |
| **Resource Leak** | Scanner not closed | Add read.close(); at the end of the program |

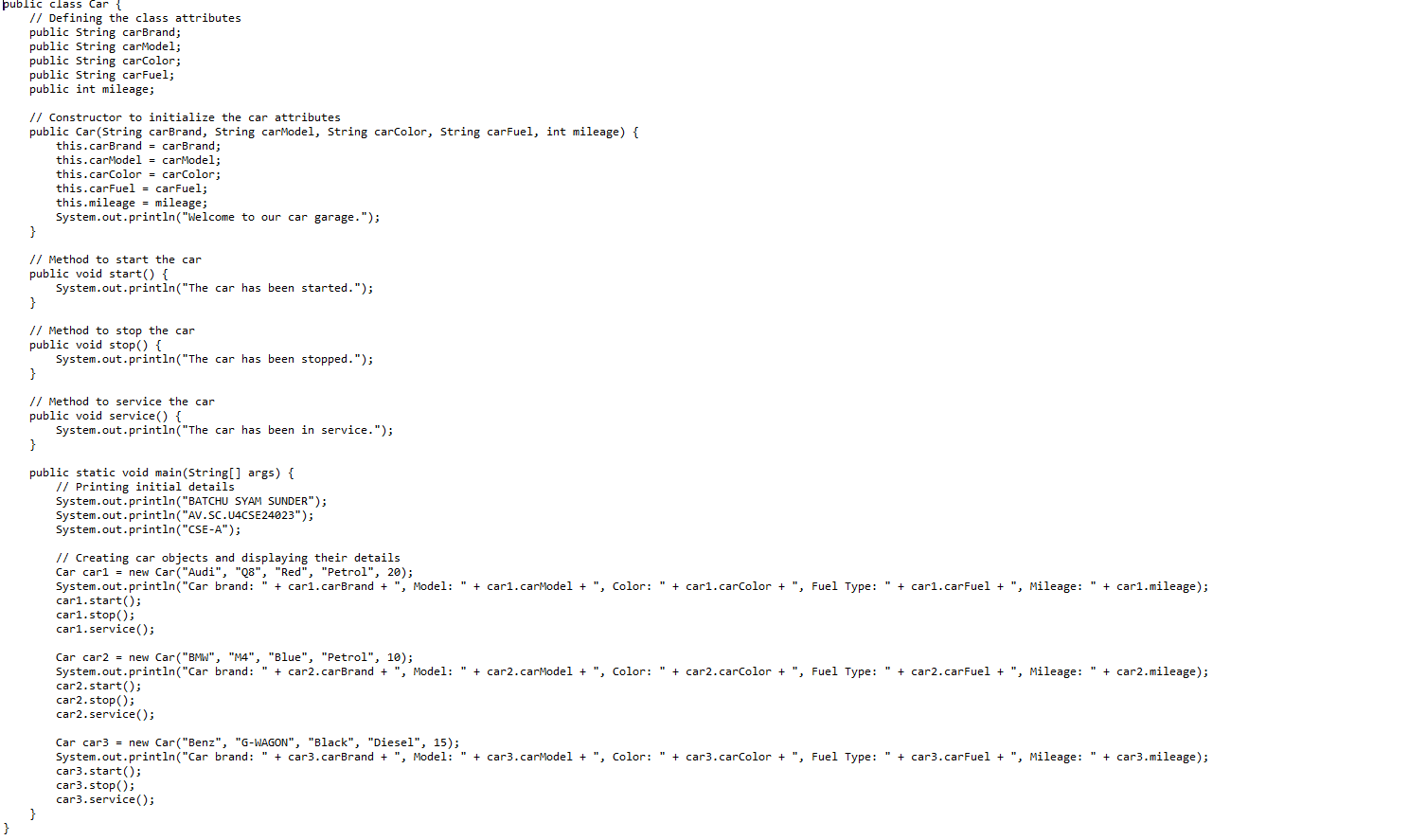
Week-3

1. Create the java program for the cars with constructor and methods

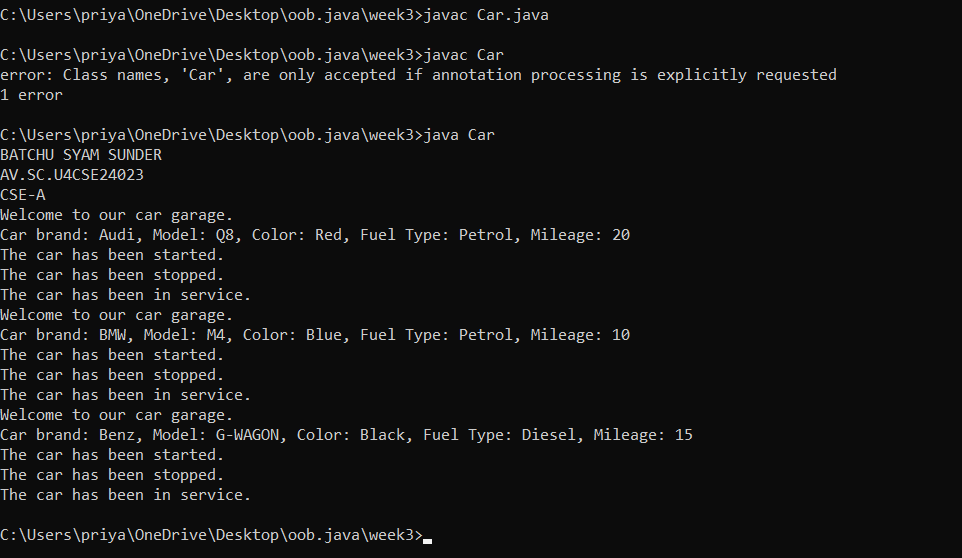
Class diagram:

|  |
| --- |
| **Car** |
| * carColor: String |
| * carBrand: String |
| * fuelType: String |
| * topSpeed: int |
| + Car(String,String,String,int) |
| + startRacing() |
| + endRace() |

Program:



Output:



**Error:**

|  |  |  |
| --- | --- | --- |
| Error Type | Incorrect Code | Corrected Code |
| Class Naming Issue | class main{ | class Main{ |
| Incorrect Object Description | "Our first car is "+car2.car\_brand; | "Our second car is "+car2.car\_brand; |

**Explanation:**

here I have created a class named car and created a constructor with attributies car brand ,car color,car model ,car fuel and created methods named start(),stop(),service().and created objects named car1,car2,car3.and assigned values and displayed the desired output.

