**23CSE111**

**OBJECT-ORIENTED PROGRAMMING**

**LAB REPORT**

****

**Department of Computer Science Engineering**

**Amrita School of Computing**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**VERIFIED BY : NAME :B.PRANATHI**

**ROLL NO : AV.SC.U4CSE24022**

**INDEX**

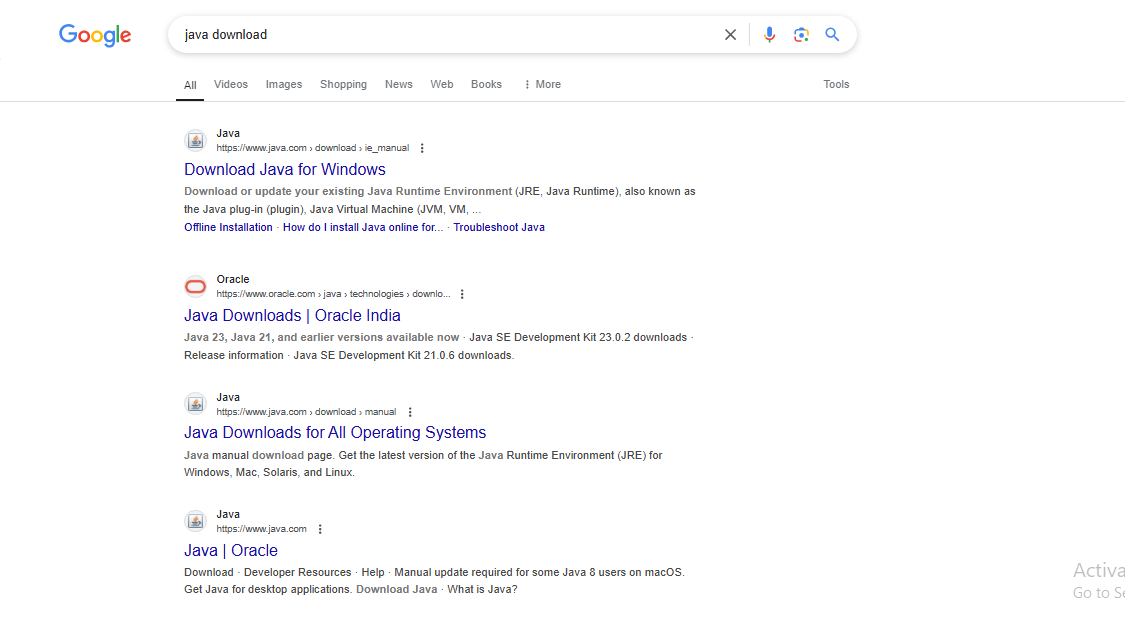
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Title** | **Date** | **Page No.** | **Signature** |
| Week 1 |  | 27-01-2025 |  |  |
| 1. | How to download and install Java Software. |  |  |  |
| **2.** | Write a Java Program to print the message “Welcome to Java Programming”. |  |  |  |
| **3.** | Write a Java program that prints: Name, Roll.no. , section of a student. |  |  |  |
| Week 2 |  | 10-02-2025 |  |  |
| **1.** | Write a java program to calculate the area of a rectangle. |  |  |  |
| **2.** | Write a java program to temperature from Celsius to Fahrenheit and vica-versa. |  |  |  |
| **3.** | Write a java program to calculate the simple interest. |  |  |  |
| **4.** | Write a java program to find the largest of three numbers, using ternary operator. |  |  |  |
| **5.** | Write a java program to find the factorial of a number. |  |  |  |
| Week 3 |  | 24-02-2025 |  |  |
| 1. |  |  |  |  |

**WEEK-1**

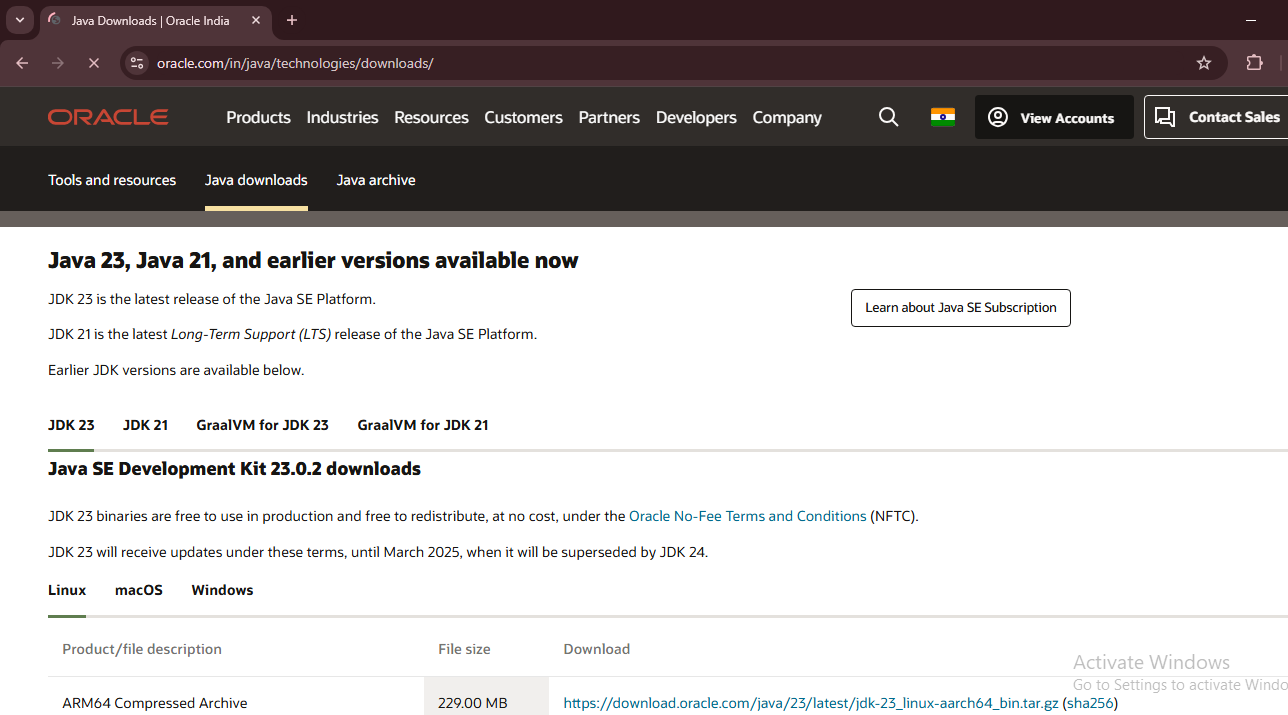
**PROGRAM-1:**

**Aim:** To download and install java software.

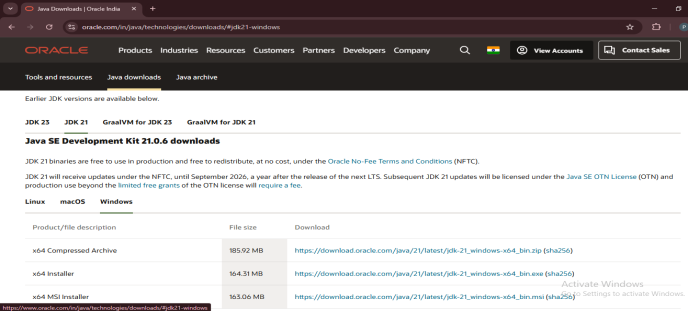
Step-1: search java download .



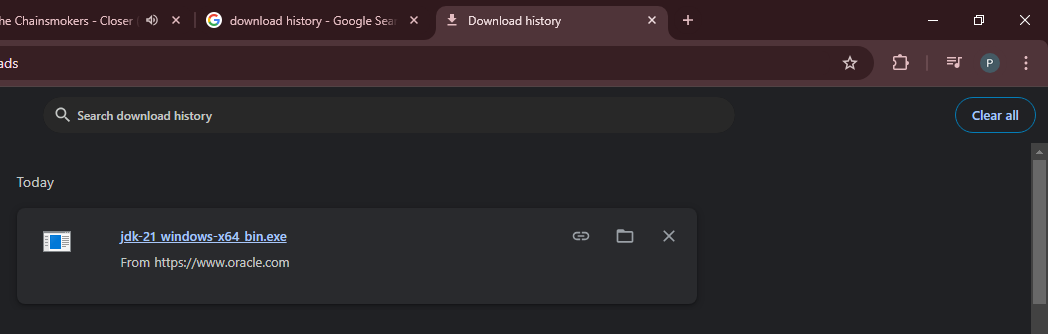
Step-2: click oracle java download (official website).



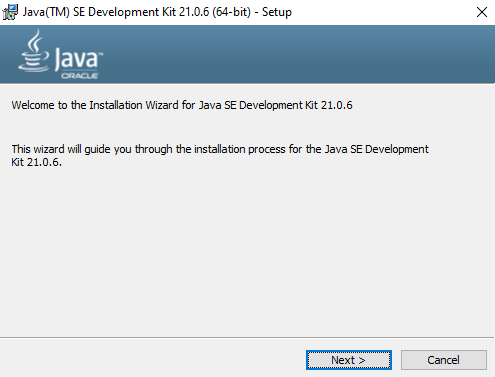
Step-3: click JDK 21, after that click windows.



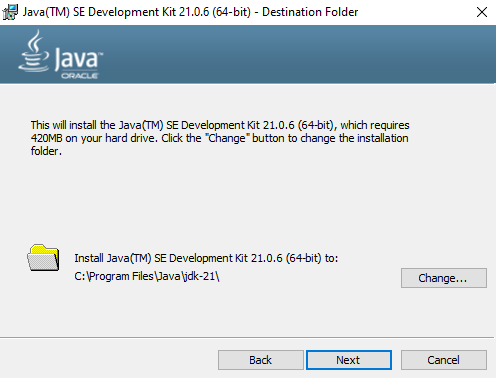
Step-4: click \*64 installer and then install.

