**Object Oriented Programming**

**23CSE111**



**Department of Computer Science and Engineering**

**Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Verified by: Name:**

**Roll number :AV.SC.U4CSE24040**

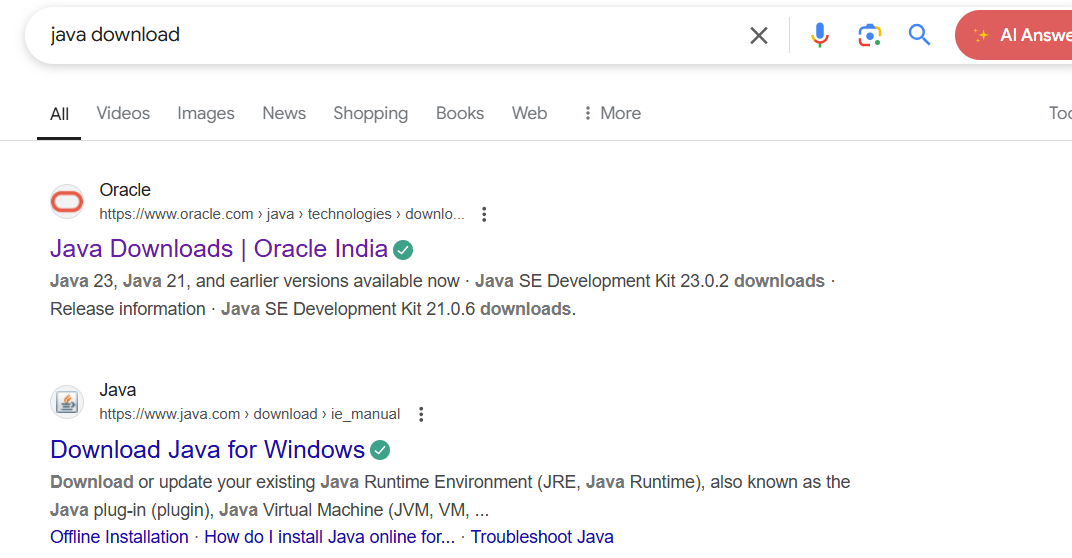
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Questions | Page No | Date | Sign |
| 1 | i. Download and install Java software  ii. Write a java program to print the message welcome to java programming  iii. Write a java program to print the name, roll no and section of the student |  |  |  |
| 2 | i. To calculate the area of the rectangle  ii. Program to convert the temperature in celsius to Fahrenheit.  iii. Program to calculate the simple interest.  iv. Program to find the largest of three numbers using the ternary operators.  v. Program to find the factorial of the number. |  |  |  |
| 3 | i. Create the java program for the cars with constructor and methods.  ii. Create the java program to withdraw and deposit money in the bank account. |  |  |  |
| 4 | i. Create the java program for the books by using the constructor and display its details using methods.  ii. Program to explain the final and the static variables. |  |  |  |
| 5 | i. Create the java program for the calculator.  ii. |  |  |  |
| 6 | i. Write a java program to create a vehicle class with a method display info.  ii. Write a java program to verify student eligibility for UG and PG.  iii. Create a calculator for overloaded method to perform addition.  iv. Create a class with calculated area that is overloaded for different shapes. |  |  |  |
|  |  |  |  |  |

**Week-1:**

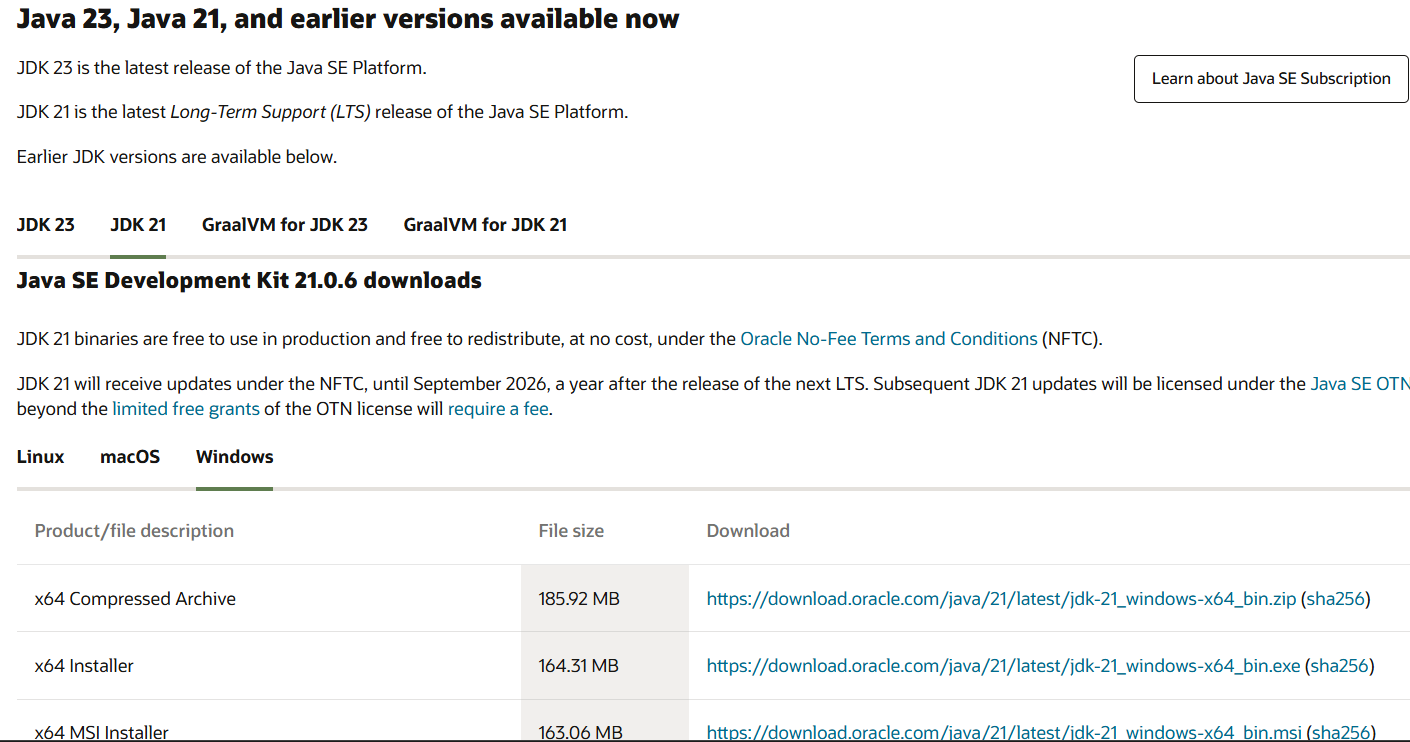
**Aim:** Download and install Java software

**Procedure:**

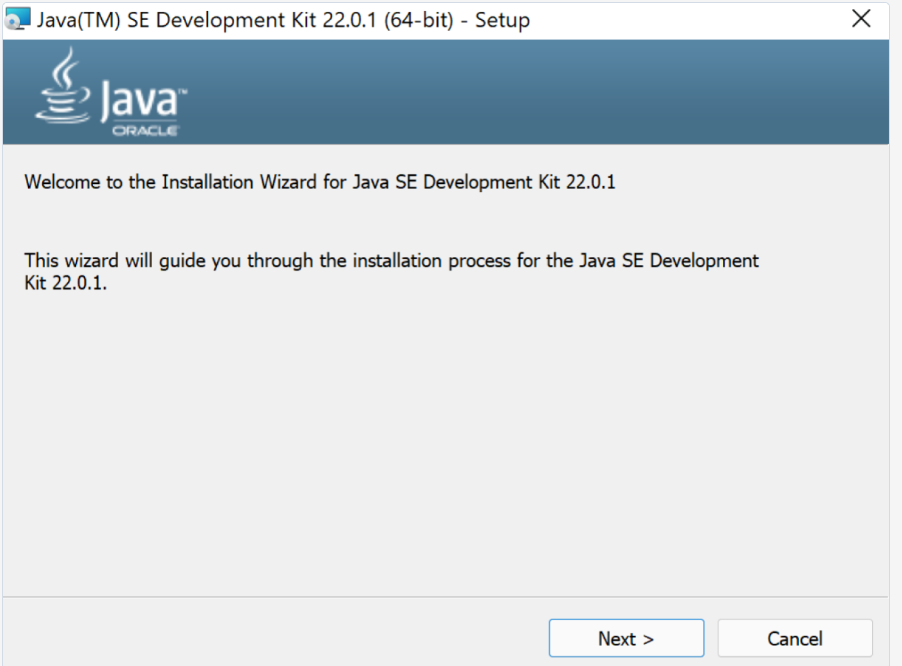
Step-1: Open any browser and search java download. Then open oracle website.