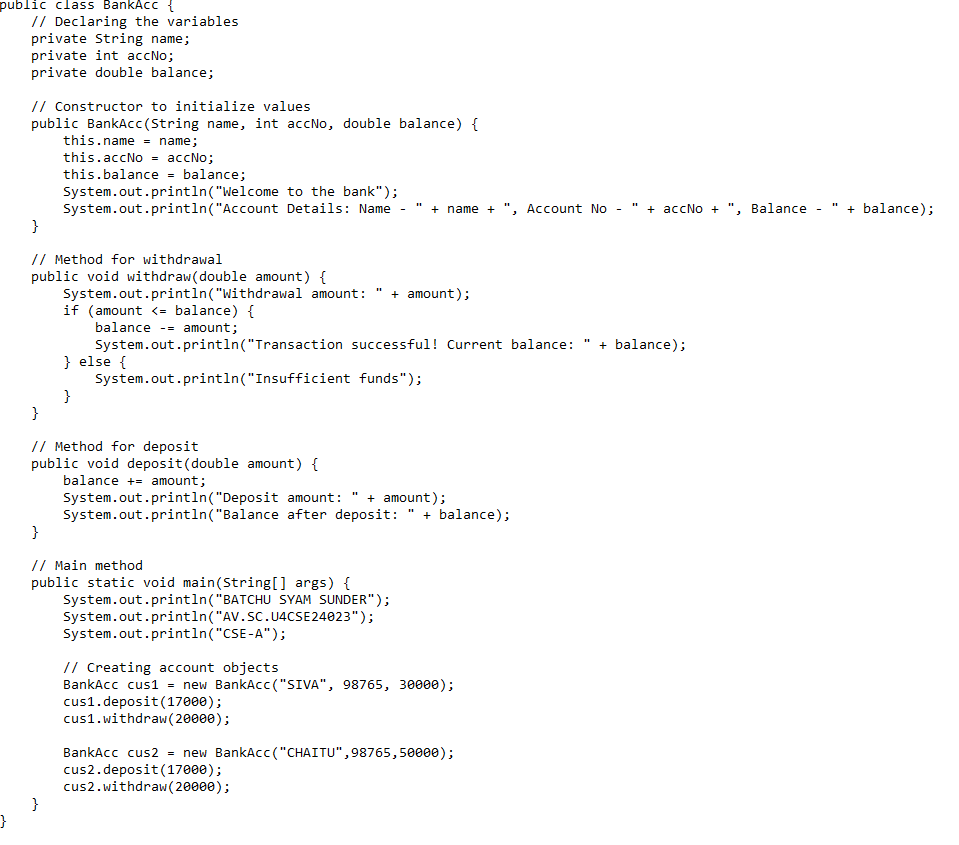
Program-2

1. Create the java program to withdraw and deposit money in the bank account.

Program:

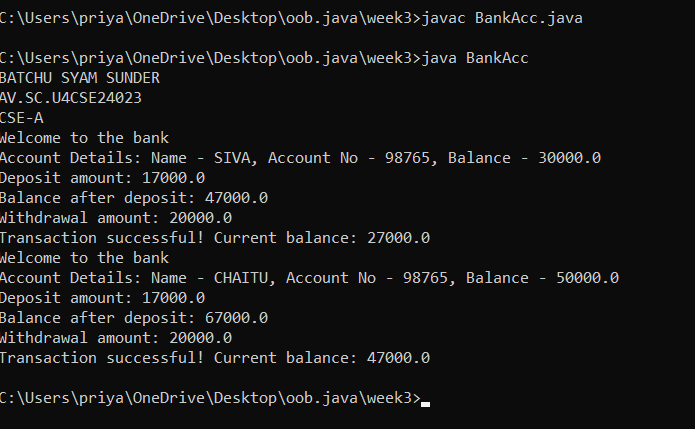
Class diagram:

|  |
| --- |
| BankAcc |
| * currentAmount: double |
| + BankAccount(initialAmount:double) |
| +deposit(amount: double):void |
| +withdraw(amount: double):void |
| +getCurrentAmount():double |

Program: 

Explanation:

Here I have created a class named BankAcc and created a constructor with attributies name,accNo and balance and created a method withdraw ()with deposit() parameters double amount and and used conditional statements to withdraw amount and I have created two objects named cus1 and cus2 and displayed the desired output by calling the methods.

Output: 

**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected code** |
| **ClassName Capitalization** | class Bankaccunt | class BankAccount (Java follows PascalCase for class names) |
| **Object Naming Issue** | BankAccount person-1 (hyphen is not allowed) | BankAccount person1 |
| **Missing Semicolon** | System.out.println ("Balance is "+ person-1.deposit (50,000)) | System.out.println ("Balance is "+ person1.deposit (50000)); (semicolon added) |