****

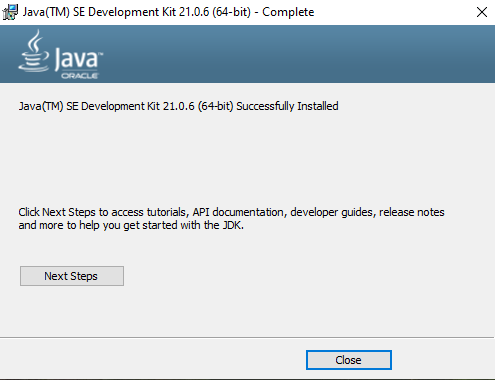
**Step-5:**

****

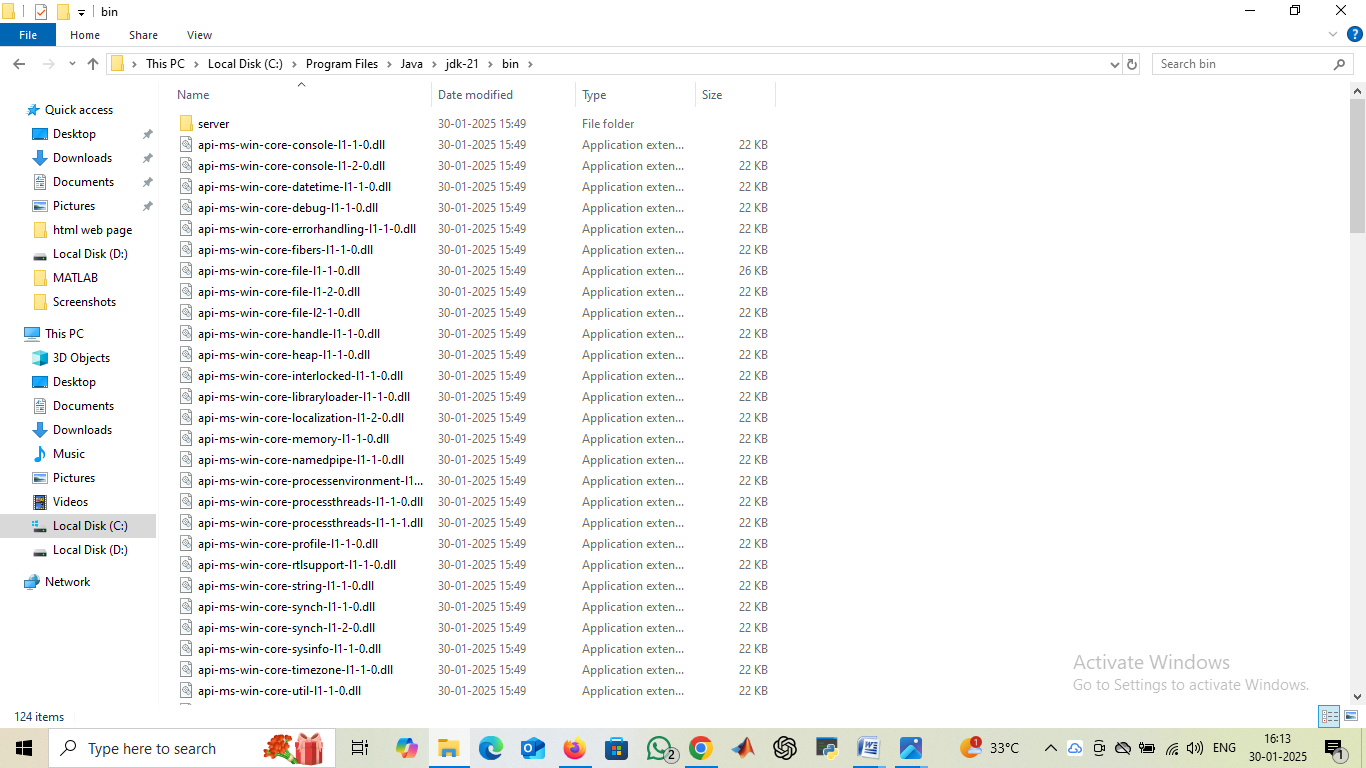
**Step-6:**

****

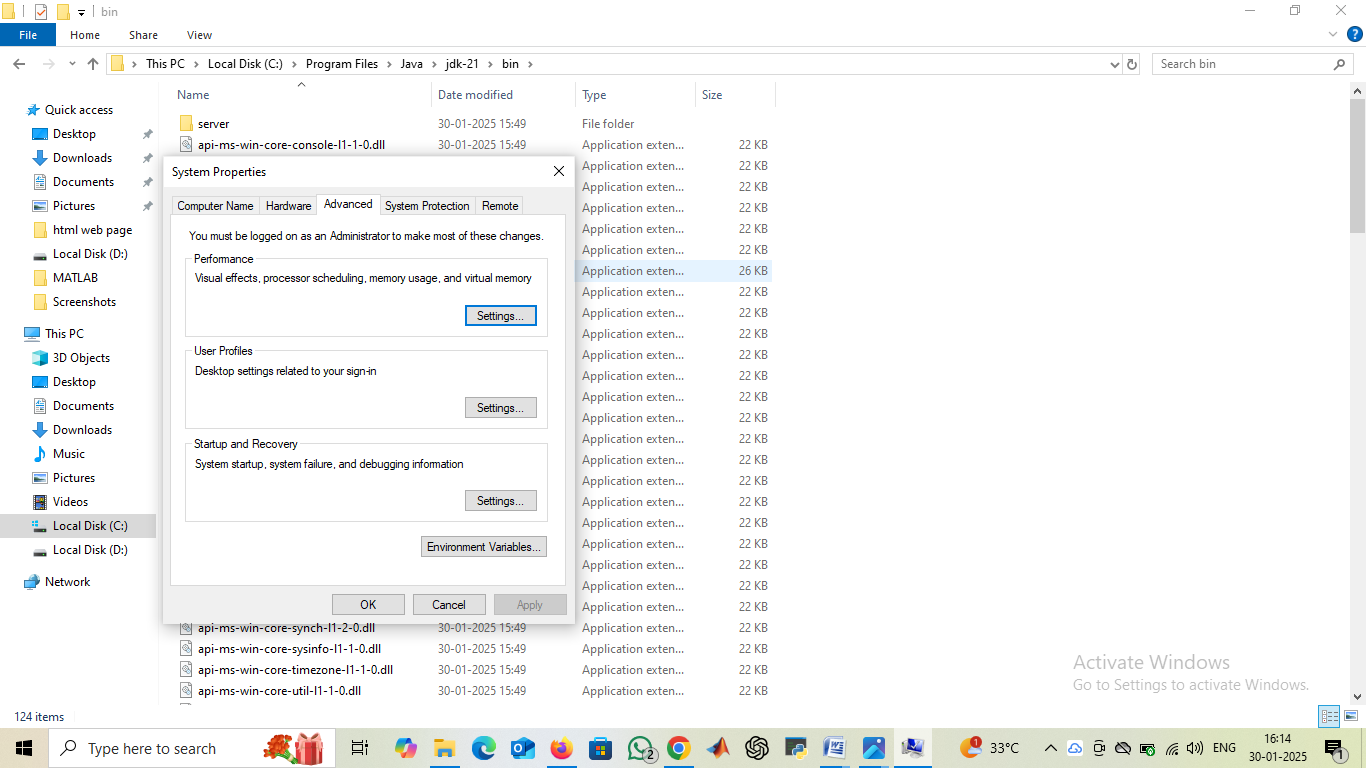
**Step-7:**

****

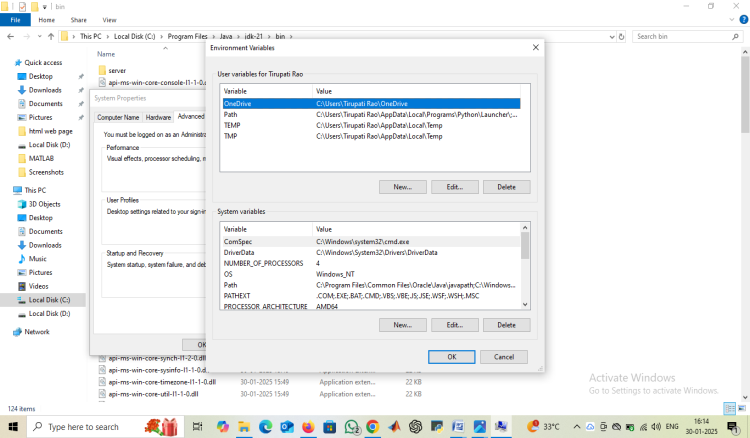
**Step-8:**

****

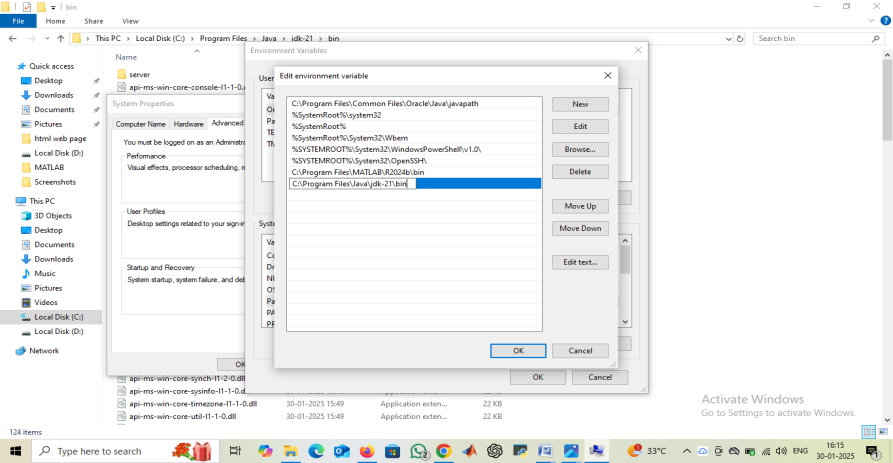
**Step-9:**

****

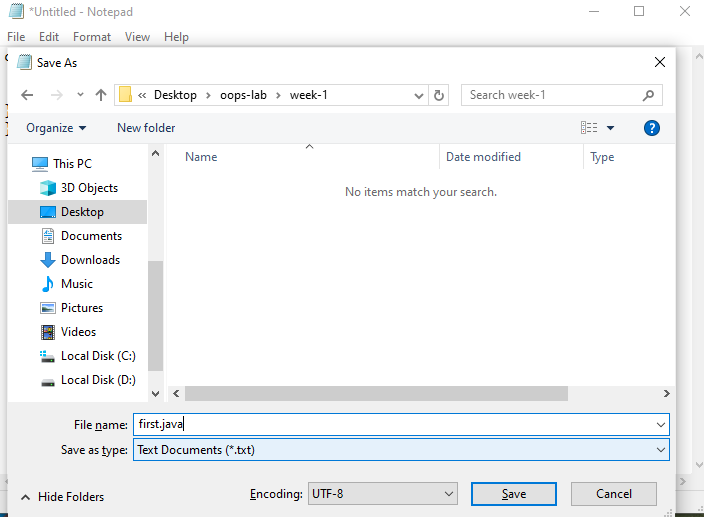
Step-10:



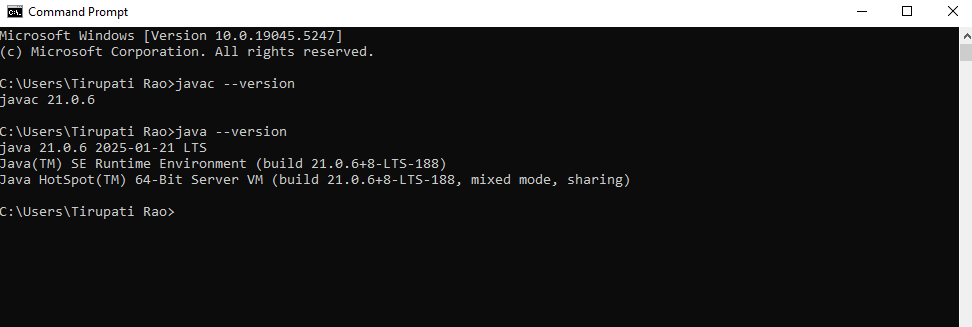
Step-11:



Step-12:



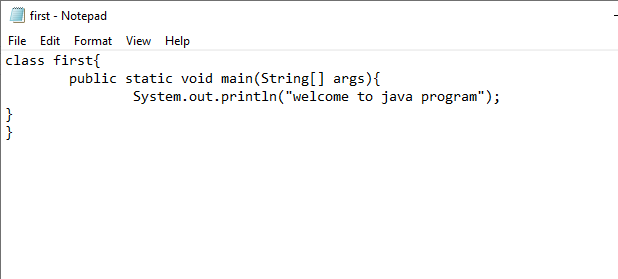
Step-13:



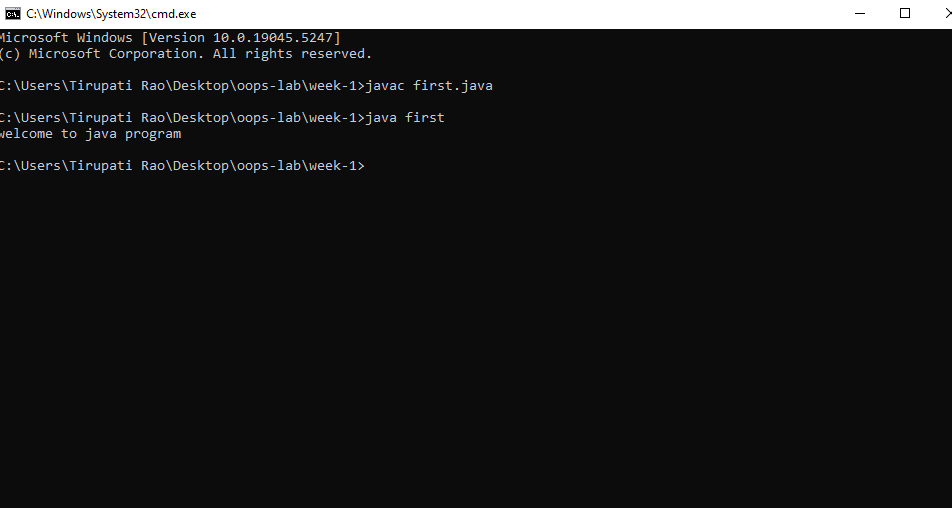
**PROGRAM-2:**

**Aim**: Write the program to print the message “Welcome to Java Programming”.

**Code**:



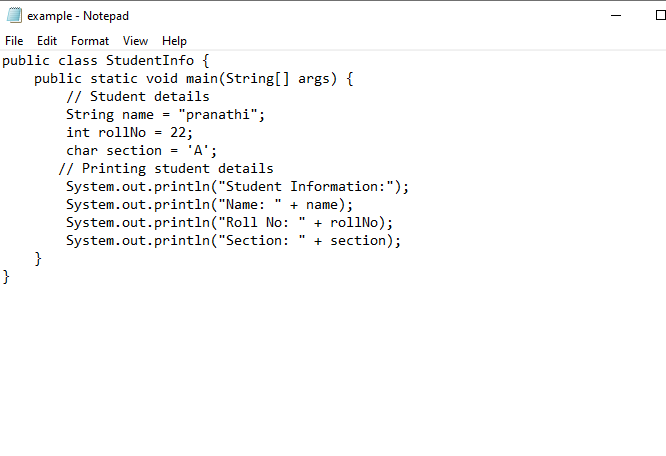
**Output:**



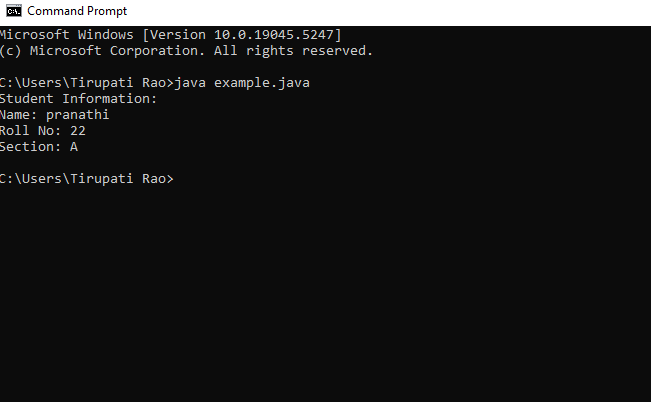
**Program-3:**

**Aim:** Write a java program that prints name, roll no and section of a student.

**CODE:**

****

**OUTPUT:**

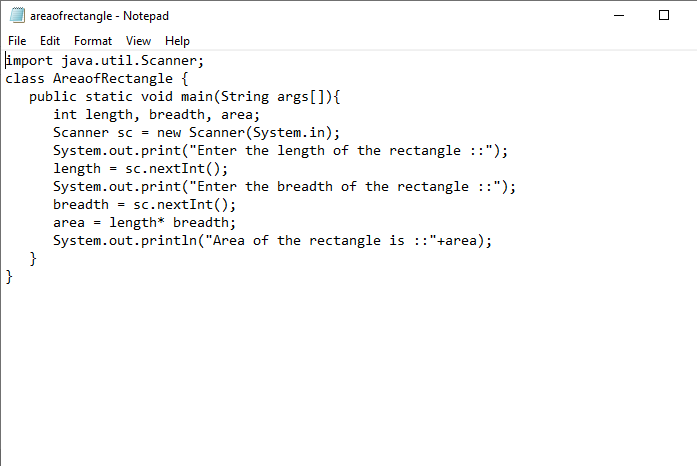


**WEEK-2**

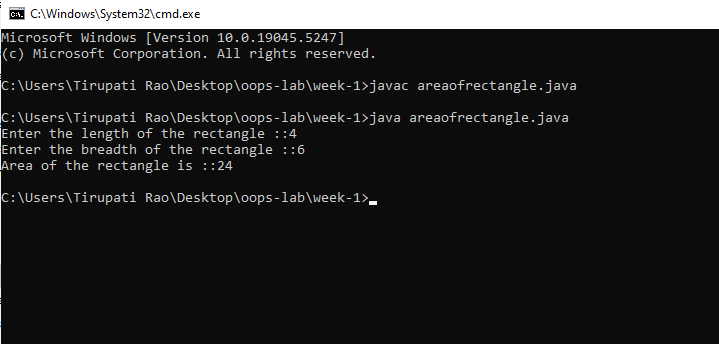
**PROGRAM-1:**

**AIM:** Write a Java program to calculate area of rectangle.

**CODE:**



**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MEASSAGE | ERROR RECTIFICATION |
| 1. | Error:”;”expected | Inserted “;”in line7 |
| 2. | Error:”?”unkownsysmbol | Replaced”?”with “:” |

**IMPORTANT POINTS:**

1. used Scanner library to get input from user in run time .

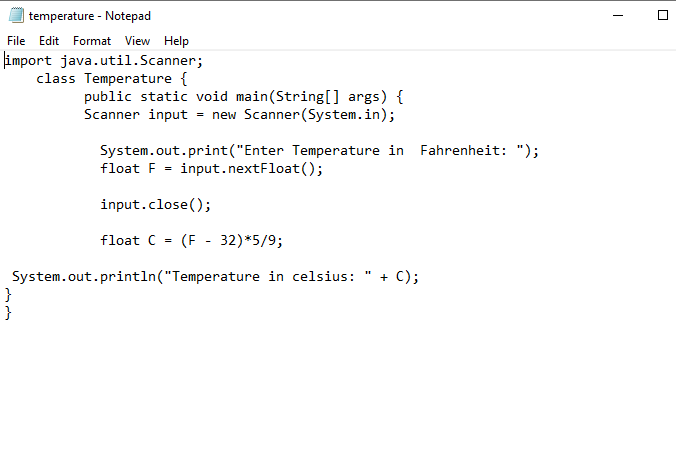
2. ”importjava.util.Scanner;”-step to import library.

3. “Scanner input=new Scanner(System.in);”-step to use the scanner .

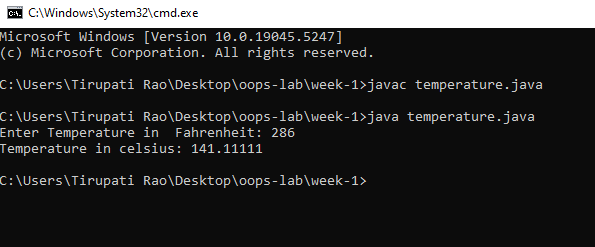
**PROGRAM-2**

**AIM:** Write a Java program to convert temperature from Fahrenheit to Celsius and vice versa.

**CODE: (A)**

****

**OUTPUT:**



**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
| 1. | Error :”oout” unknow keywoard | Replace”oout” with”out” |
| 2. | Error: “scanner”small letter case censitive | “Scanner” |

**IMPORTANT POINTS:**

1.used Scanner library to get input from user in run time.