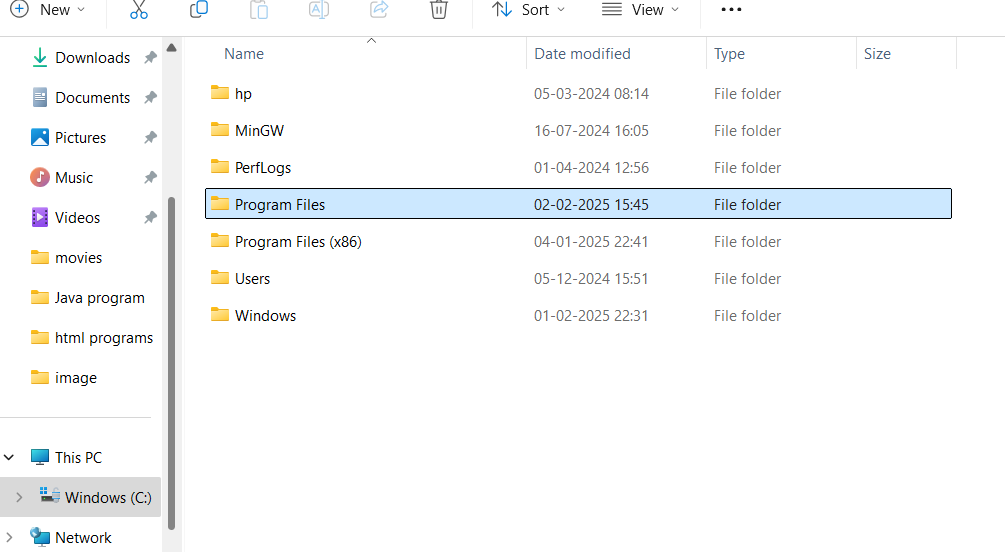
Step-2:It shows different java versions. In it download LTS version of your operating system. Download exe file



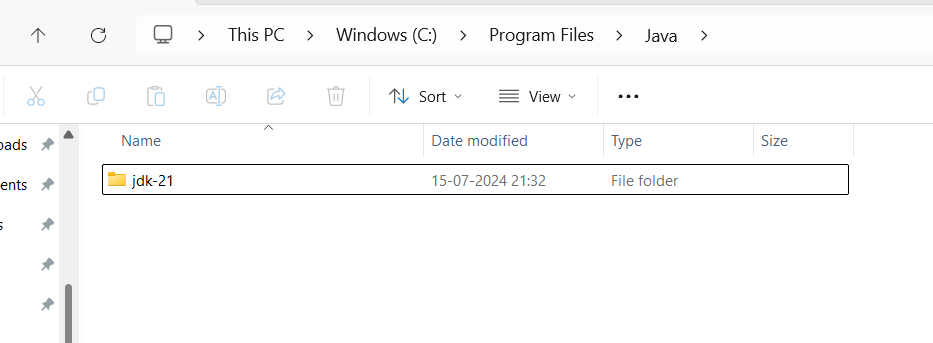
Step-3: Then open the downloaded file and allow all permissions and install it in the system.



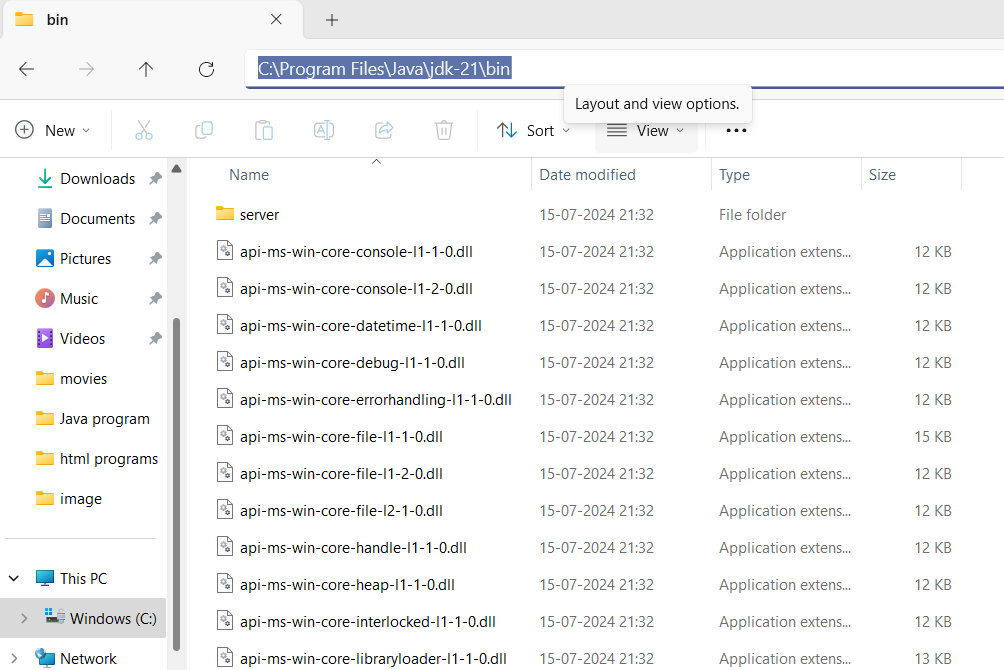
Step-4: Then open the files and go to the drive and enter into the program files.



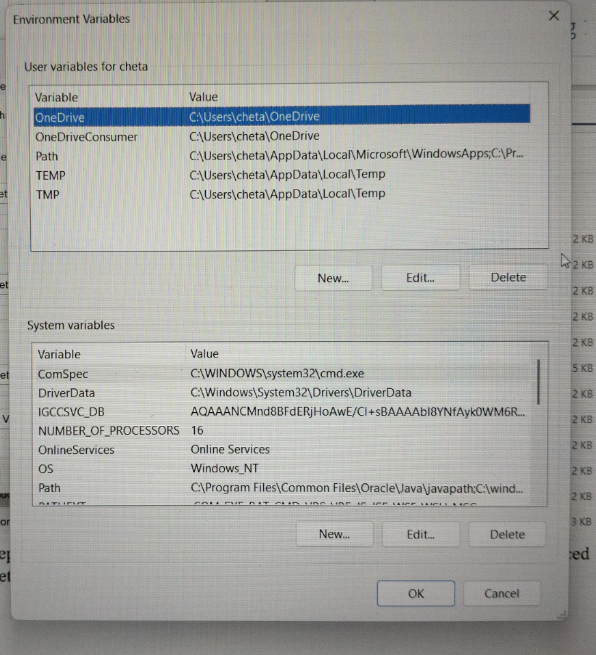
Step-5: Then open the program files and enter into the java, then open the jdk 21.



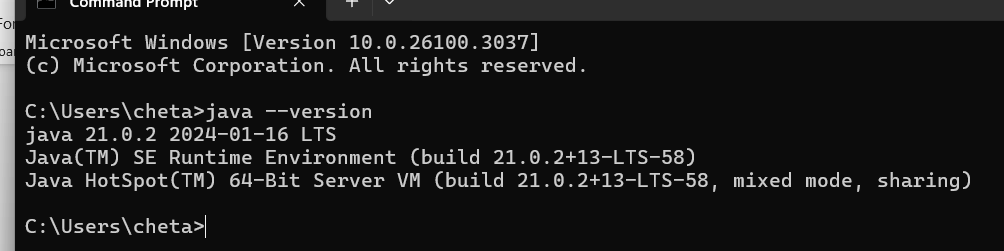
Step-6: Then enter into the bin. Then click on the on the bar to copy the path.



Step-7: Then right click on this PC and enter the properties, click on the advanced system settings, click on the environmental variables.



Step-8: Click on the new and paste the path and apply it. Then open the command prompt the enter java --version.



Hence java is installed.

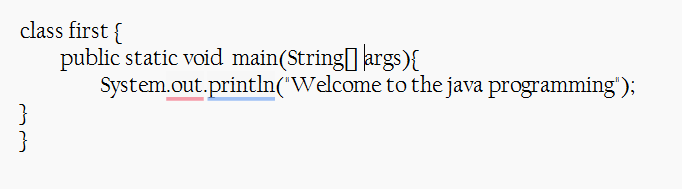
**Aim:** (ii) Write a java program to print the message Welcome to java programming

**Procedure:**

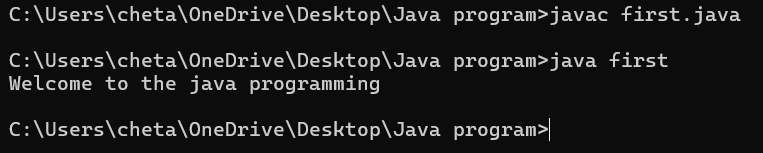
Step-1: To perform the first program open the new file in notepad and save it with name of the and .html.

Step 2: Write the program to print “Welcome to Java programming”.

**Code:**



Step-3: Open the file in the files and delete the path and type cmd. Then type javac name.java and type java first. Then we get the output.



**Error:**

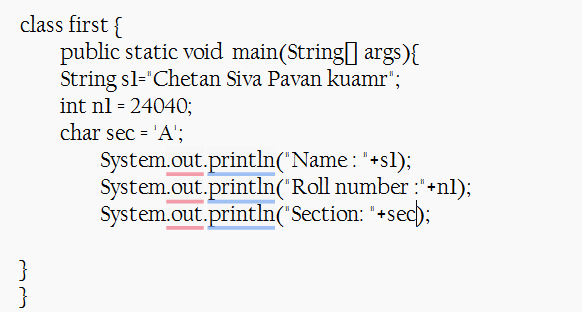
|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Identifiers Error** | Forgot to write the main method | Ensure public static void main(String[] args) { ... } is present |
| **Syntax Error** | Used s instead of S in Scanner | Use uppercase S → Scanner scanner = new Scanner(System.in); |

**Aim:** (iii) Write a java program to print the name, roll no and section of the student

**Procedure:**

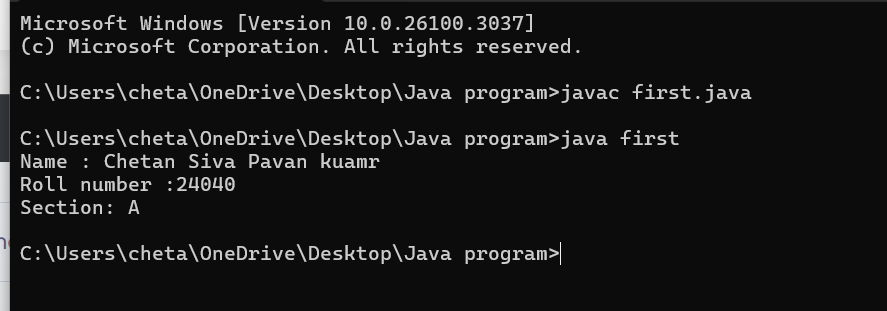
Step-1: Open the new file the save it and declare the varibles as per its date type.