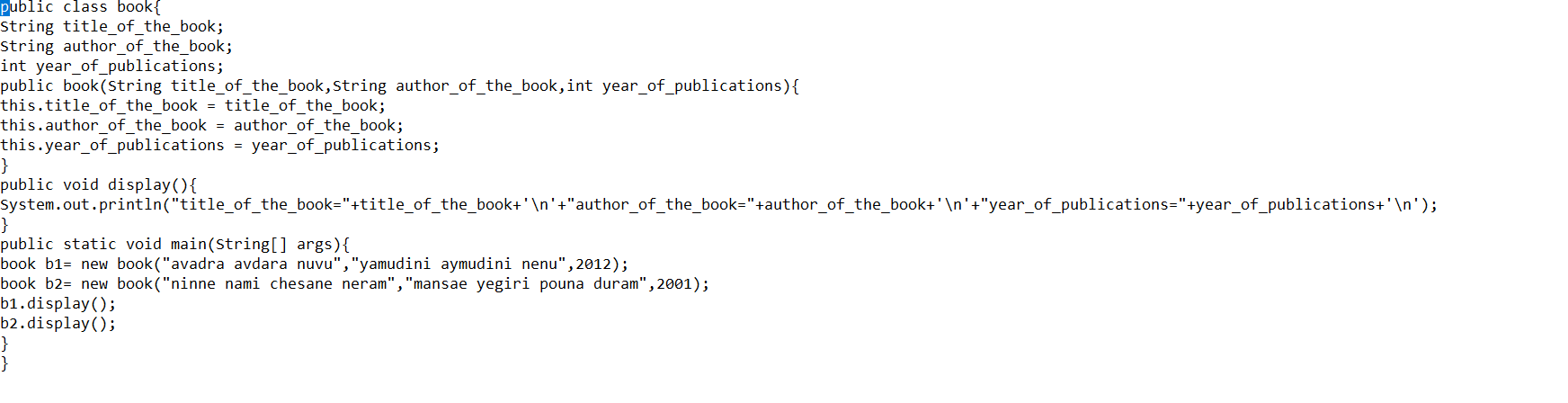
Week-4

1. Create the java program for the books by using the constructor and display its details using methods.

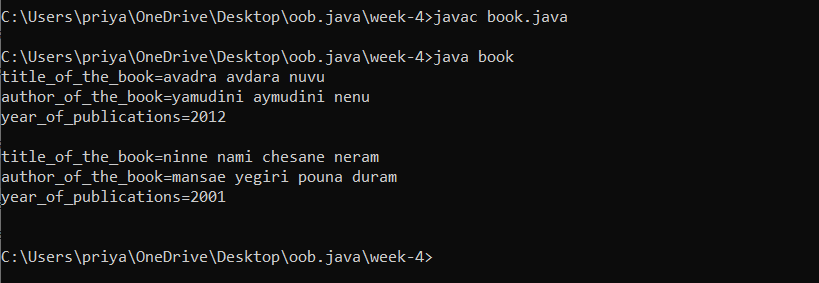
Program

Class diagram:

|  |
| --- |
| **Book** |
| - title: String  - author: String  - yearOfPublication: int |
| + Book(title: String, author: String, yearOfPublication: int)  + displayDetails(): void |



Output:



**Errors:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Error Type** | |  | | --- | | **Incorrect Code** |  |  | | --- | |  | | **Corrected Code** |
| **Class Name Capitalization** | public class book | public class Book (Java follows PascalCase for class names) |
| **Constructor Name Mismatch** | new book(...) | new Book(...) (Constructor name must match class name) |

**Explanation:**

Here I created a class book and I have took there attributies named title of the book ,author of the

book ,and the year of publications and I created a constructor and I crated a new class named main

and created two objects named b1 and b2 and displayed the details which I have stored in this

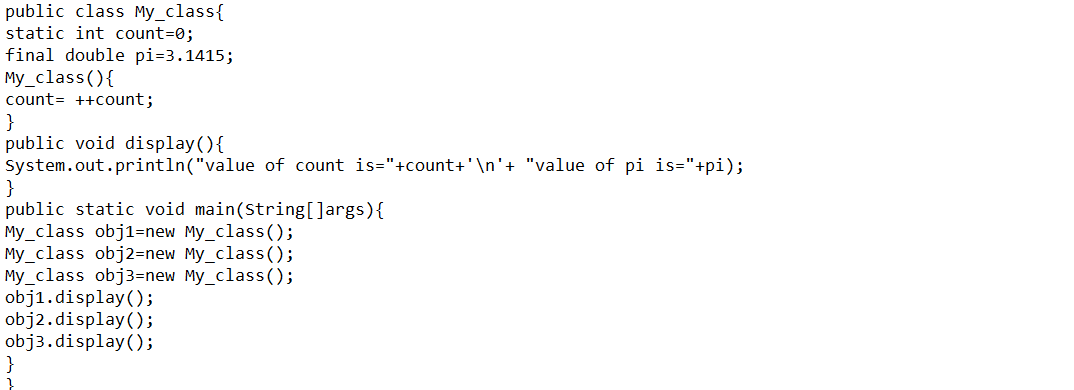
object.

1. Program to explain the final and the static variables.

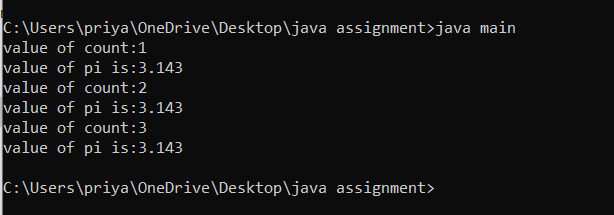
Program:

Class diagram:

|  |
| --- |
| **MyClass** |
| - Count: int  + pi: double |
| + MyClass()  + getCount(): int |



Output:



**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **Attempt to Modify final Variable** | pi = 3.14; (if added inside the constructor or method) | Remove this line (final variables cannot be reassigned) |
| **Incorrect Class Name** | public class Myclass | public class MyClass (Java follows PascalCase for class names) |

Explanation:

here I hava created class named My\_class . and I inistalizes the value of pi =3.14 and count=0.

And I gave a operation that count++ .i created a method named display() and wrote a print statement . and I created a new class main and created 3 new objects. And printed them.

**Week-5**

**Program 1:**

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

**Class diagram:**

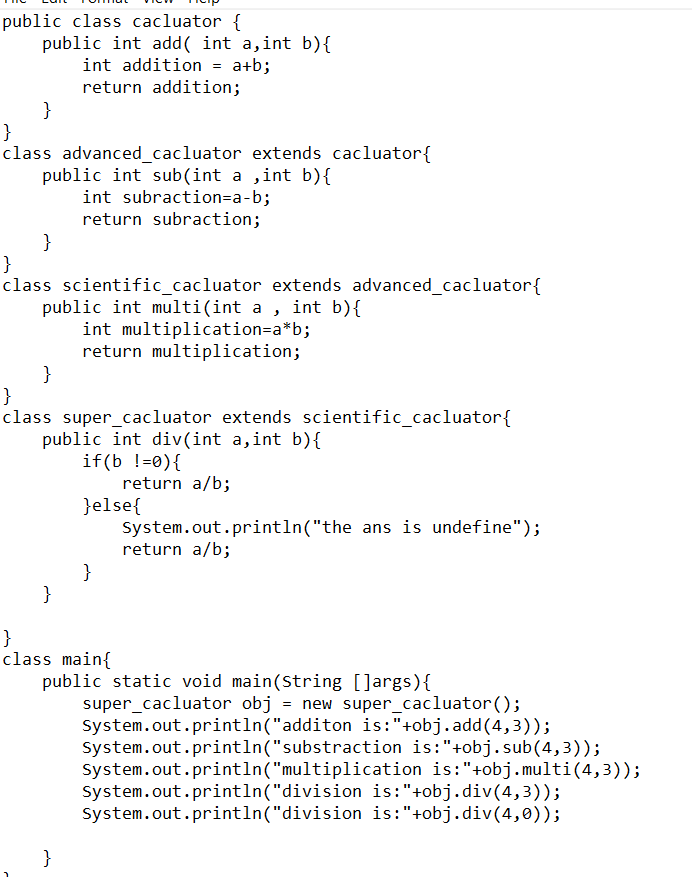
**CLASS DIAGRAM:-**

|  |
| --- |
| **CLASS ADDITION** |
| **+add(int a, int b):int** |

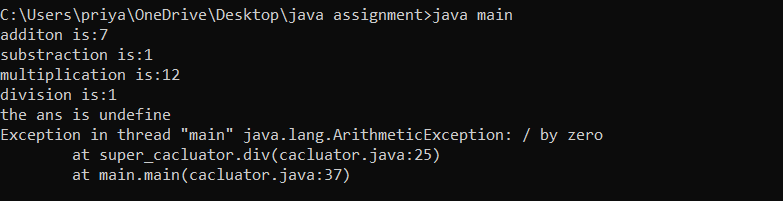
|  |
| --- |
| **Class Subtraction** |
| **+sub(int a, int b):int** |