2.”importjava.util.Scanner;”-step to import library.

3.”Scanner input= new Scanner(System.in);”-step to use the scanner.

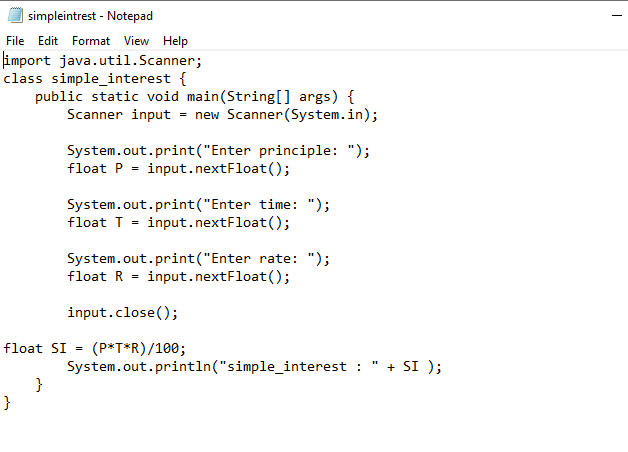
**(B).**

**CODE:**

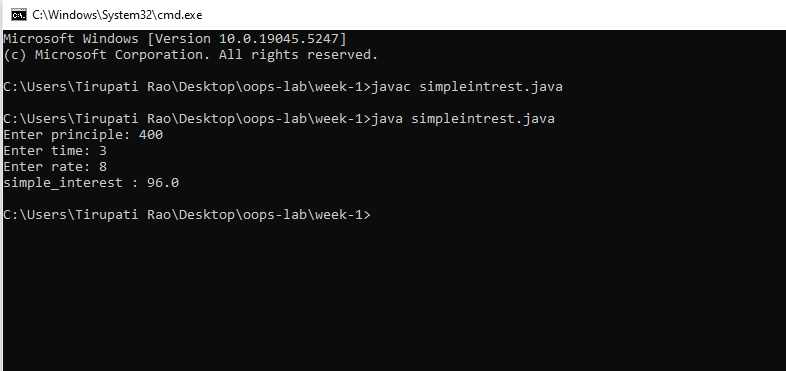
**PROGRAM-3:**

**AIM:** Write a Java program to calculate simple intrest.

**CODE:**



**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
| 1. | Error :”T” is not declred | Replace:”T” with”t” |
| 2. | Error : expected’;’ in line 8 | Insert ‘;’ in line 8 end |

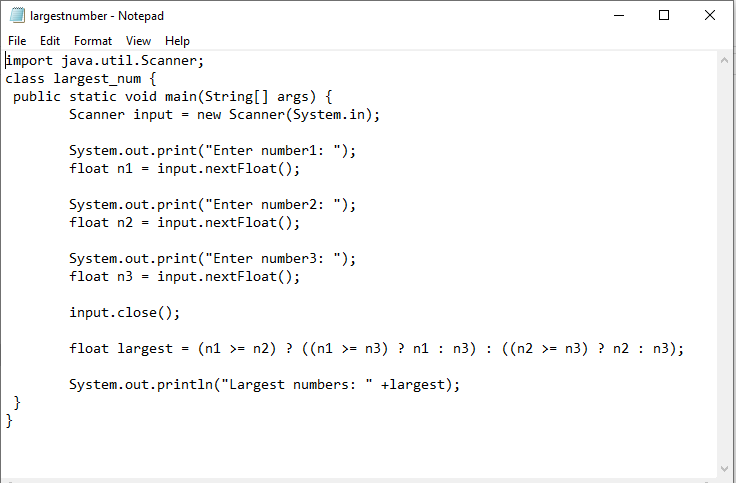
**IMPORTANT POINTS:**

1.java is a case sensitive language so “apple” is different from “APPLE”,so clear declaration of variables is important..

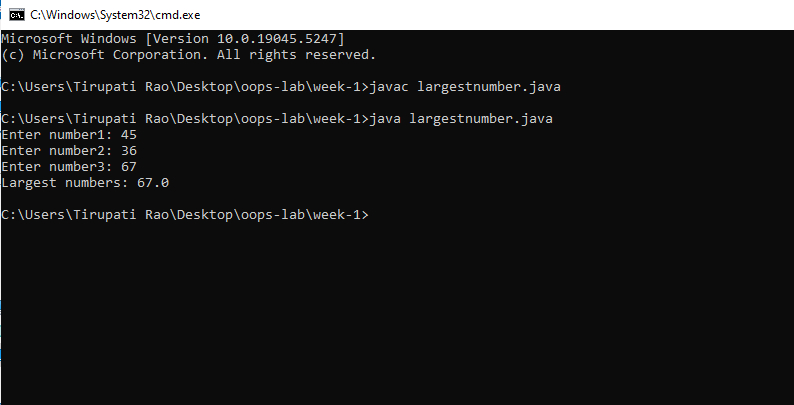
**PROGRAM-4:**

**AIM:** Write a Java program to calculate largest of 3 numbers using ternary operators.

**CODE:**

****

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error :”;” expected in line 6 | Insert”;” in end of line 6 |
|  | Error :”nextint();” non identified | Replace”next.Int();” |

**IMPORTANT POINTS:**

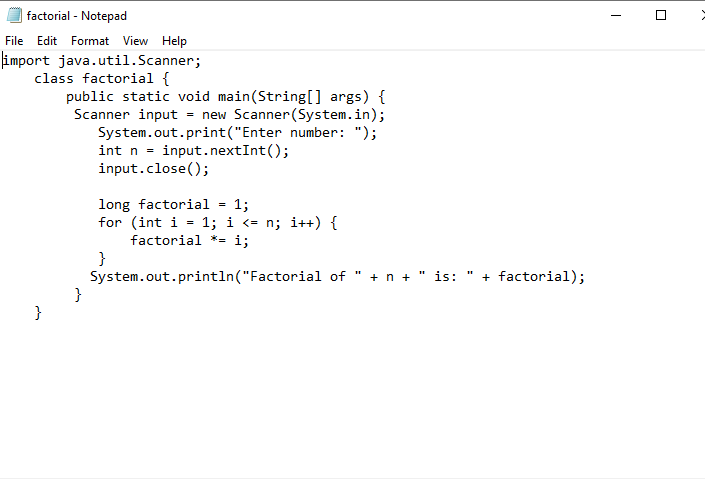
1.Ternary operators:a shorthand for the if-else statement, used to execute condition-based operations in a single line.

2.It evaluates a Boolean condition and returns trueValue if the condition is true, otherwise it returns falseValue.

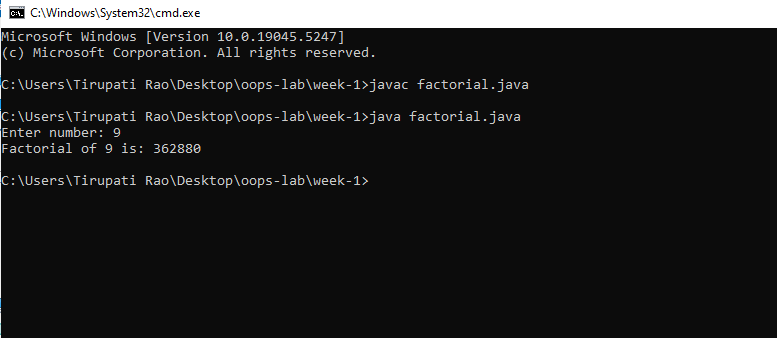
**PROGRAM-5:**

**AIM:** Write a Java program to calculate factorial of a number.

**CODE:**

****

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error: line-9 illegal start of expression | Rebuilt of the line -9 |
|  | Error :iteration error | Correct iteration inserted |

**IMPORTANT POINTS:**

1.Java for loop is a control flow statement that allows code to be executed repeatedly based on a given condition.

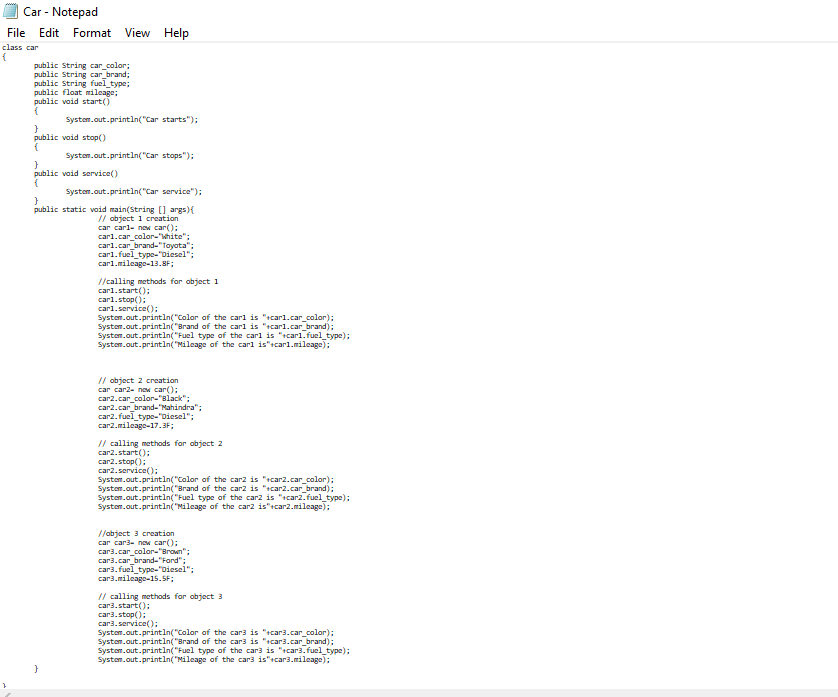
2.The for loop in java provides an efficient way to iterate over a range of values ,execute code multiple times,or traverse arrays and collections.