**Code:**



Step-2: Save it and run it using the command prompt.

**Output:**



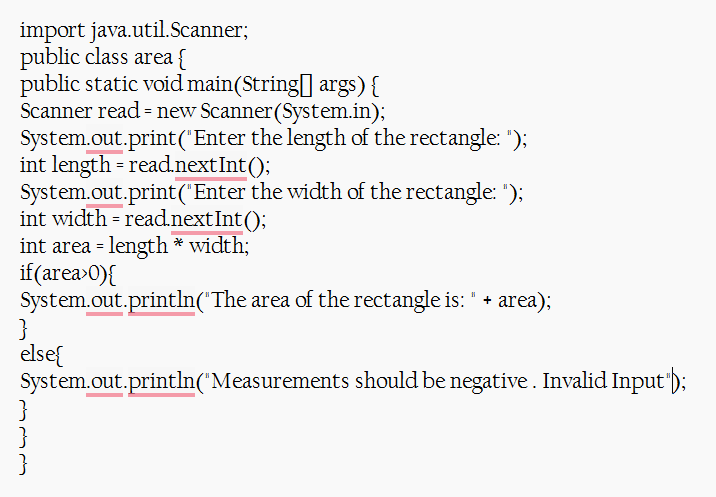
**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Description** | **Correction** |
| **Syntax Error** | Forgot to write ; (semicolon) | Ensure every statement ends with a semicolon (;) |
| **Variable Declaration Error** | Wrong variable declaration | Use correct data types and proper syntax |

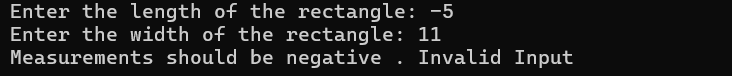
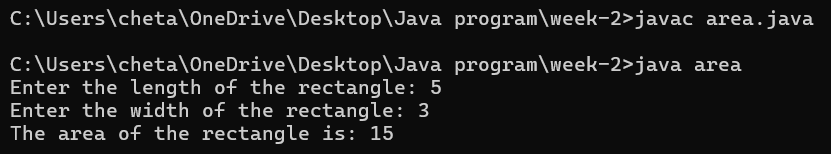
**Week-2:**

**Aim:** (i) Program to calculate the area of the rectangle.

**Code:**



**Output:**



**Explanation or Important points :**

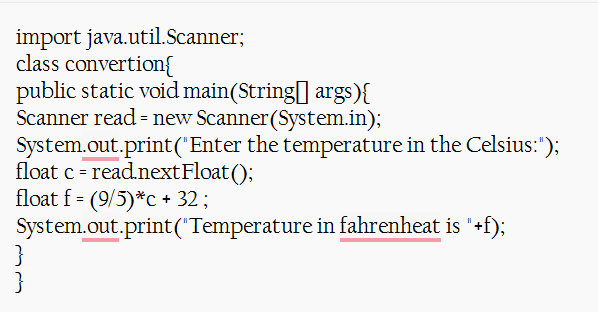
* Here the input is taken from the user. For taking the input we imported the util package and the Scanner class.
* By using it the input is taken from the user.
* For area we used the area logic and for negative entries we print the mesurements should not be negative

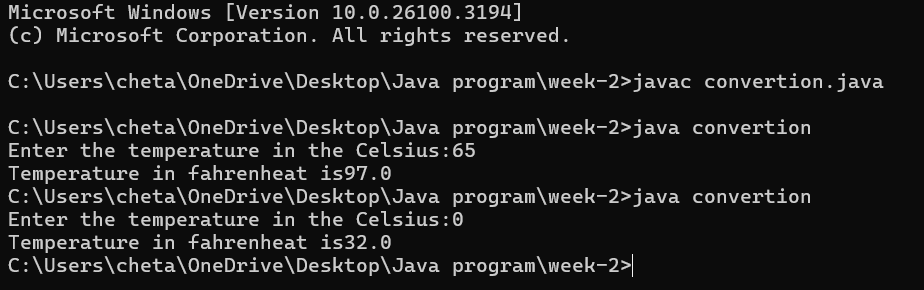
**Errors:**

|  |  |  |
| --- | --- | --- |
| **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) { ... }) |

**Aim:** (ii)Program to convert the temperature in celsius to Fahrenheit.

**Code:**



**Output:** 

**Explanation or Important points :**

* Here the formula to convert the celcius into Fahrenheit is f = (9/5)\*c +32 .
* Here the scanner class is included.

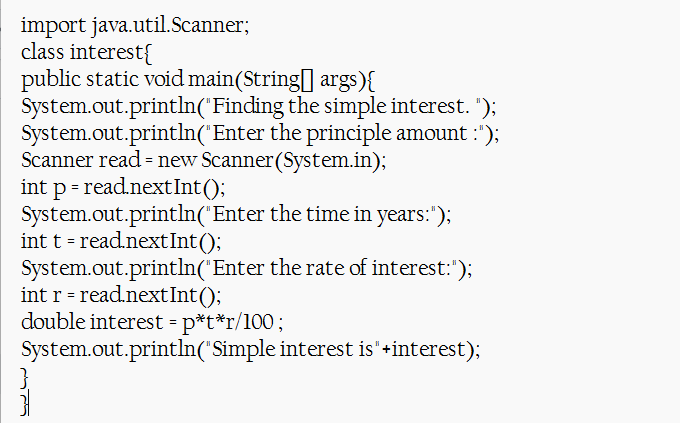
**Error:**

|  |  |  |
| --- | --- | --- |
| **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 |

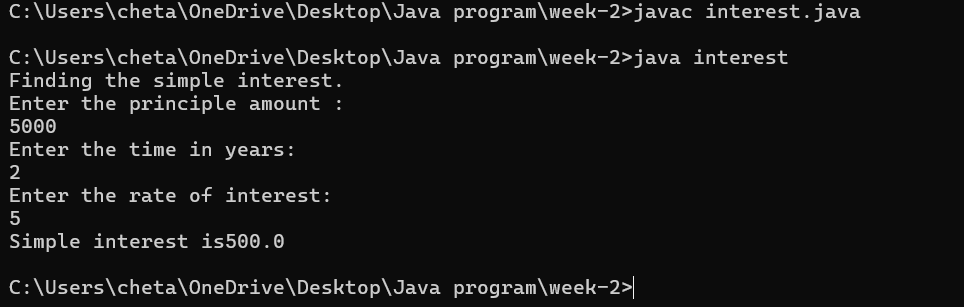
**Aim:** (iii) The program will calculate the simple interest.

**Procedure:**

**Code:**



**Output:**



**Explanation or Important points :**

* In the above program the scanner class is included to take the inputs like principle amount, years, rate of interest .
* By using the formula of simple interest we calculate the value.

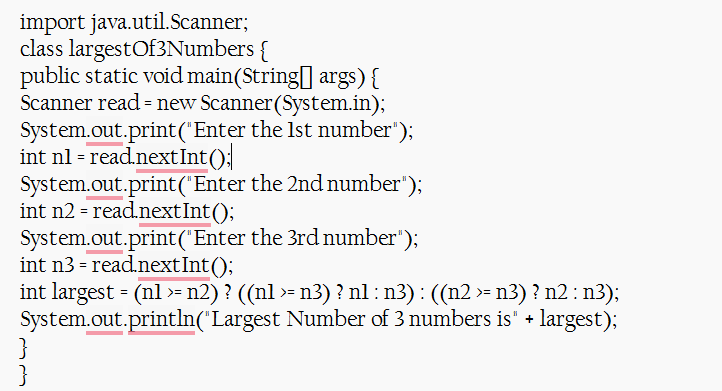
**Error:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
| **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 | |

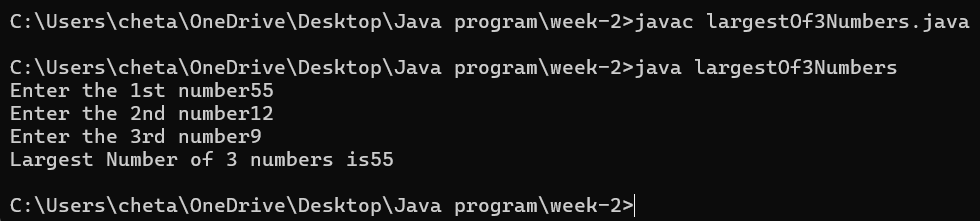
**Aim:** (iv)Program to find the largest of three numbers using the ternary operators.