|  |
| --- |
| **Class Multiplication** |
| **+mult(int a, int b):int** |

|  |
| --- |
| **Class Division** |
| **+div(int a, int b):int** |



Output:



**Error table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Error Type** | **Reason for error** | **Rectification** |
| **1.** | Constructive error | Invalid method name declared | Created class name |
| **2.** | Syntax error | Haven’t included ‘;’ | Added ‘;’ |
|  |  |  |  |

**Explation:**

**Here I have created a class named calculator and created a method add() using return type int and and used multi inheritance and created subclasses**

**Advanced calculator and it extends scientific calculator and it extends super calculator and used the methods sub(),multi(),div() using return types and created a new object named obj and called all the methods and printed the desired output.**

**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 truck and they need a program to

Store details about each vehicle, such as brand and speed

Cars should have an additional property: number of doors

Bikes should have a property indicating whether they have gears or not

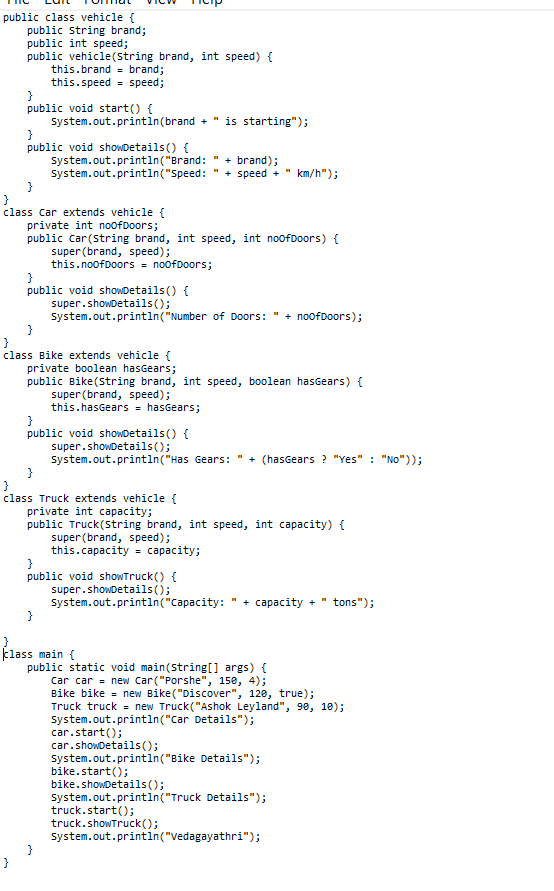
The system should also include a function to display details about each vehicle

And indicate when a vehicle is starting.

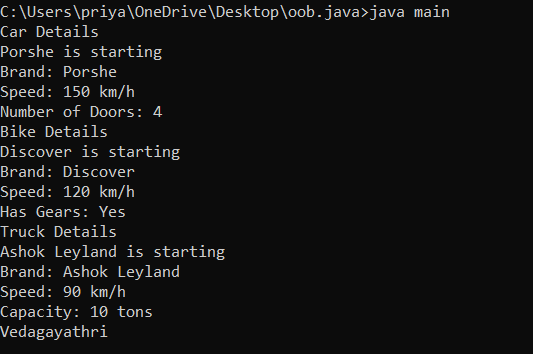
**Program:**

**Class diagram:**



****

Output:



**Explanation:**

Here I have created a class named vehicle and it consists of subclasses car

Bike and truck and I have used hiracy inheritance and the main class consists of attributies brand and speed and I created a constructor using these attributies and it consists of methods start() and showDetails() and it overrides and car sub class consists of attributies noof doors and bike subclass consists of attribuite hasgears and truck subclass consists of attributie capacity and I used super constructor to inherit the values and I have created a new class main and I have created objects named car ,bike and truck and I have called all the methods which I have written in the classes and I have displayed the desired output.

Error table:

|  |  |  |  |
| --- | --- | --- | --- |
| S No | ErrorType | Cause | Rectification |
| 1 | SyntaxError | Semicolon missing | Added ; |

***WEEK-6 PROGRAMS JAVA***

***Program 1:***

**1)Write a java program to create a vechicles class with a method displayinfo()override the method in the car subclass to provide specific information about a car**