**WEEK-3**

**PROGRAM-1:**

**AIM:**

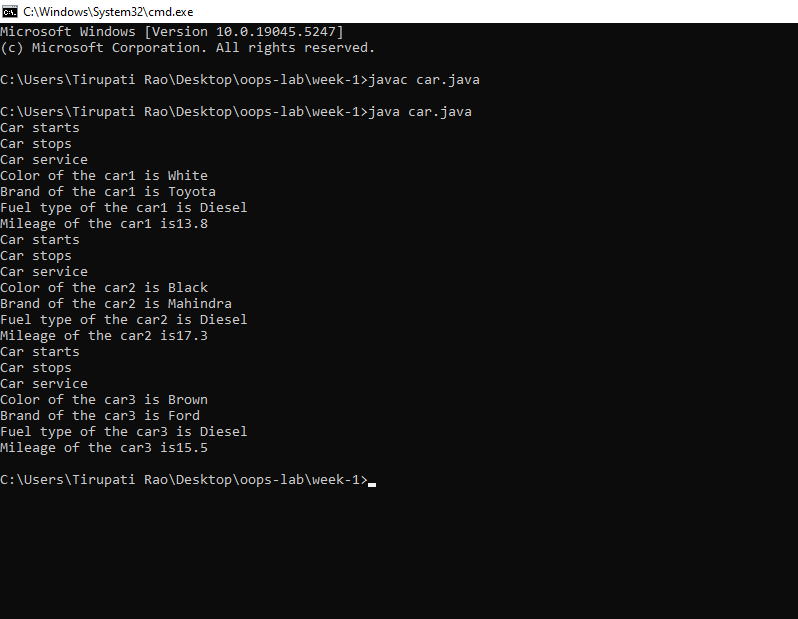
**CODE:**

****

**CLASS DIAGRAM:**

|  |
| --- |
| Car() |
| +car\_name:string  +car\_color:string  +car\_brand:string  +fule\_type: int  +maleage:int |
| +start:void()  +stop:void()  +static:void() |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error: line7 expected ‘; | Inserted ‘;’ |
|  | Error :line 7 unknow’\_\_’ | Removed ‘\_’ |
|  | Error : correct data type declararion in constructor | Rectified by declaring the data type as String and int. |

**IMPORTANT POINTS:**

1.Java constructor is used to save the variables present in different or same class or methods.

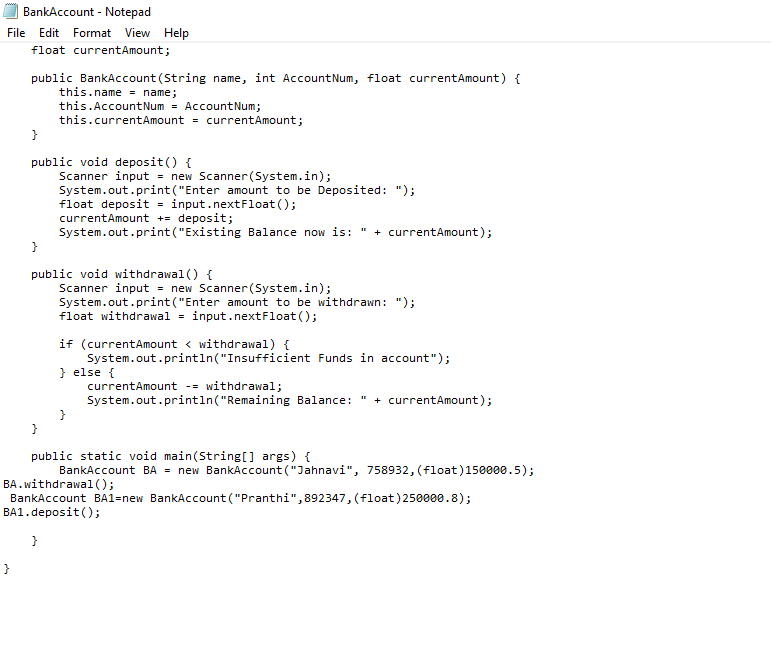
2. In Java, the this keyword refers to the current instance of a class. It is commonly used to distinguish between instance variables and parameters with the same name, or to refer to the current object from within a method or constructor.

3. In Java, a method is a block of code that performs a specific task and can be invoked to execute that task. It typically consists of a method signature (name, return type, and parameters) and the body of the method, which contains the logic.

**PROGRAM-2:**

**AIM:**

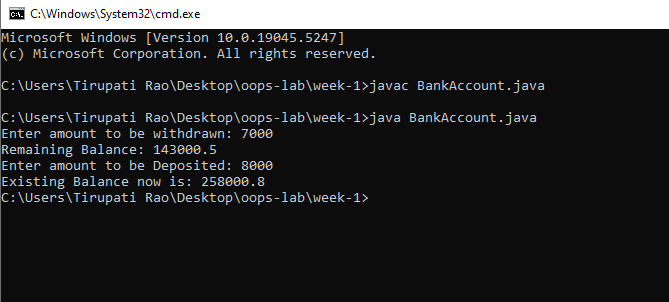
**CODE:**

****

**CLASS DIAGRAM:**

|  |
| --- |
| BankAccount |
| -existing:float  +name:String |
| +BankAccount()  +deposit:void()  +withdraw:void() |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error: nextString(); wrong identifier | Rectification: next(); |
|  | Error :line 7 unknow’\_\_’ | Removed ‘\_’ |
|  | Error : if statement ‘{}’ expected | Inserted ‘{}’ |

**IMPORTANT POINTS:**

1.Java constructor is used to save the variables present in different or same class or methods.

2. In Java, the this keyword refers to the current instance of a class. It is commonly used to distinguish between instance variables and parameters with the same name, or to refer to the current object from within a method or constructor.

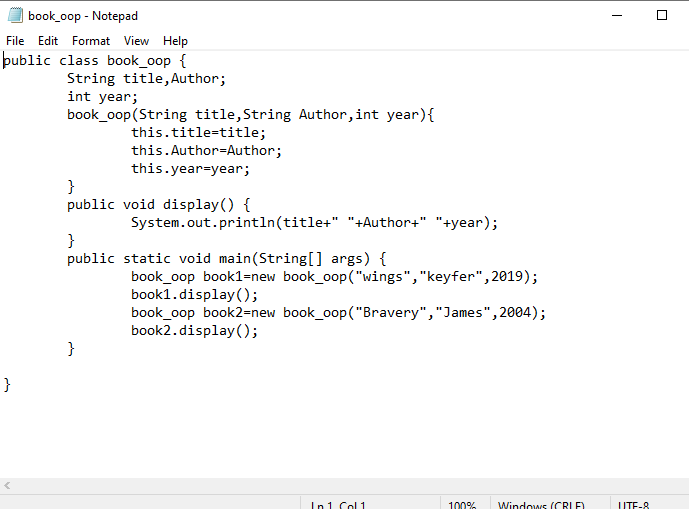
3. In Java, a method is a block of code that performs a specific task and can be invoked to execute that task. It typically consists of a method signature (name, return type, and parameters) and the body of the method, which contains the logic.

**WEEK-4**

**PROGRAM-1:**

**AIM:**

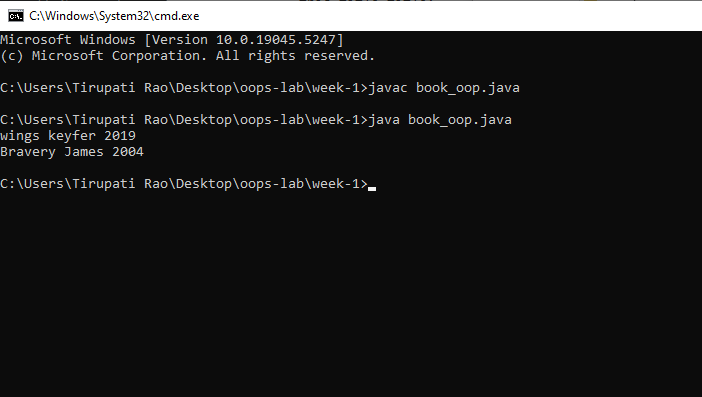
**CODE:**

****

**CLASS DIAGRAM:**

|  |
| --- |
| Book |
| +title\_of\_book:string  +author:string  +year\_publication:int |
| +book()  +detailes:void() |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
| 1. | Error: “ this.year\_public;=year\_public;” | Rectification: removed the ‘;’ |
| 2. | Error :”missing ‘;’-“System.out.println(“..”); | Inserted the ‘;’ in the line. |

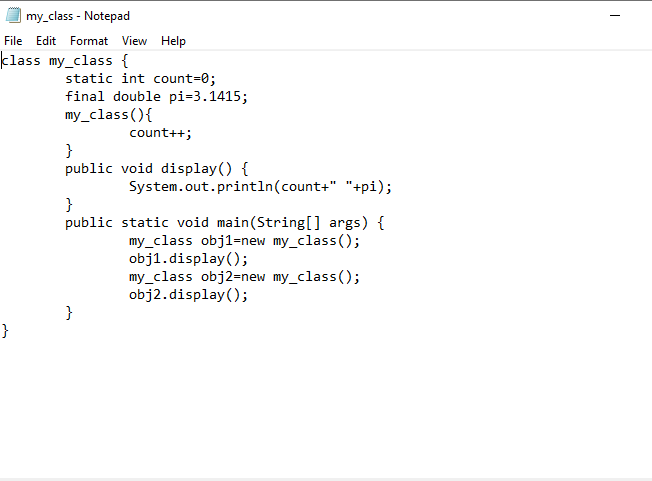
**IMPORTANT POINTS:**

1. Java constructor is used to save the variables present in different or same class or methods.
2. In Java, the this keyword refers to the current instance of a class. It is commonly used to distinguish between instance variables and parameters with the same name, or to refer to the current object from within a method or constructor.