**Procedure:**

**Code:**

****

**Output:**



**Explanation or Important points :**

* Here the Nested Ternary Operators is used to find the larger among the three numbers.
* If n1 is greater then n2 then it check with the n3 and return the larger value similarly for the n2 is greater then n1.
* By using the Nested Ternary Operator to find the greater the number

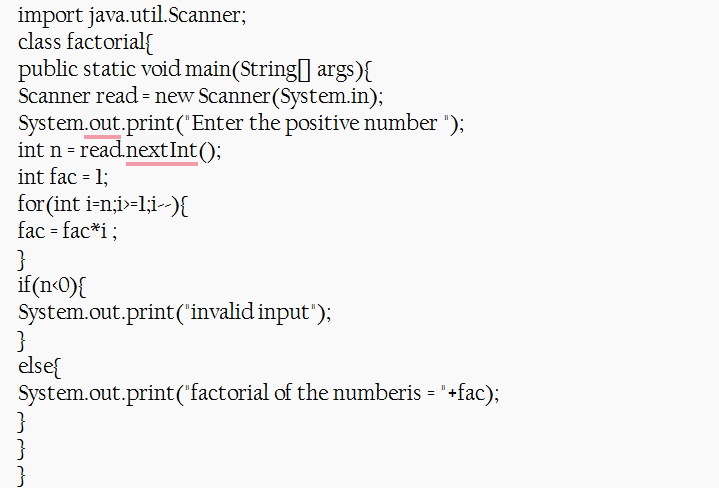
**Error:**

|  |  |  |
| --- | --- | --- |
| **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 |

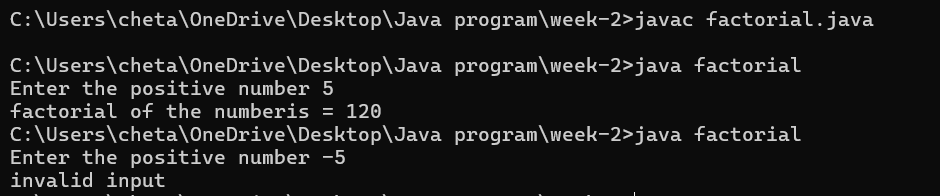
**Aim:** (v) Program to find the factorial of the number.

**Procedure:**

**Code:**



**Output:**



**Explanation or Important points :**

* To find the factorial of the number we use for the loop based on the condition to satisfy the loop runs repeatedly.
* We usually decrement the operator and multiply with the fact then we print the result

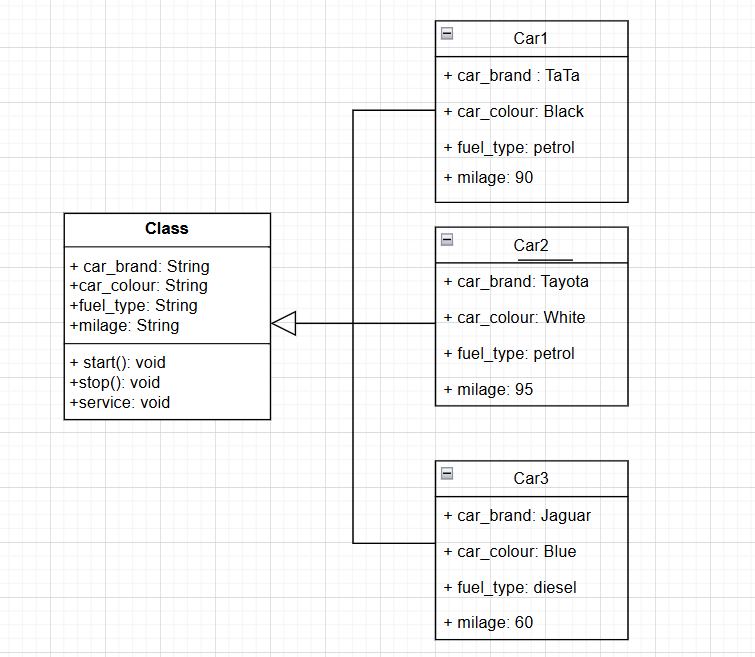
**Error:**

|  |  |  |
| --- | --- | --- |
| **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 |

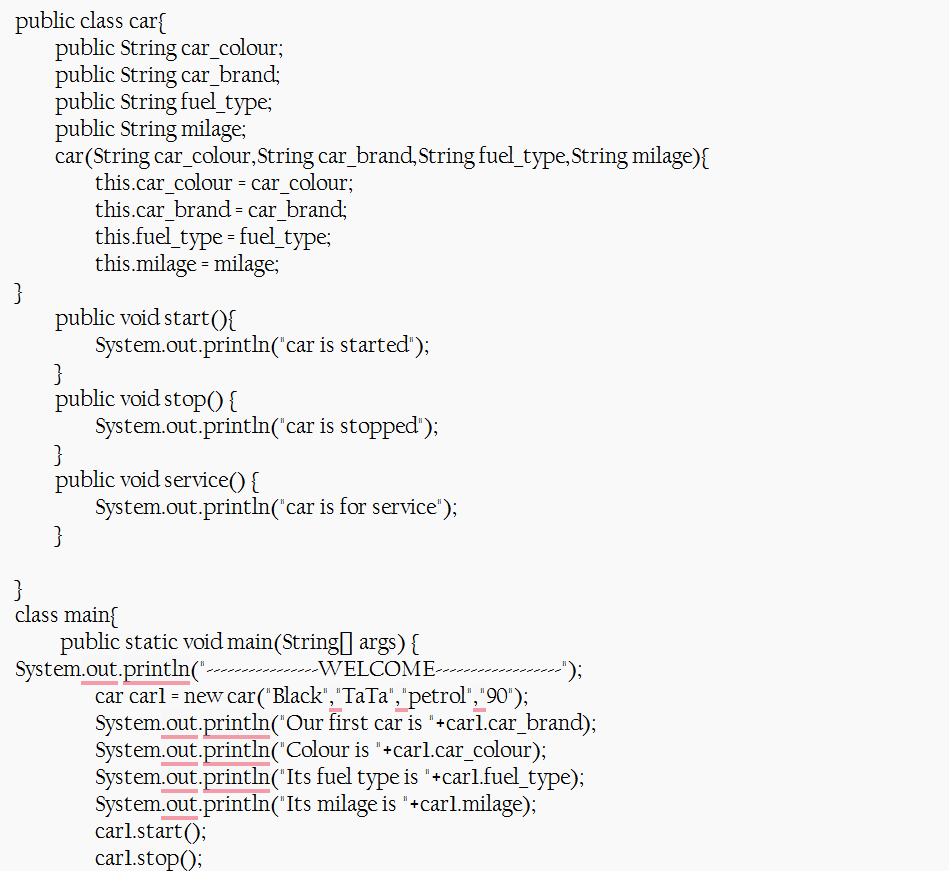
**Week-3:**

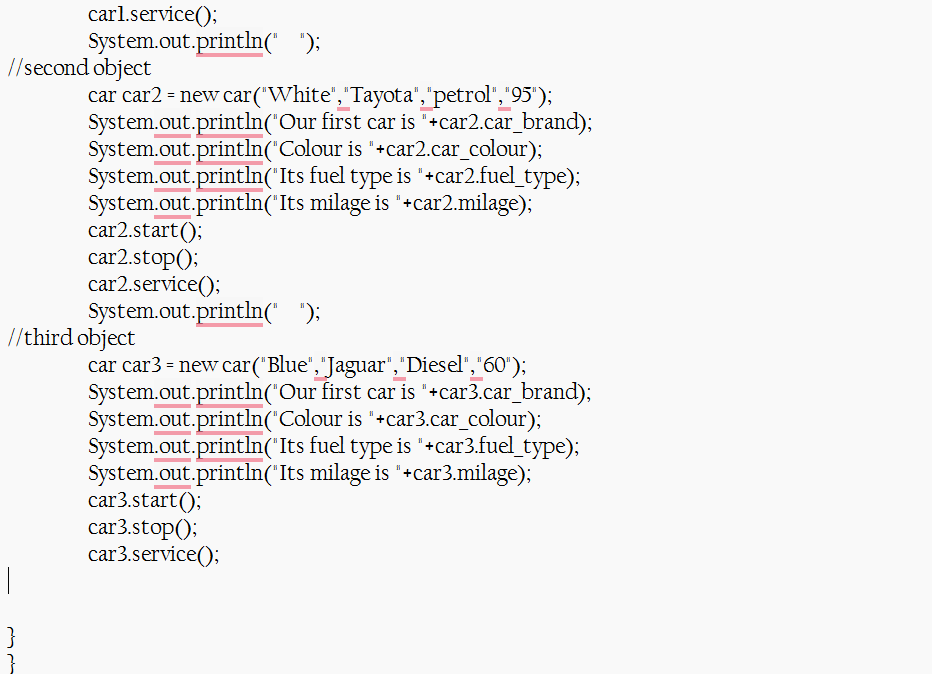
**Aim: (i)** Create the java program for the cars with constructor and methods.