**Company**

**Model**

**Price**

**Seating capacity**

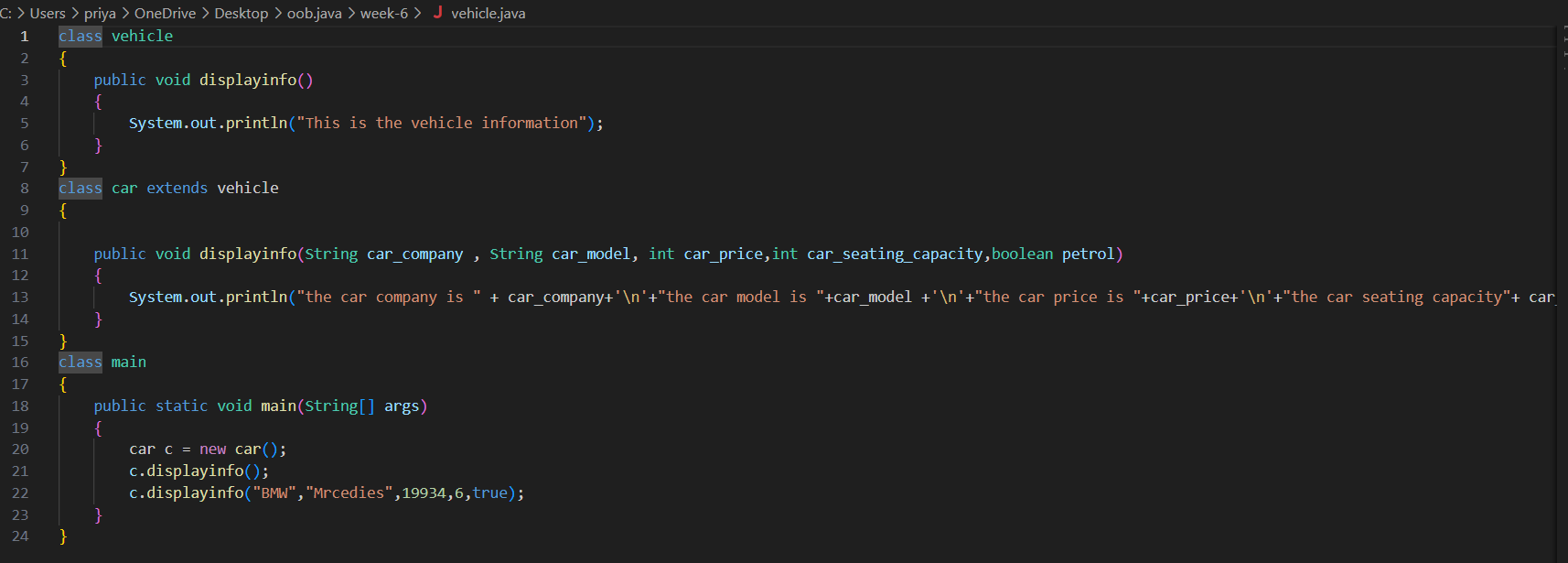
**Petrol or not**

**Program:**

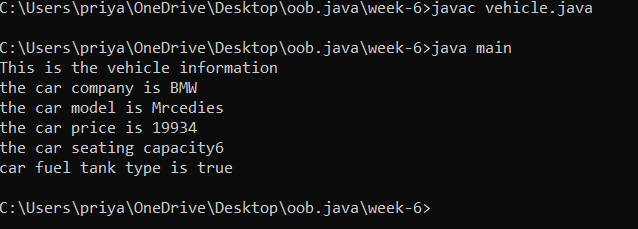
**Class Diagram:**

|  |
| --- |
| **Vehicle** |
| **+displayInfo(): void** |

|  |
| --- |
| **car** |
| **+displayInfo(): void** |

****

**Output:**

****

**Explanation:**

Here I have created a class named vehicle and it extends a subclass vehicle and It overrides the method displayinfo() and they consists of different parametes and I have created an another class named main and I have created an object c and I have called the method displayinfo() and I have displayed the desired output.

**Error table:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **ExpectedError** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**2Q)A college is developing automated admission system that verifies students eligibility for UG and PG programs .Each program has different eligibility criteria based on the students percentage in their previous qualification.**

**1) UG admission require minimum 60%**

**2)PG admission require minimum 70%**

**Program:**

**Class diagram**

|  |
| --- |
| **adm** |
| **elg():void** |

|  |  |
| --- | --- |
| **ug** | **pg** |
| **+elg():void** | **+elg():void** |
|  |  |

**Program:**

**import java.util.Scanner;**

**class College {**

**String name;**

**int qualification;**

**int percentage;**

**College(String name, int qualification, int percentage) {**

**this.name = name;**

**this.qualification = qualification;**

**this.percentage = percentage;**

**}**

**public void Eligibility() {**

**System.out.println("Name: " + name + ", Qualification: " + qualification + ", Percentage: " + percentage);**

**System.out.println("The candidate is a fluke");**

**}**

**}**

**class UG extends College {**

**UG(String name, int qualification, int percentage) {**

**super(name, qualification, percentage);**

**}**

**public void Eligibility() {**

**System.out.println("Name: " + name + ", Qualification: " + qualification + ", Percentage: " + percentage);**

**System.out.println("The candidate is eligible for UG");**

**}**

**}**

**class PG extends College {**

**PG(String name, int qualification, int percentage) {**

**super(name, qualification, percentage);**

**}**

**public void Eligibility() {**

**System.out.println("Name: " + name + ", Qualification: " + qualification + ", Percentage: " + percentage);**

**System.out.println("The candidate is eligible for PG");**

**}**

**}**

**public class Main {**

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

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

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

**String name = input.nextLine();**

**System.out.println("Enter your qualification (e.g., 12 for high school, 10 for 10th, etc.):");**

**int qualification = input.nextInt();**

**System.out.println("Enter your percentage:");**

**int percentage = input.nextInt();**

**input.close()**

**College candidate;**

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

**candidate = new PG(name, qualification, percentage);**

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

**candidate = new UG(name, qualification, percentage);**

**} else {**

**candidate = new College(name, qualification, percentage);**

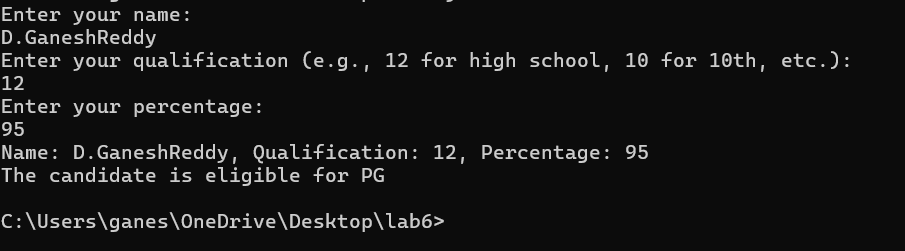
**}**

**candidate.Eligibility();``**

**}**

**}**

**Output:**



**Error table:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **ExpectedError** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**Explanation:**

**3Q)Create a calculator class with overloading methods to perform addition**

**1)Add two doubles**

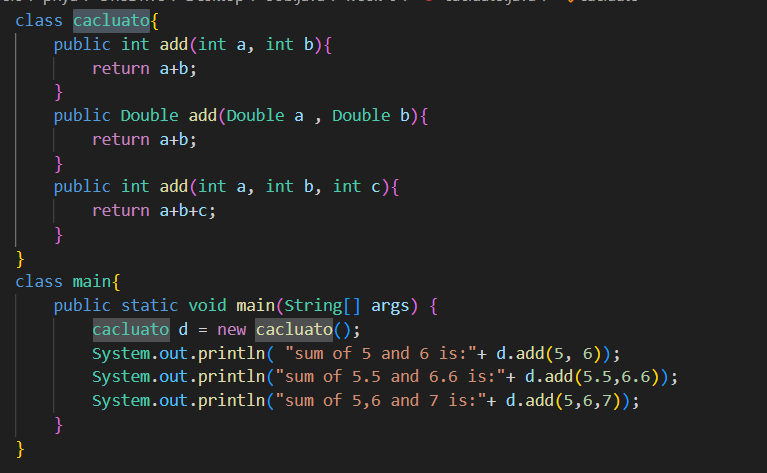
**2)Add two integer**

**3)Add three integer**

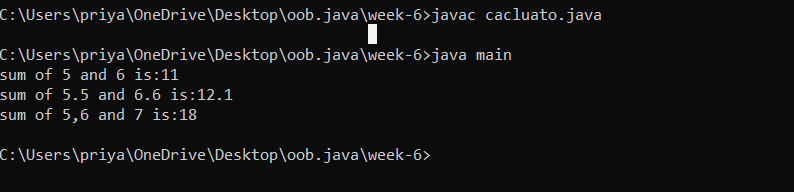
**Program:**

**Class diagram:**

|  |
| --- |
| **cacluato** |
| **+add(int a,int b):int**  **+add(double a,double b):double**  **+add(int a,int b,int c):int** |