**PROGRAM-2:**

**AIM:**

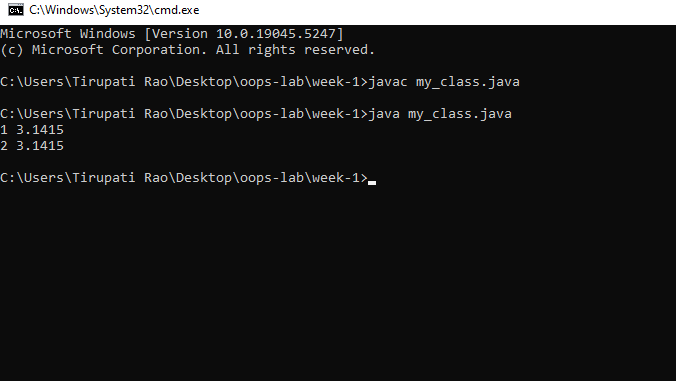
**CODE:**

****

**CLASS DIAGRAM:**

|  |
| --- |
| Myclass |
| -count:0  -pi:3.1415 |
| +myclass()  +values:void() |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error: argument required of type int | Rectification: rectified the argument issue. |
|  | Error :line 7 unknow’\_\_’ | Removed ‘\_’ |
|  | Error : if statement ‘;’ expected | Inserted:count++; |

**IMPORTANT POINTS:**

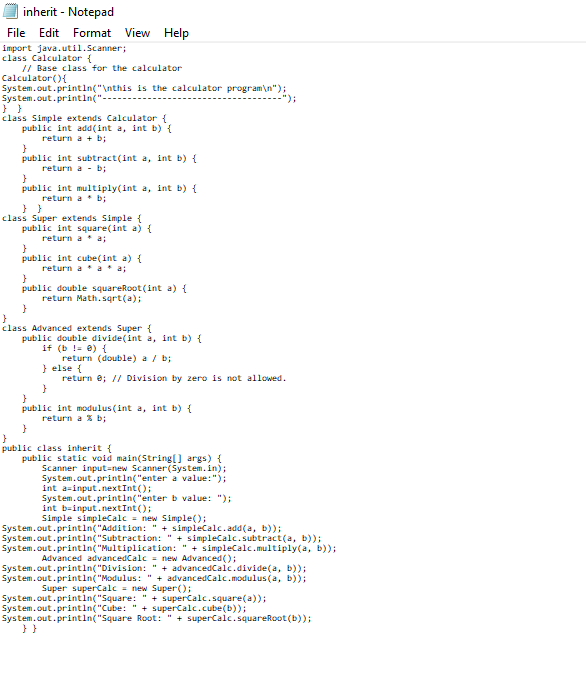
1. Java constructor is used to save the variables present in different or same class or methods.
2. In Java, the ++ operator increments a variable by 1, either as **pre-increment** (++x) or **post-increment** (x++).
3. In Java:
4. **static**: A static variable belongs to the class, not instances, meaning all objects share the same value.
5. **final**: A final variable cannot be modified once assigned, making it constant.

**WEEK-5**

**PROGRAM-1**

**AIM:** create a calculator using the operations including add, sub, multi and div using multilevel inheritance and display the desired output

**CODE:**



CLASS DIAGRAM:

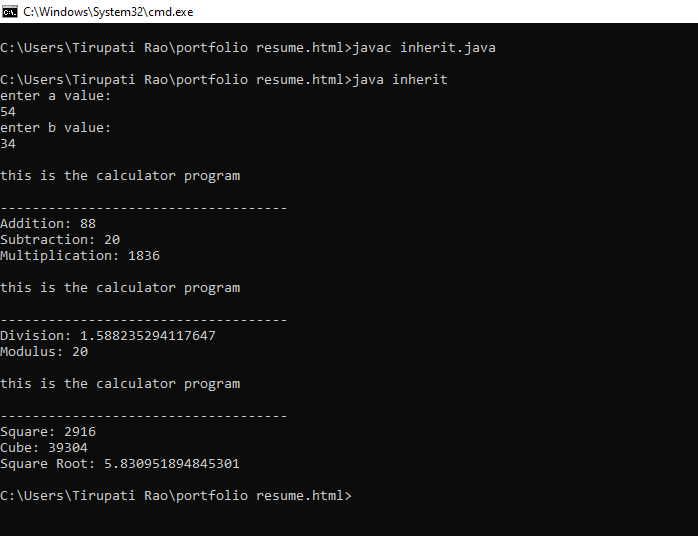
|  |
| --- |
| Calculator |
| +Calculator() |

|  |
| --- |
| Siimple |
| +add:int  +substract:int  +multiply:int |

|  |
| --- |
| Super |
| +square:int  +cube:int  +squareRoot:double |

|  |
| --- |
| Advanced |
| +divide:double  +module:int |

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
|  | Error: mutipile inheritance in the Advanved class | Implemented Advanced class from Super class. |
|  | Error :Scanner; | Scanner(System.in); |

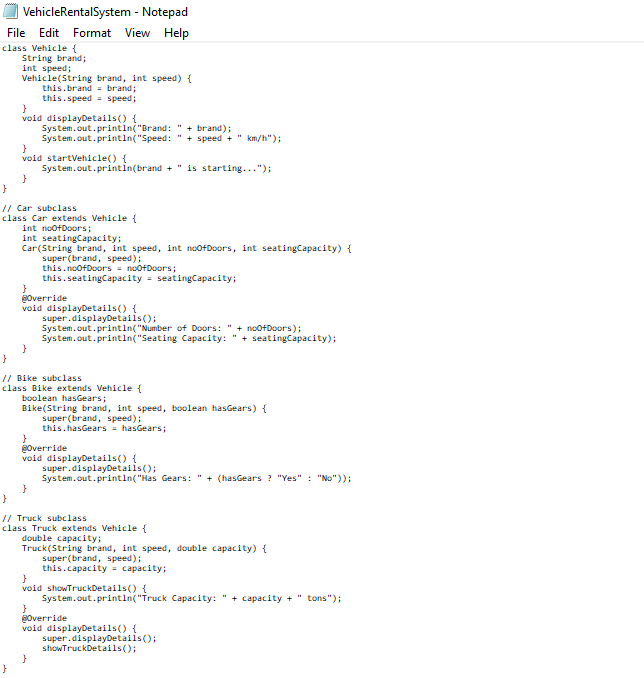
**IMPORTANT POINTS:**

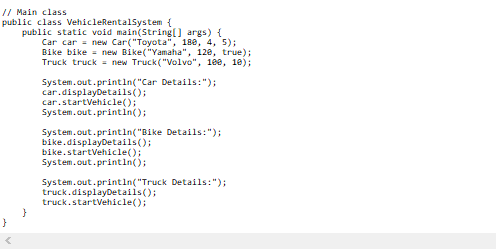
1. Multiple inheritance lets a class inherit from multiple parents, combining their features, but can cause issues like the diamond problem, resolved by MRO.
2. Math.sqrt() in Java calculates the square root of a non-negative double value and returns a double result, or NaN if the input is negative.
3. The import java.util.Scanner; statement in Java allows you to use the Scanner class from the java.util package, which is commonly used to read user input from the console.

**PROGRAM-2**

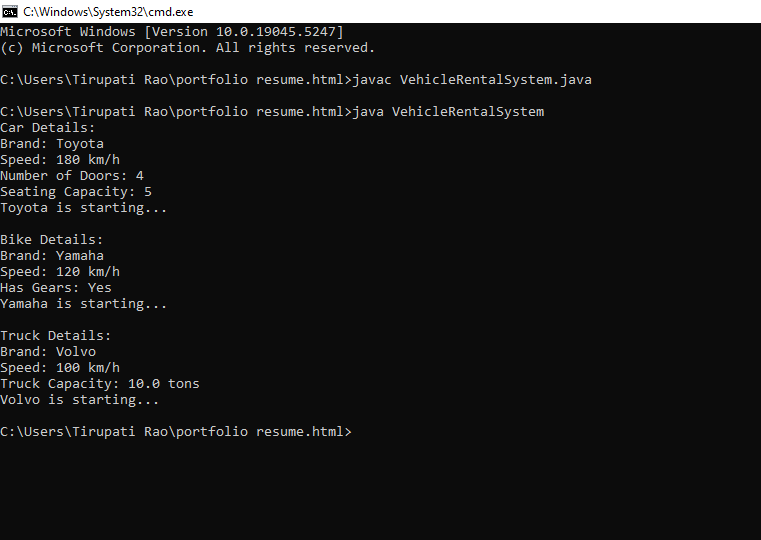
**AIM:** Create a java program of a vehicle entry company hireachical wants to develop his system that maintains information about different types of cars and bikes and they need a program to store details about each vehicle such as brand and speed

**CODE:**

****

****

**OUTPUT:**

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| S.NO | ERROR MESSAGE | ERROR RECTIFICATION |
| 1. | Error: Incorrect Constructor Arguments. | the arguments passed when creating an object match the constructor's parameter list in both **number** and **type**. |
| 2. | Error : Scanner; | Scanner(System.in); |

**IMPORTANT POINTS:**

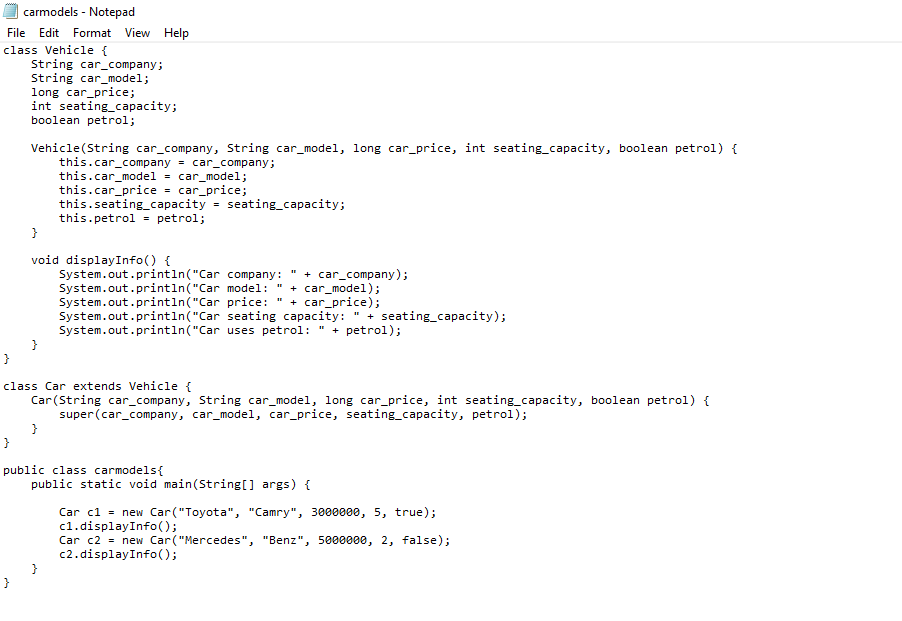
1. Hierarchical inheritance is a type of inheritance where multiple subclasses inherit from a single parent class, allowing code reuse and reducing redundancy.
2. A **constructor** is a special method in a class used to initialize new objects with default or provided values. It is automatically called when an object is created and sets up the object's initial state.