**Class Diagram:**

****

**Program:**

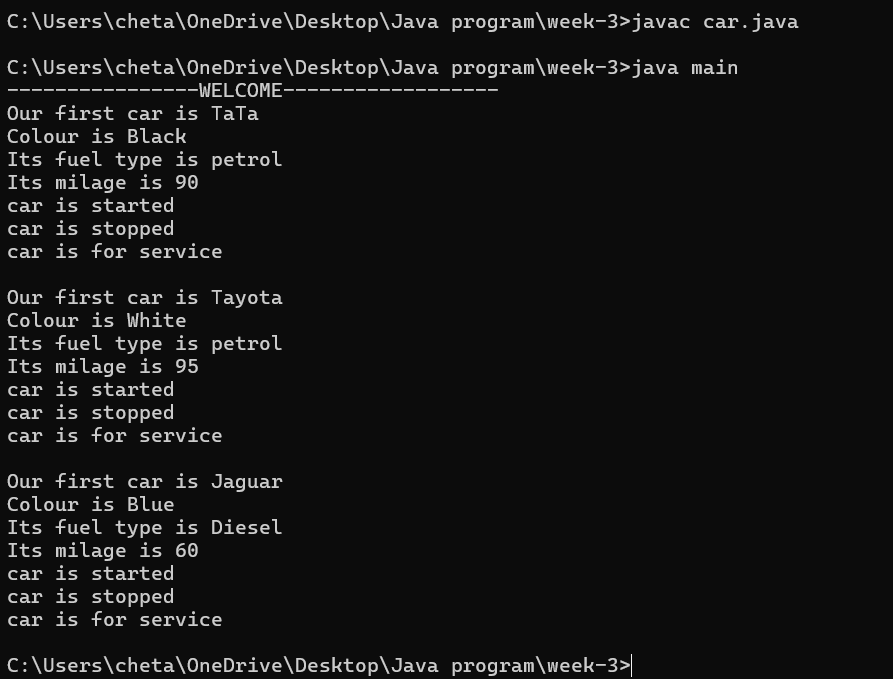
****

****

**Explanation or Important points :**

* By using a constructor(the method without return type and same name of the class) we pass the values for the each object and call the the methods using the object.
* By using this we can create many objects and the code reusability increases.

**Output:**

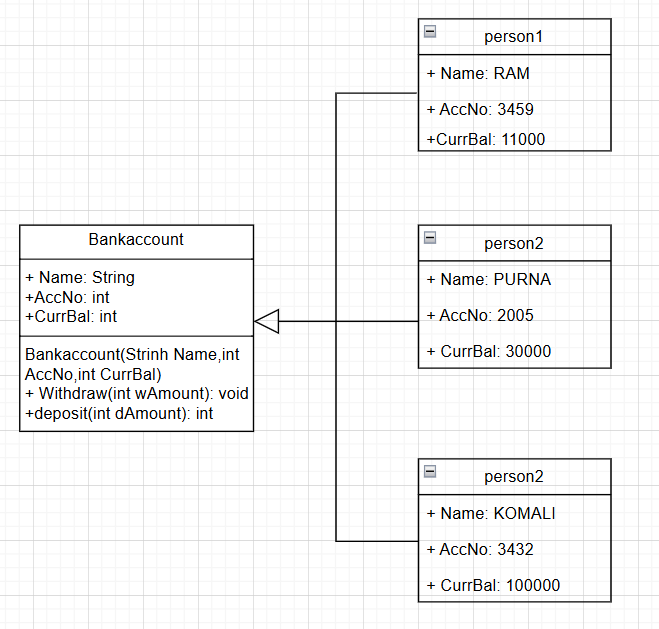
****

**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; |

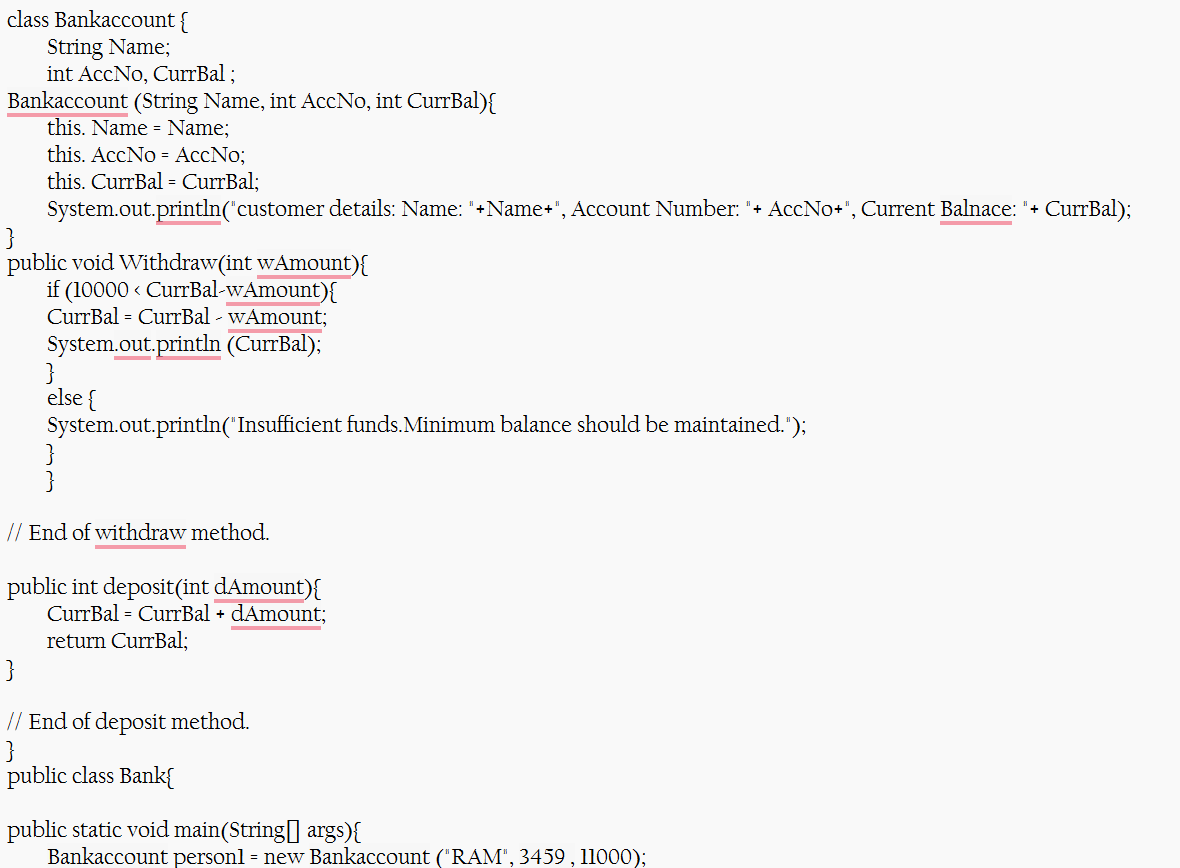
**Aim: (ii)** Create the java program to withdraw and deposit money in the bank account.

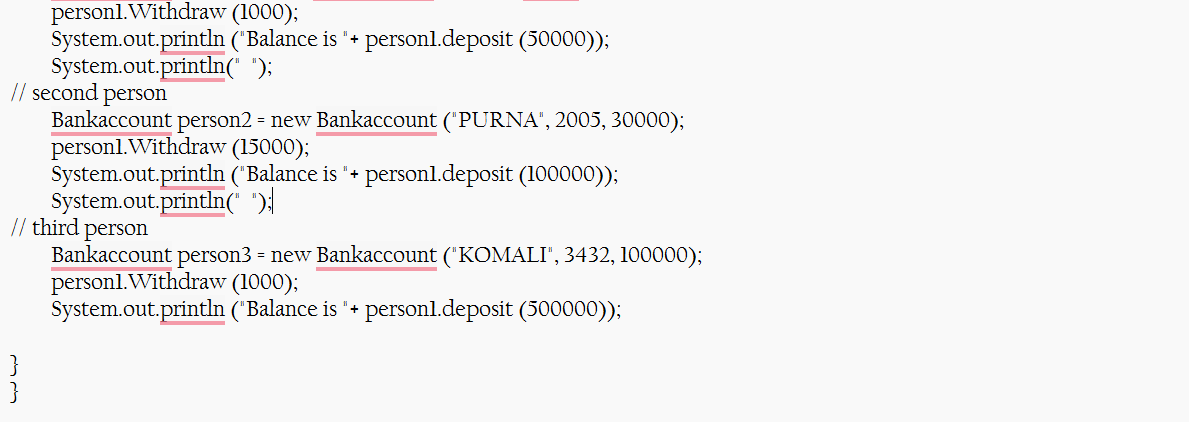
**Class Diagram:**

****

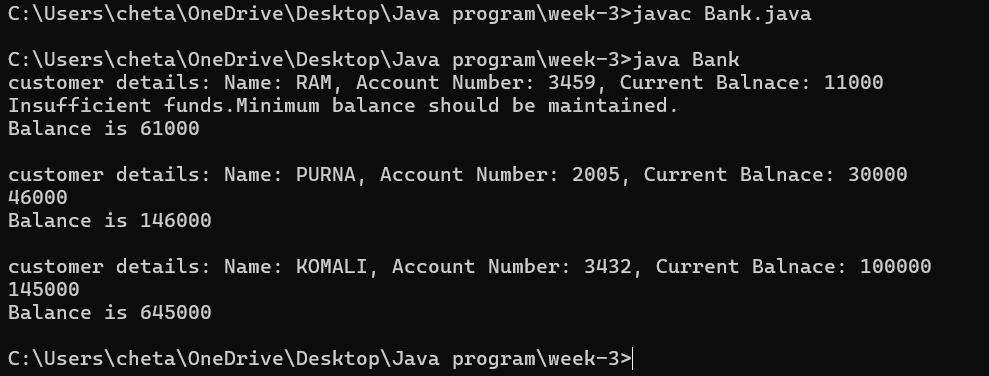
**Procedure:**

Code:





**Output:**

****

**Explanation or Important points :**

* Here we developed the bank account class and then the two methods for withdrawing and depositing money.
* Then we created the objects and the called the different methods.
* By this we condition (withdrawal amount the greater the balance) for the withdrawal.