****

**Output:**

****

**Error table:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside theconstructor** | **We cannot pass the valuesinside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

**Explanation:**

Here I have created the class named calculato and I have created method named add() and I have used overloading concept here with different parameters and I have created an another class named main and I have created a new object named d and called all the methods and displayed all the desired output.

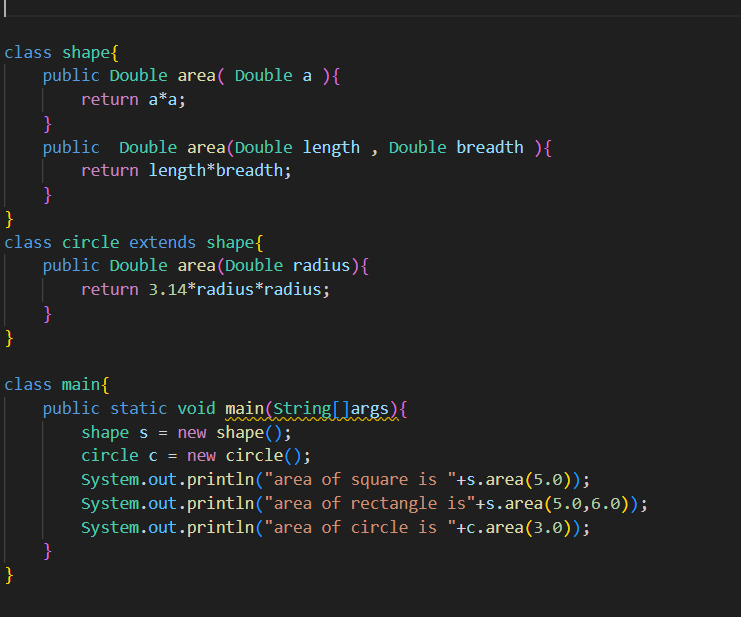
**4Q)Create a shape class with a method calculate area that is overloaded for different shapes Square,Rectangle then create a sub class circle that overerides the calculate area methods for a circle.**

**Program:**

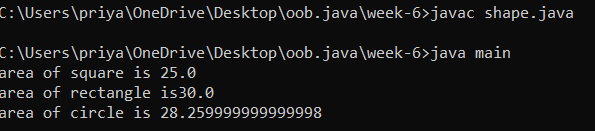
**Classdiagram:**

|  |
| --- |
| **shape** |
| **+calarea(float side):float**  **+calarea(float l,float b):float**  **+calarea(float c):float** |

|  |
| --- |
| **Circle** |
| **+calarea(double r):double** |

****

**Output:**

****

**Error table:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the valuesinside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

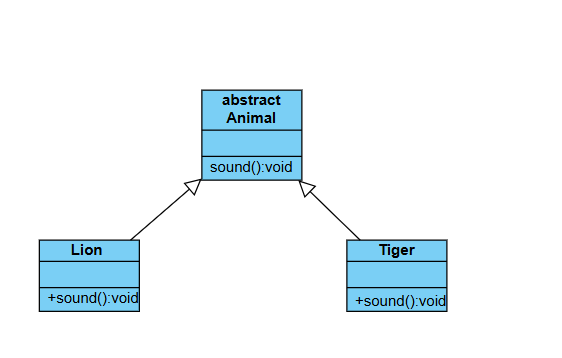
**Explanation:**

Here I have created a class named shape and it consists of method area() with different parameters and created a another subclass named circle and the method area()overrides and I have created an another class main and created a new objects named s and c and I have called all the methods and displayed the desired output.

WEEK -7

1.AIM: Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.

CLASS DIAGRAM:



CODE:

|  |
| --- |
| abstract class animals{  abstract void sound();  }  class lion extends animals{  public void sound(){  System.out.println("lion roars");  }  }  class tiger extends animals{  public void sound(){  System.out.println("tiger growls");  }  }  class main{  public static void main(String[]args){  System.out.println("BATCHU SYAM SUNDER\nAV.SC.U4CSE24024\nCSE-A");  tiger t = new tiger();  t.sound();  lion l = new lion();  l.sound();  }  } |

OUTPUT:



ERROR: **Error table:**

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Fix** |
| **Method name mismatch** | **Wrong case used in method name** | **Change Eat() to eat()** |

IMPORTANT POINTS:

1. Animal class is declared as an abstract.

2.the method sound has no parameters in super class and subclasses.

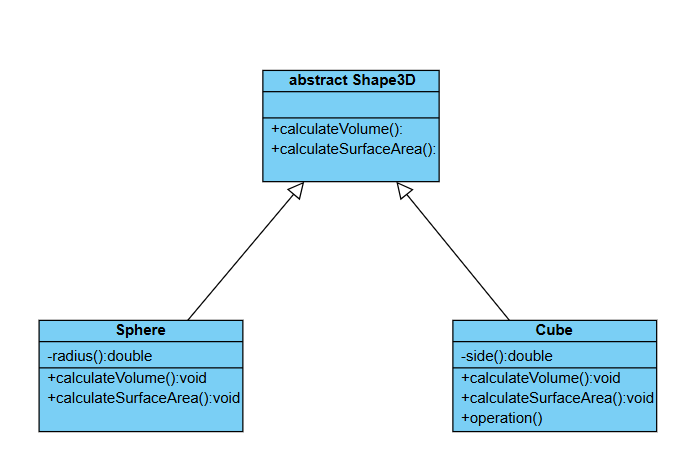
3.each subclass extends from super class.

4.method overriding is happened here

5.the method sound declared as abstract in super class.