**WEEK-6**

**PROGRAM-1:**

**AIM:** Write a java program to create a vehicle class with a method display info(). Override this method in the car subclass to provide specific information about car (car company, seating capacity, petrol or not).

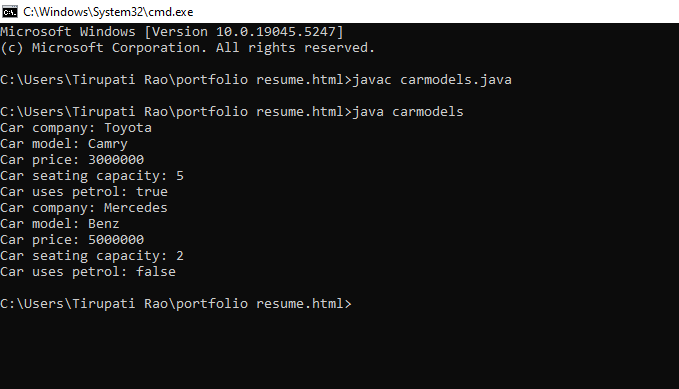
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| Vehicle |
| -Brand: String  -Speed: int |
| + vehicle(brand: string  Speed: int)  +start vehicle(): void  +displaydetails():void |

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| Code error | Code rectification |
| 1.Incorrect class name for main method(Truck).    2. Inconsistent car model output in displayinfo(). | 1.Rename Truck to Main or place main inside car or vehicle.  2. Ensure Car correctly passes Toyota” to super(car\_model,color,fueltype) |

**IMPORTANT POINTS:**

1.Inheritance**:** The Car class extends the Vehicle class, demonstrating inheritance in Java.

2.Constructor Chaining:The Car class calls the parent constructor using super(car\_model, color, fuel\_type); to initialize inherited attributes.

3.Method Overriding:The Car class overrides the displayInfo() method from Vehicle and calls super.displayInfo() to reuse the parent method before adding its own output.

4.Incorrectmain Class Name:The main method is inside Truck, which is unrelated to Vehicle and Car. The class should be renamed for clarity.

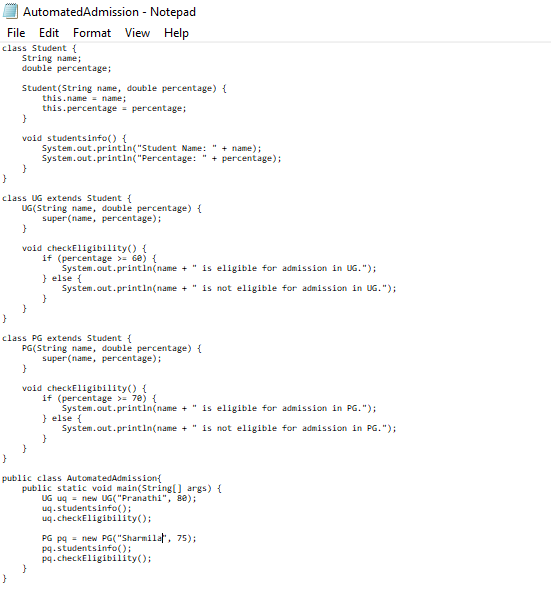
**PROGRAM-2:**

**AIM:** A college is developing an automated admission system that verifies students eligibility(UG) and postgraduation(PG) programs. Each program has different eligibility criteria based on the students percentage in their previous qualification.

1. UG admission recquire a minimum of 60%.

2. PG admission recquire a minimum of 70%.

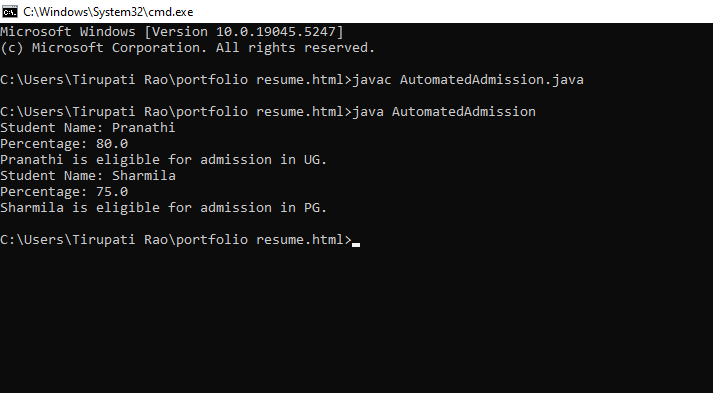
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| AutomatedAdmission |
| -Scanner: scanner  -Name: String  -Percentage : double  -Program: stirng |
| + main(args:String[]): void  +takeInput(): void  +checkEligibility(): void  +closeScanner(); void |

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| Code error | Code rectification |
| **1.Scanner nextLine() issue after nextDouble():** After scanner.nextDouble(), the newline character remains in the buffer, causing nextLine() to be skipped.  **2.Program type input case sensitivity issue**: If the user enters ug or pg in lowercase, it may cause incorrect comparisons. | **1**.Add scanner.nextLine(); after nextDouble(); to consume the leftover newline.  **2.**Use program.toUpperCase() to ensure case-insensitive comparison. |

**IMPORTANT POINTS:**

1.User Input Handling:Uses Scanner to take user input for name, percentage, and program type.

2.Decision Making with Conditions**:** Uses if-else statements to check eligibility criteria.

3.String Handling:Converts program input to uppercase (toUpperCase()) to handle case variations.

4.Closing Scanner:Properly closes scanner using scanner.close(); to prevent resource leaks.

**PROGRAM-3:**

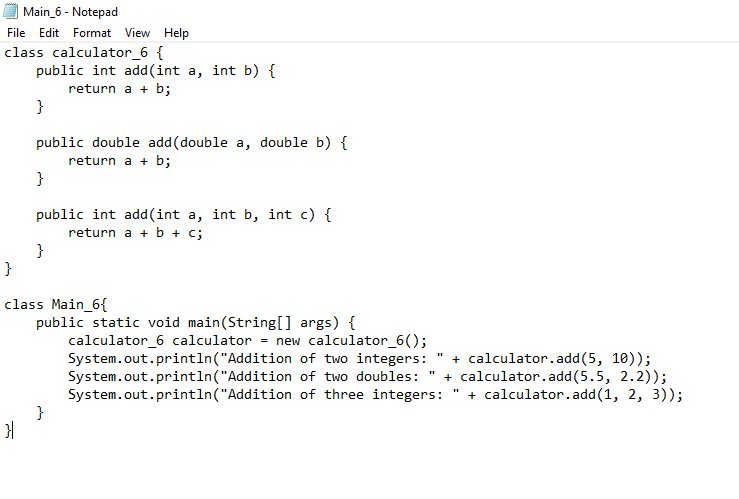
**AIM: Create a calculator class with overloaded methods to perform** addition of:

1. Add two integers

2. Add two doubles

3. Add three integers

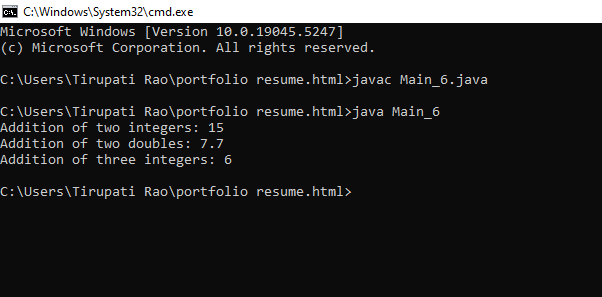
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| Calculator |
| + add(int, int): int  +add(double, double): double  +add(int,int,int): int |

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| Code error | Code rectification |
| 1.Method parameters missing spaces. E.g.,”inta, intb”should be “int a, int b”  2.Inconsistent indentation in method bodies | 1**.** Add proper spacing between parameters: (int a, int b)  2. Fix indentation:  Consistent 4 space o indentation. |

**IMPORTANT POINTS:**

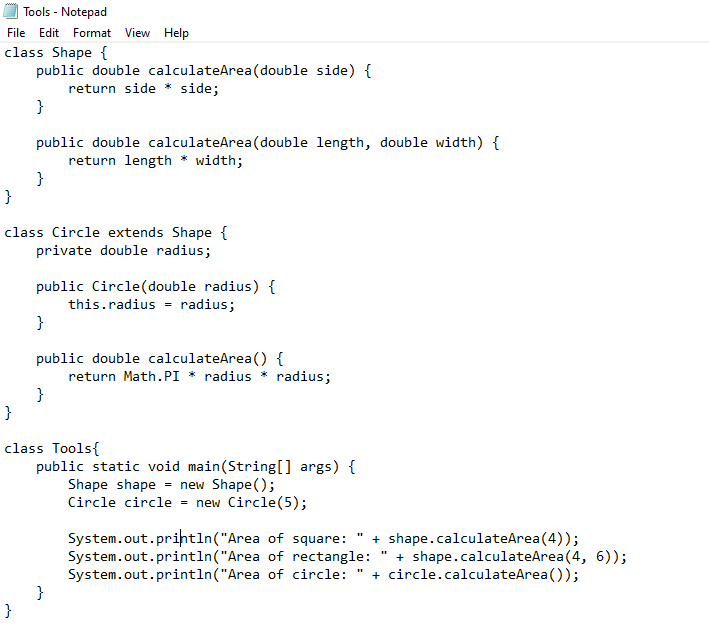
1.Method Overloading**:** The add method is overloaded with different parameter types and counts, demonstrating compile-time polymorphism.

2.Automatic Method Selection:Java selects the appropriate add method based on the argument types during compilation.

**PROGRAM-4:**

**AIM:** Create a shape class with a method to calculate area i.e., overloaded for different shapes eg: Squares, Recatangle. Then create a subclass circle that overrides the calculateArea() method for a circle.