**Error:**

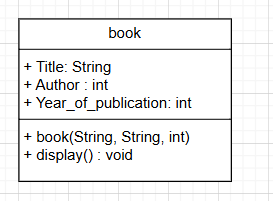
|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected code** |
| **Class Name Capitalization** | class Bankaccount | class BankAccount (Java follows PascalCase for class names) |
| **Object Naming Issue** | BankAccount person-1 (hyphen is not allowed) | BankAccount person1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

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Week-4:**

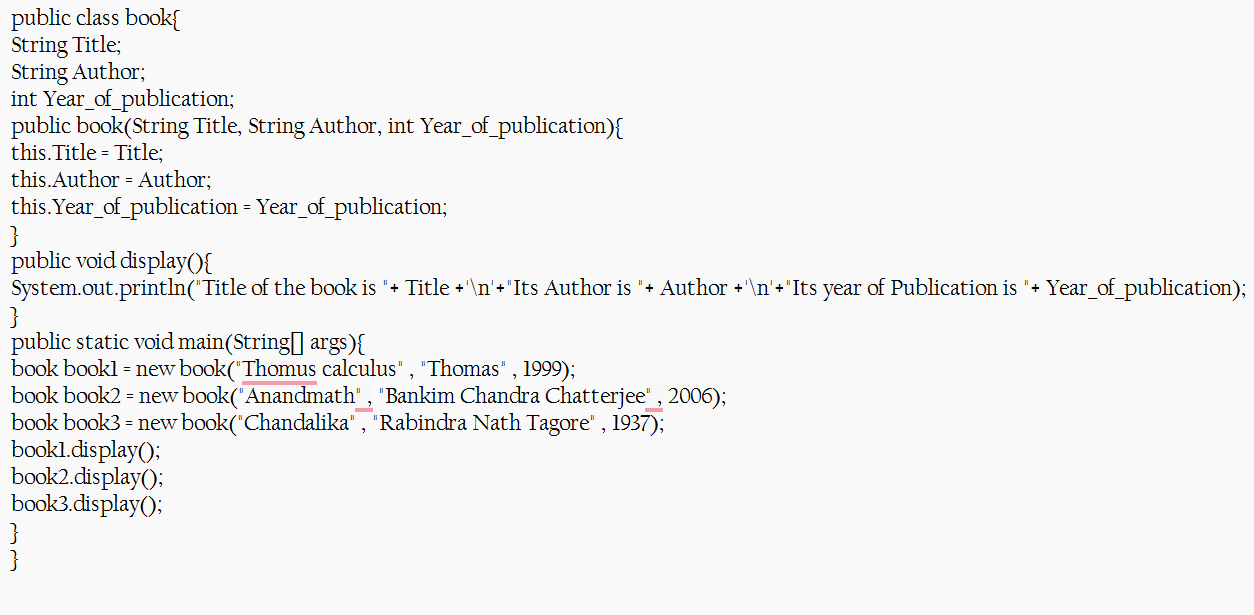
**Aim: (i)** Create the java program for the books by using the constructor and display its details using methods.

**Class Diagram:**

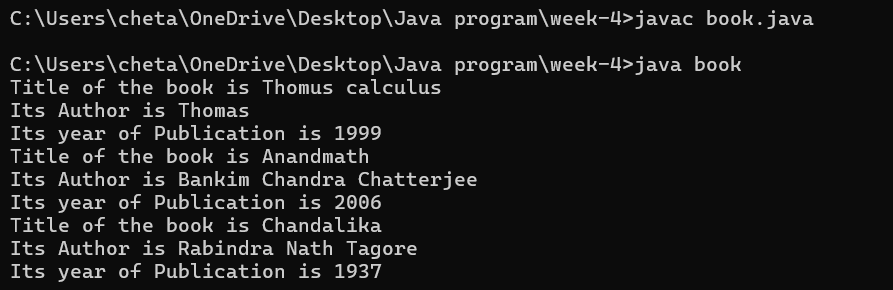
****

**Procedure:**

**Code:**



**Output:**

****

**Explanation or Important points :**

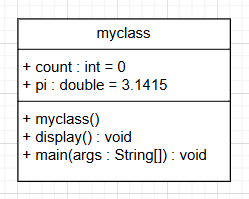
* We created a book class with different attributes. By using the constructor we pass the details of the book and the display method for the showing the details.

**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) |

**Aim: (ii)** Program to explain the final and the static variables.

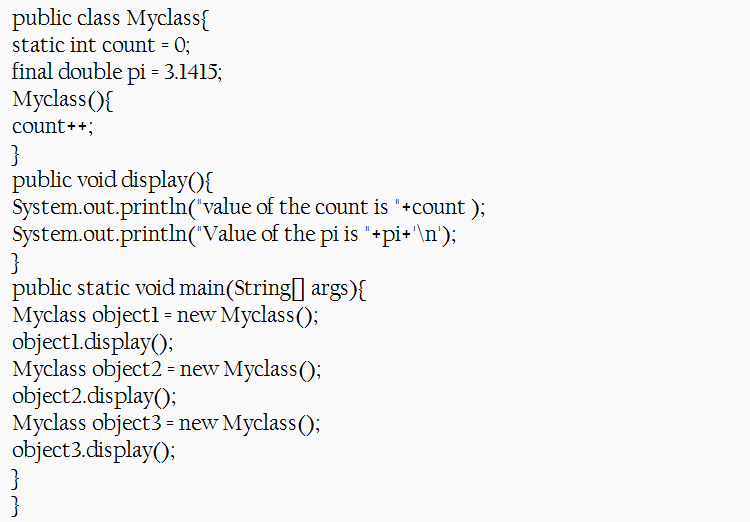
**Class Diagram:**

****

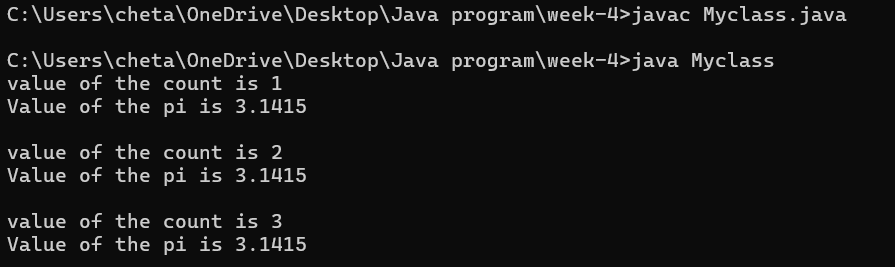
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**Procedure:**

**Code:**



**Output:**

****

**Explanation or Important points :**

* Here we used the final and static variable.
* For final it is constant over the file , static is chages in all places if it is changed. By using those keywords we printed the output.

**Error:**

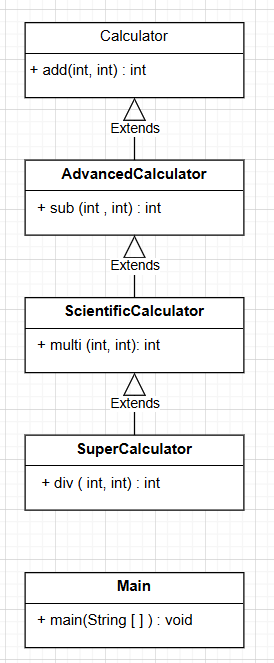
|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **Attempt to Modify final Variable** | pi = 3.14; (if added inside the constructor or method) | Remove this line (final variables cannot be reassigned) |
| **Incorrect Class Name** | public class Myclass | public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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Week-5:**

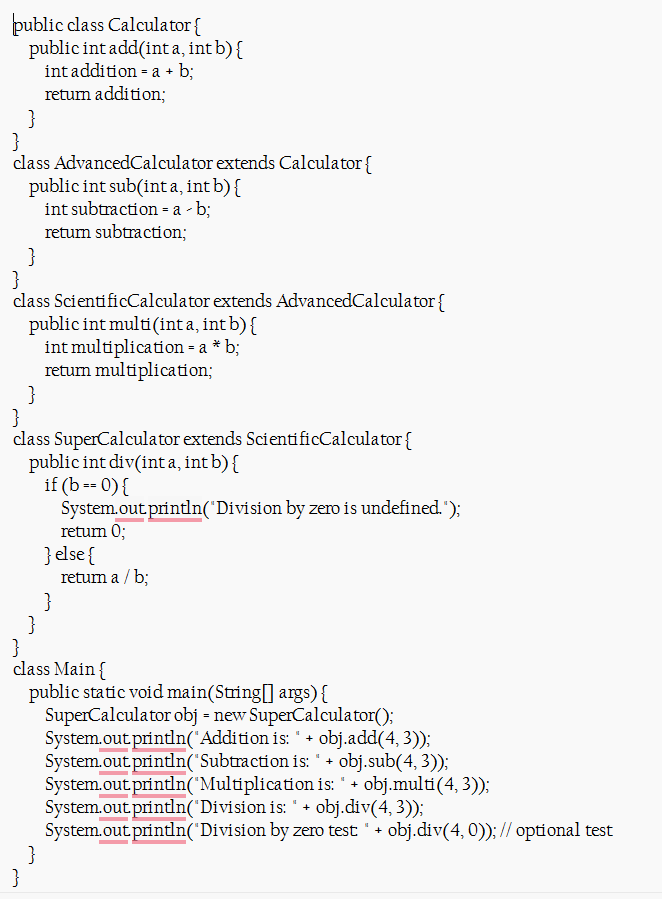
**Aim: (i)** Create the java program for calculator using multi-level inheritance.