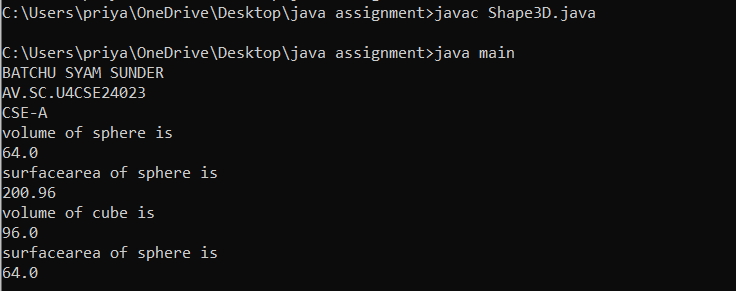
2.AIM: Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

CLASS DIAGRAM:



CODE:

|  |
| --- |
| abstract class Shape3D{   public abstract void calculatevolume();  public abstract void calculatesurfacearea();  }  class sphere extends Shape3D{   private Double radius;  sphere(Double radius){  this.radius = radius;  }   public void calculatevolume(){  System.out.println( (4/3) \* radius \* radius \* radius);  }   public void calculatesurfacearea(){  System.out.println((4.0)\*3.14\*radius\*radius);  }  }  class cube extends Shape3D{   private Double length;  private Double breadth;  Double height;  cube(Double length,Double height,Double breadth){  this.length = length;  this.breadth = breadth;  this.height = height;  }  public void calculatevolume(){  System.out.println( length\*breadth\*height);  }   public void calculatesurfacearea(){  System.out.println( (2.0)\*(length\*breadth+breadth\*height+length\*height));  }  }  class main{  public static void main(String[]args){  System.out.println("BATCHU SYAM SUNDER\nAV.SC.U4CSE24023\nCSE-A");  sphere s = new sphere(4.0);  cube c = new cube(4.0,4.0,4.0);  System.out.println("volume of sphere is");  s.calculatevolume();  System.out.println("surfacearea of sphere is");  s.calculatesurfacearea();  System.out.println("volume of cube is");  c.calculatesurfacearea();  System.out.println("surfacearea of sphere is");  c.calculatevolume();  }  } |

Output:

**Error table:**

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Fix** |
| **Calculation Error** | **Missing π (3.14) in volume formula** | **Add \* 3.14 in volume formula** |
| **Redundant Code** | **Unused variables in cube class** | **Remove unused breadth and height** |

**Important Points and Explanation:**

* Animal is an abstract class with an abstract method eat().
* Lion and Tiger extend Animal and override eat().
* Both classes have their own sound() method.
* Objects are created in the main() method.
* Methods eat() and sound() are called to show output.

AIM: Write a java program using an abstract class to define a method for pattern printing.

create an abstract class named PatternPrinter with an abstract method PrintPattern(int n) and a concrete method to display the pattern title

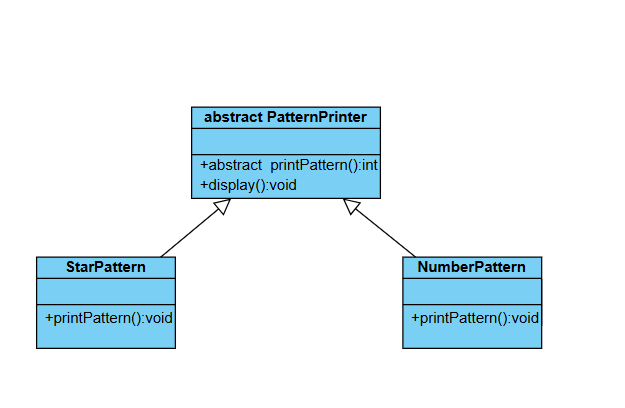
Implement two subclasses:

1)Star pattern -prints a right angled triangle of stars(\*)

2)Number pattern-prints a right angled triangle of increasing numbers

In the main() method ,create objects of both subclasses and print the patterns for a given number of rows

CLASS DIAGRAM:



CODE:

abstract class PatternPrinter{

public abstract void printPattern(int n);

public void display(String str){

System.out.println("/n " + str);

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

}

}

class StarPattern extends PatternPrinter{

public void printPattern(int n){

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

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

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

}

System.out.println();

}

}

}

class NumberPattern extends PatternPrinter{

public 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();

}

}

}

class Main{

public static void main(String[] args){

System.out.println("Name:BATCHU SYAM SUNDER");

System.out.println("Roll.no:24023");

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

StarPattern s =new StarPattern();

s.display("Star Pattern");

s.printPattern(5);

NumberPattern n = new NumberPattern();

n.display("number pattern");

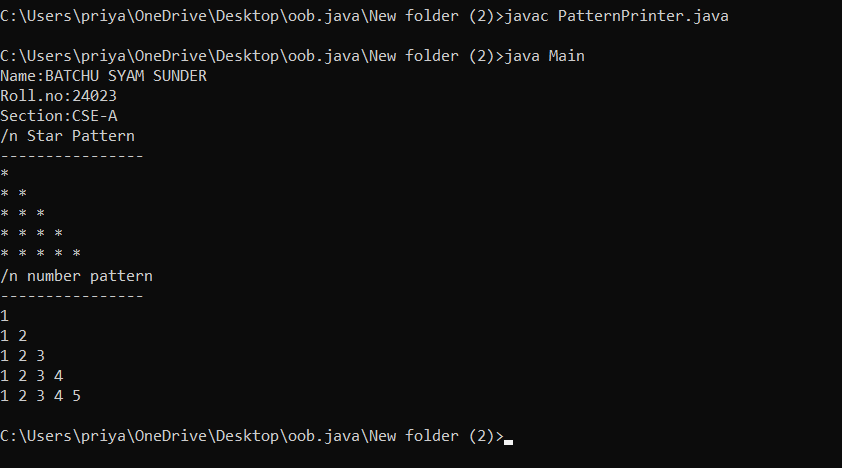
n.printPattern(5);

}

}

OUTPUT:

ERROR:



ERROR TABLE:

|  |  |  |
| --- | --- | --- |
| **S.No** | **ExpectedError** | **Reason** |
| **1** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |
| **2** | **}** | **Ending the class and main method is required** |

IMPORTANT POINTS:

1. Abstract Class (PatternPrinter)

* Defines:
  + Abstract Method → printPattern(int n) (must be implemented by subclasses).
  + Concrete Method → displayTitle(String title) (prints a formatted pattern title).
* Cannot be instantiated directly (must be extended).