**CODE:**

****

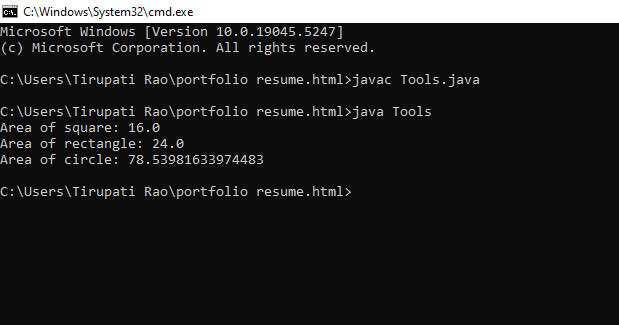
**CLASS DIAGRAM:**

|  |
| --- |
| SHAPE |
| + CalculateArea(side:double): double +CalculateArea(width: double, length: double): double |

|  |
| --- |
| CIRCLE |
| + CalculateArea(radius: double): double |

|  |
| --- |
| Tools |
| +main(args:String[]): Void |

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| Code error | Code rectification |
| 1. Method calls in main are missing an object reference (e.g., calculateArea(4) instead of s.calculateArea(4)).  2. Circle class method does not override theparent class method properly. | 1. Use s.calculateArea(4) and c.calculateArea(2) to call the method correctly.  2. Ensure @Override is used, and the method signature should match correctly. |

**IMPORTANT POINTS:**

1.Inheritance: Circle class extends Shape, inheriting its methods.

2.Method Overloading: Shape has multiple calculateArea methods with different parameters.

3.Method Overriding: Circle overrides calculateArea from Shape to implement its own formula.

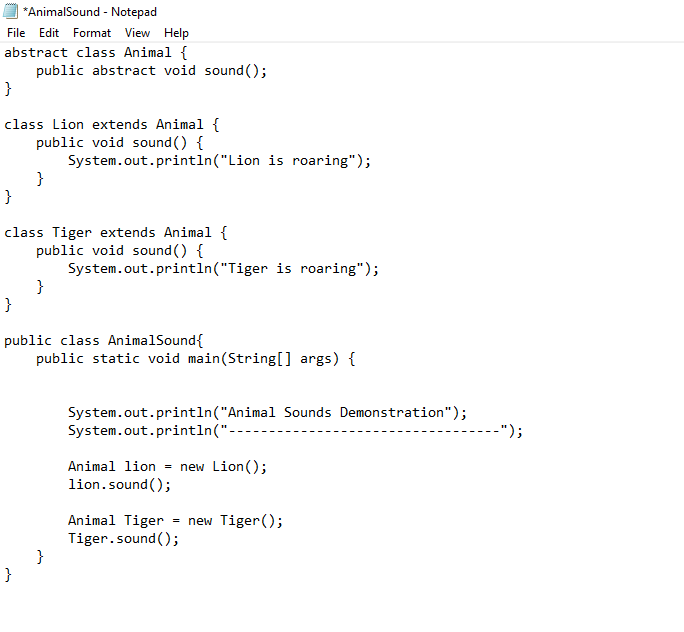
4.Polymorphism: The overridden method in Circle demonstrates runtime polymorphism.

**WEEK-7**

**PROGRAM-1:**

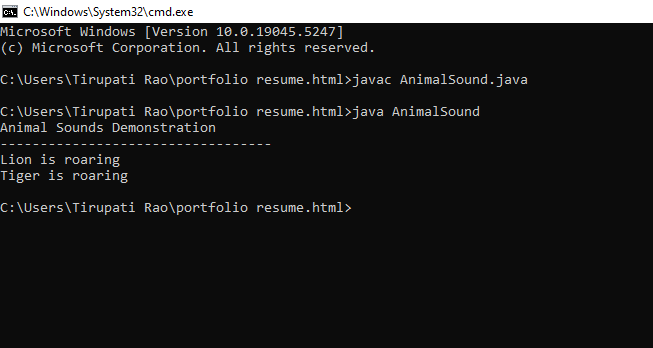
**AIM**: create Java program to create an abstradcass animal with an abstract method called sound ().Create a subclass Lion and tiger that extend the Animal class and implement the sound () method to make a specific sound for each animal.

**CODE:**



**CLASS DIAGRAM:**

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| **Code error** | **Code rectification** |
| 1. Forgetting to use abstract keyword for the sound() method.  2 Not overriding the sound() method in subclasses. | 1.Rectified as abstract void sound();  2. Added void sound() { ... } in each subclass. |

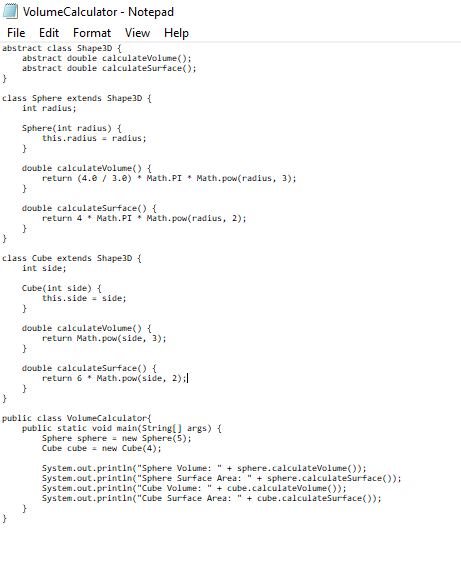
**IMPORTANT POINTS:**

* 1. abstract class Animal: Can't be directly used to create objects.
  2. abstract void sound(): Forces subclasses to implement this method.
  3. Lion and Tiger both override sound().
  4. Animal a = new Lion(); uses runtime polymorphism.

**PROGRAM-2:**

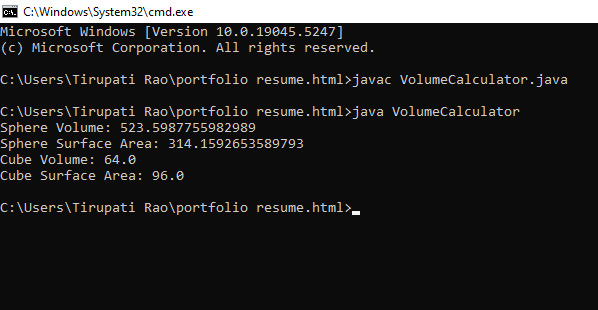
**AIM**: Write a Java program to create an abstract class shape 3D with abstract methods calculate volume ()and calculate surface Area ()create subclasses Sphere and cube that extend the Spape 3D clas and implement the respective methods to calculate ine volume and surface area of each shape.

**CODE:**



**CLASS DIAGRAM:**

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| **Code error** | **Code rectification** |
| |  | | --- | |  |  |  |  | | --- | --- | | 1. int used instead of double for Volume surface | | |  |  |  |  |  | | --- | --- | | 2.(4 / 3) used instead of (4.0 / 3.0) | | |  |  |  |  | | --- | | 3.14 used as approximation for π |  |  | | --- | |  |  |  | | --- | |  | | 1. Changed return types of calculateVolume() and calculateSurface() to double 2. Used floating-point division to avoid integer division loss. 3. 3.Used Math.PI for more accurate calculations. |

**IMPORTANT POINTS:**

**1.Abstract Class Used**:Shape3D is an abstract class with abstract methods – it can't be directly used to create objects.

**2.Method Overriding**:Sphere and Cube both override calculateVolume() and calculateSurface() with their own formulas.

**3.Return Type: double** :Volume and surface area can be decimal, so methods return double, not int.

**4.Use of Math.PI and Math.pow()**: More accurate than hardcoding 3.14 and r\*r\*r. It's a good practice for real calculations.

**PROGRAM-3:**

**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)
* A concrete method to display the pattern title

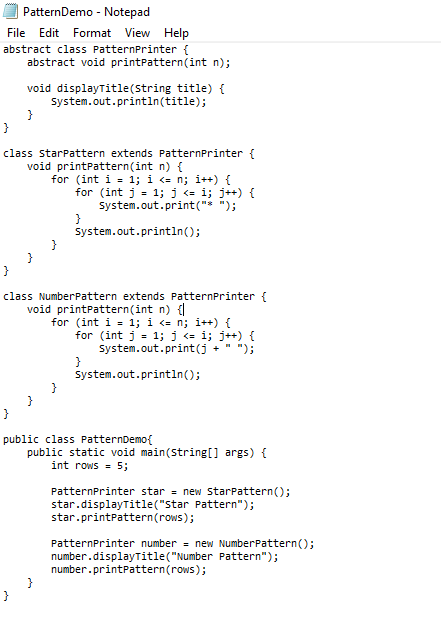
Create two subclasses:

1.StarPattern: Prints a right-angled triangle of stars (\*)

2.NumberPattern: Prints a right-angled triangle of increasing numbers

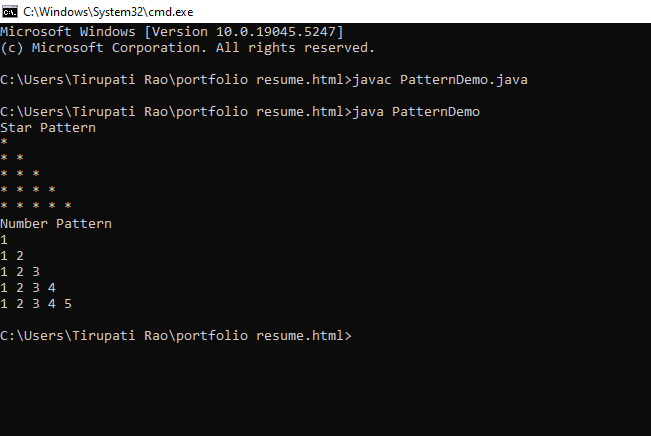
In the main() method, create objects of both subclasses and print the patterns for a given number of rows.

**CODE:**



**CLASS DIAGRAM:**

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| **Code error** | 1. **Code rectification** |
| |  | | --- | |  |  |  |  | | --- | --- | | 1.Wrong loop logic ( printing \* without loop). | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | 2.displayTitle method not used before pattern printing |  |  | | --- | | 3.Forgot to implement printPattern(int n) in subclass | | | | |  |  |  |  | | --- | |  |  |  | | --- | |  | | 1.Use nested loops: outer loop for rows, inner loop for printing symbols or numbers.  2.Call displayTitle() before printing the pattern for proper formatting  3.Implemented the method in both subclasses |

**IMPORTANT POINTS:**

1. Abstract class PatternPrinter cannot be instantiated directly.
2. Abstract method printPattern(int n) must be implemented in all subclasses.
3. Concrete method displayTitle(String title) is reusable by both subclasses.
4. Use of inheritance: StarPattern and NumberPattern extend the abstract class.