**Class Diagram:**

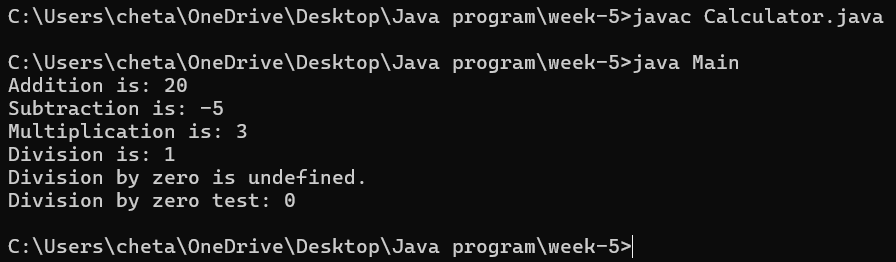
****

**Procedure:**

**Code:**



**Output:**

****

**Explanation or Important points :**

* By using the inheritance concept we are extending all the classes and creating the objects for the superCalculator and accessing all methods from it.
* That by using this we decrease the code by inheritance concept.

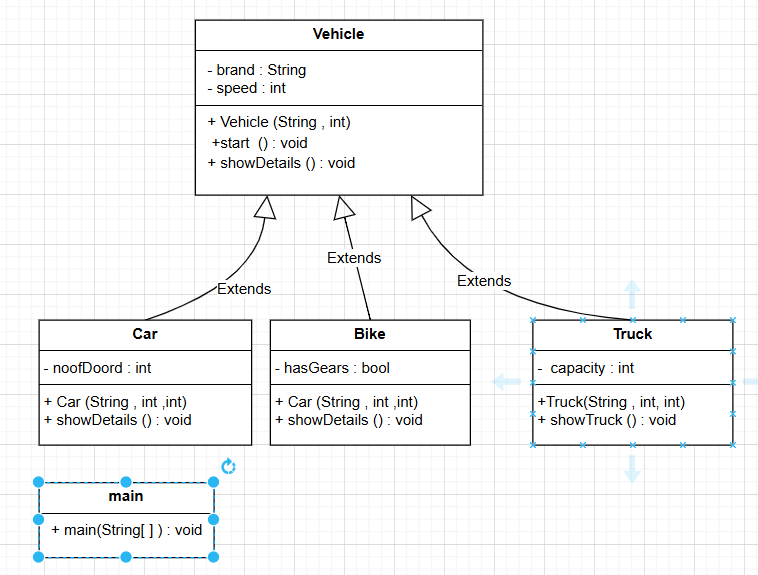
**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **Invalid file name** | Given calculator as the file name | For public class file should be class name |

**i**

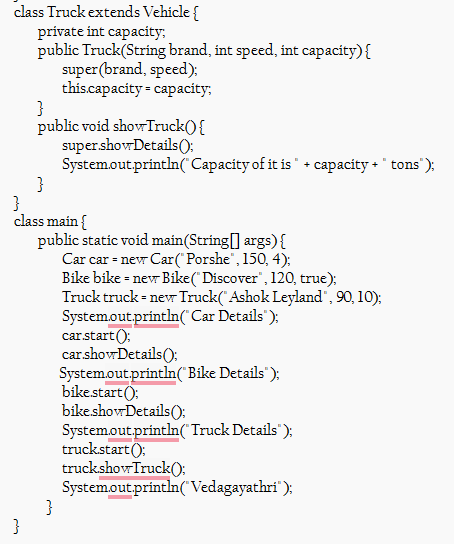
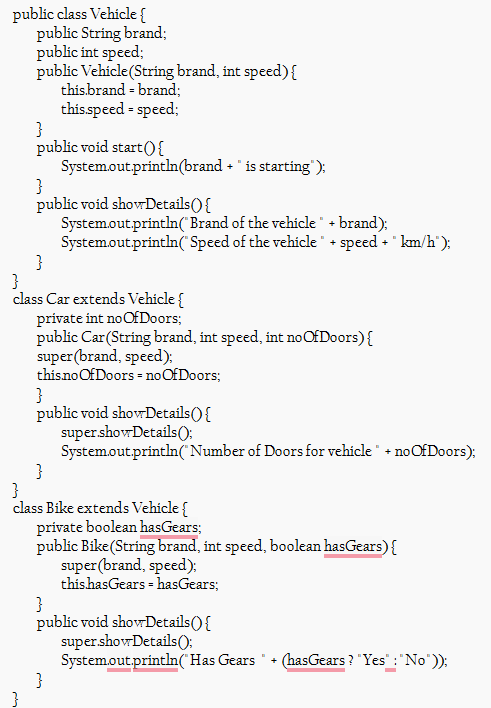
**Aim: (ii)** A vehicle rental company wants a system to manage details of vehicles available for rent, including cars, bikes, and trucks. Each vehicle should store basic information such as brand and speed. Cars must include an additional property for the number of doors, while bikes should indicate whether they have gears. The system should also provide a function to display the details of each vehicle and indicate when a vehicle is starting.

**Class Diagram:**

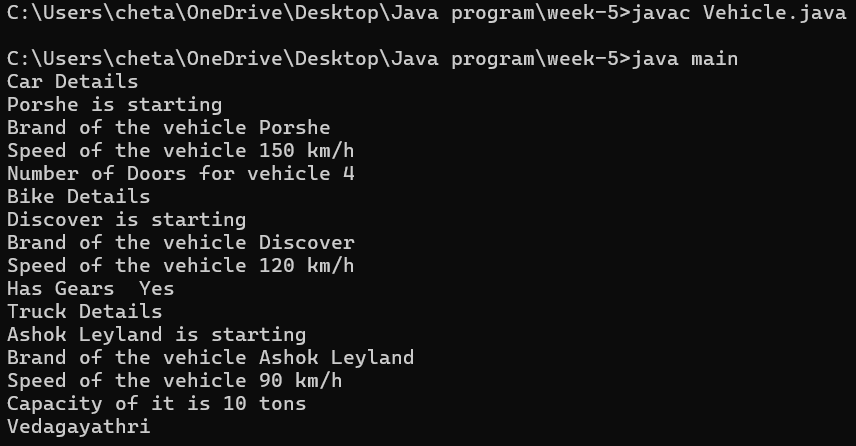
****

**Procedure:**

Code:



**Output:**

****

**Explanation or Important points :**

* 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
* 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.

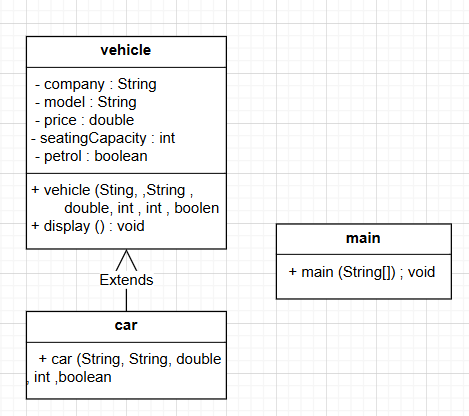
**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **Syntax Error** | Missing comma between constructor parameters | public Transport(String model, int maxSpeed) |

**Week-6:**

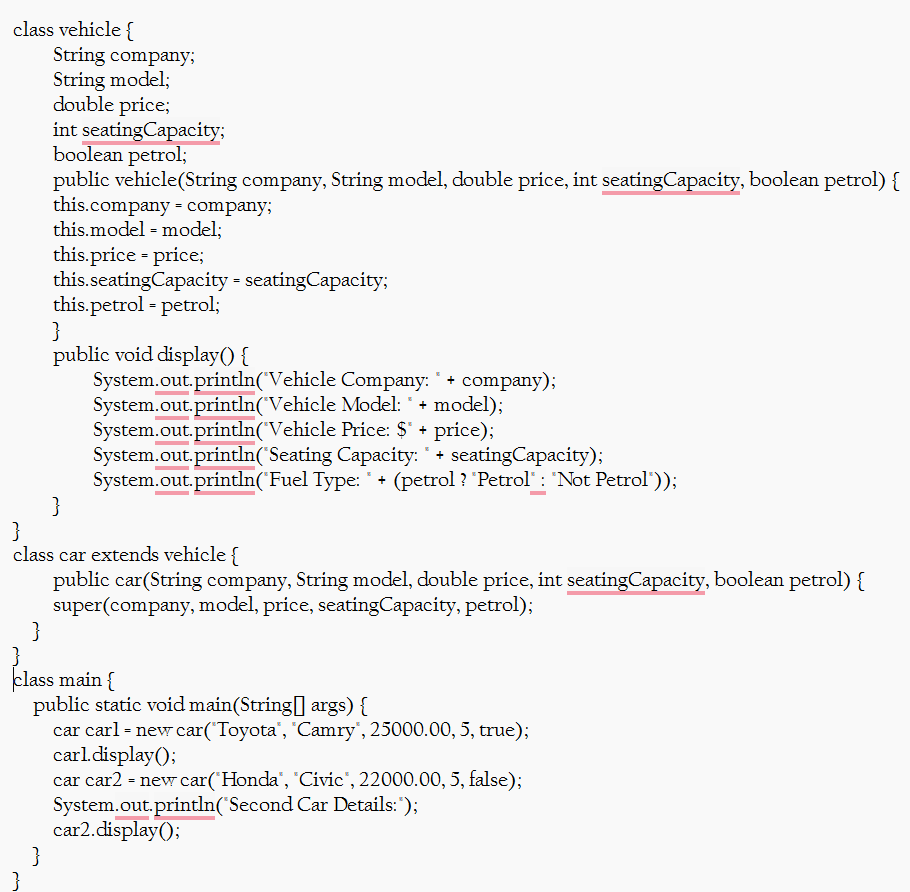
**Aim:** (i) Write a java program to create a vehicle class with a method display info . Overisde this method in car subclass to provide specific information.