2. Subclasses (StarPattern and NumberPattern)

* Extend PatternPrinter and override printPattern(int n).
* StarPattern:
  + Prints a right-angled triangle of stars (\*)

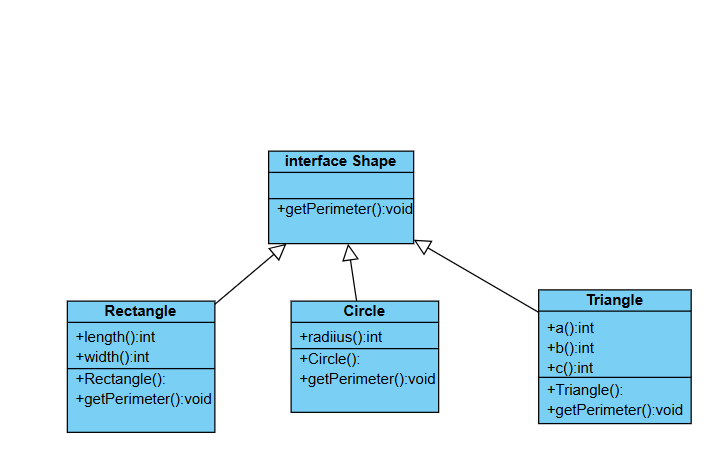
NumberPattern:

* Prints a right-angled triangle of increasing numbers.

WEEK-8

1.AIM: Write a java program to create an interface Shape with the getPerimeter() method . Create three classes Rectangle, Circle, Triangle that implement the Shape interface ,Implement the getPerimeter() method for each of the three classes.

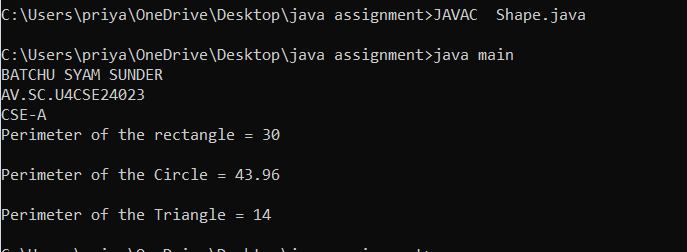
CLASS DIAGRAM:



CODE:

|  |
| --- |
| interface Shape {      void getPerimeter();  }  class Rectangle implements Shape{      int length ;      int width ;      Rectangle(int length , int width){      this.length = length ;      this.width = width ;      }      public void getPerimeter(){          System.out.println("Perimeter of the rectangle = "+ 2\*(length + width));      }  }  class Circle implements Shape{      int Radius ;      Circle(int Radius){      this.Radius = Radius ;      }      public void getPerimeter(){          System.out.println("Perimeter of the Circle = "+ (2\*3.14\*Radius));      }  }  class Triangle implements Shape{      int a ;      int b ;      int c ;      Triangle(int a, int b , int c){      this.a = a  ;      this.b = b  ;      this.c = c  ;      }      public void getPerimeter(){          System.out.println("Perimeter of the Triangle = "+ (a+b+c));      }  }  class main{      public static void main(String[] args){  System.out.printlln("BATCHU SYAM SUNDER\nAV.SC.U4CSE24023\nCSE-A");      Rectangle r = new Rectangle(5,10);      Circle c = new Circle (7);      Triangle t = new Triangle (5,6,3);      r.getPerimeter();      System.out.println();      c.getPerimeter();      System.out.println();      t.getPerimeter();      }  } |

Output:



**Errors:**

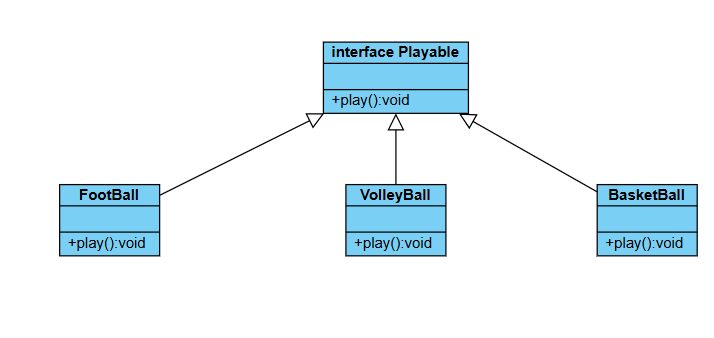
|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Fix** |
| Syntax Error | Typo in method name getPerimter() in Rectangle class | Rename method to getPerimeter() |
| Logical Error | Constructor parameter height doesn't match field name width | Change parameter name to width or assign this.width = height |

**Important Points and Explanation:**

* Interface is used to achieve abstraction and enforce a contract that any implementing class must follow.
* This shows runtime polymorphism where the same method (getPerimeter) has different behavior based on the shape.
* Method overriding requires the exact method name and parameters. Java uses method signatures to distinguish between methods.

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

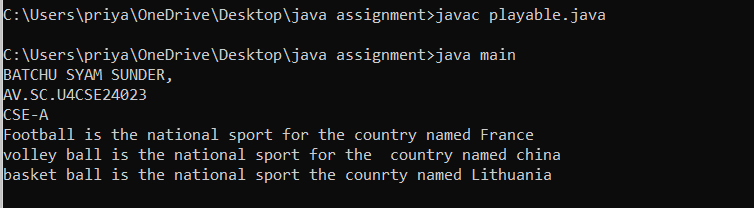
CLASS DIAGRAM:



CODE:

|  |
| --- |
| import  static java.lang.System.\* ;  interface playable {      void play();  }  class Football implements  playable {      public void play() {          out.println("Football is the national sport for the country named France ");      }  }  class Volleyball implements playable {       public void play() {          out.println("volley ball is the national sport for the  country named china");      }  }  class Basketball implements  playable{     public void play() {          out.println("basket ball is the national sport the counrty named Lithuania");      }  }  class main{      public static void main(String []args){          out.println("BATCHU SYAM SUNDER,"+'\n'+"AV.SC.U4CSE24023"+'\n'+"CSE-A");          Football f = new Football();          Volleyball v = new Volleyball();          Basketball b = new Basketball();          f.play();          v.play();          b.play();      }  } |

Output:



**Errors:**

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Fix** |
| Compile-time Error | Object instantiation uses new Football() instead of new football() | Change to new football() to match class definition |

**Important Points and Explanation:**

* The playable interface is defined with a method void play();. This sets a rule that any class implementing this interface must define its own version of the play() method.
* Football, volleyball, and basketball classes all implement the playable interface.
* Each class provides its specific implementation of the play() method, printing a message about the national sport of a country.