**Class Diagram:**

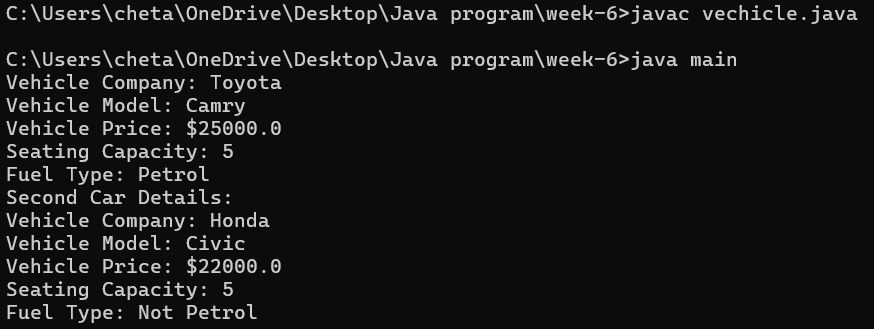
****

**Procedure:**

**Code:**

****

**Output:**

****

**Explanation or Important points :**

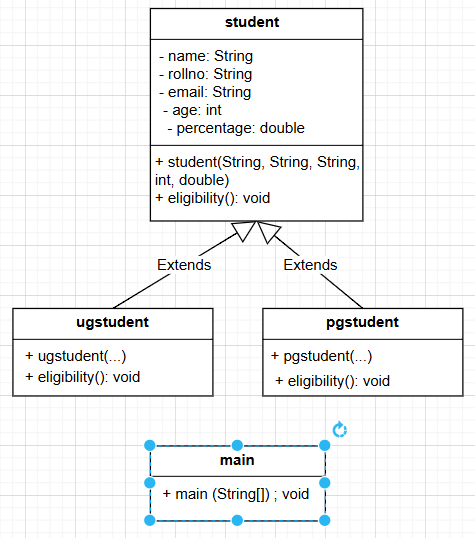
* We taken the vehicle class and then it is extended to the car class .
* Then we prepared the objects to the car class and then we perform the display method.

**Error:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **i.Unused variable** | Declared instance variable in vehicle class | Use this.Car\_company = Car\_company; to set instance variable |
| **ii. Reductant method override** | The car class just calls super.displayinfo(..) without adding any new behaviour | Either remove the override if it is not needed or add a custom behavior |

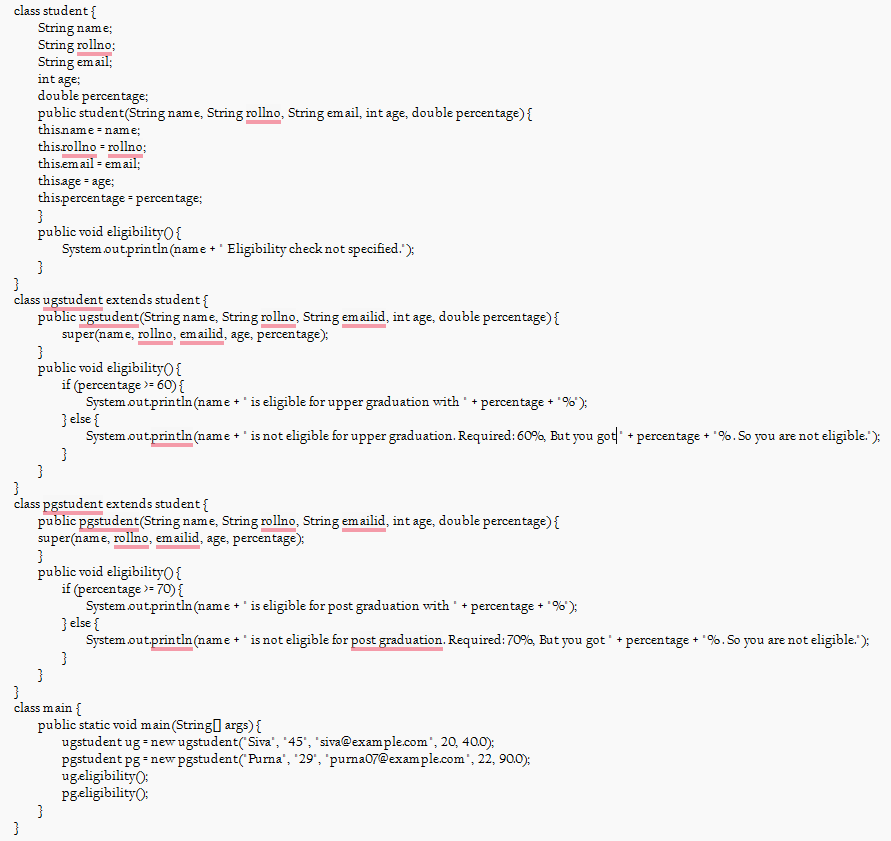
**Aim:** (ii)A college is developing an automated admission system that verifies students eligibility for undergraduate (UG) and postgraduate (PG) programs. Each program has different eligibility for UG and PG program qualifications.UG admission requires a minimum of 60% and PG requires a minimum of 70%.

**Class Diagram:**

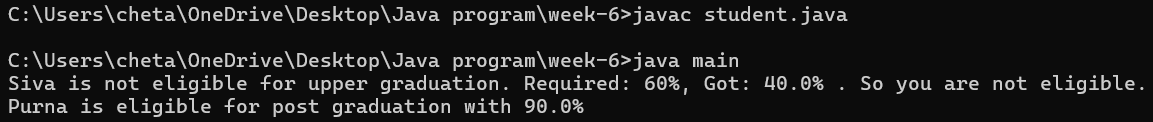
****

**Procedure:**

**Code:**

****

**Output:**

****

**Explanation or Important points :**

* In this the Student class is create then for the upper-graduation another class is created and the input of student details and if the percentage is greater then 60% he is eligible for this college.
* Similarly for the post-graduation the percentage is greater then the 70% then they are eligibile for this college.

**Error :**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Code** | **Corrected Code** |
| **i. Logical error** | Age is declared as a string, but it represents a number | int age; |
| **ii. Data validation error** | Email string uses a comma instead of a dot | **“**gorgilla.com**”** |

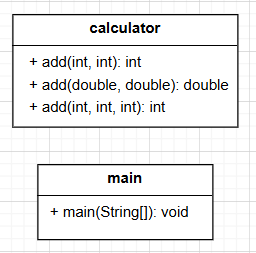
**Aim:** (iii)Create a calculator class with overloaded methods to perform addition

i. Add two integers.

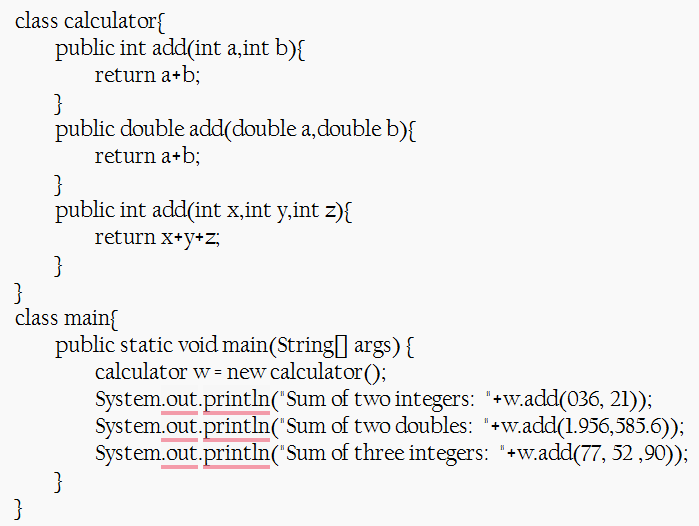
ii. Add two doubles.

iii. Add three integers.

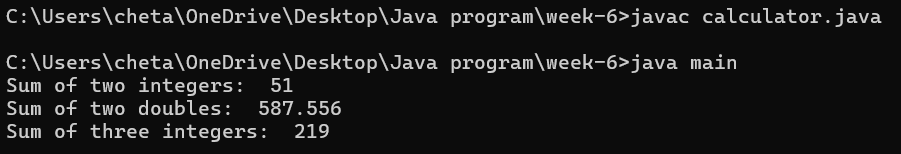
**Class diagram:**

****

**Procedure:**

**Code:  
**

**Output:**

****

**Explanation or Important points :**

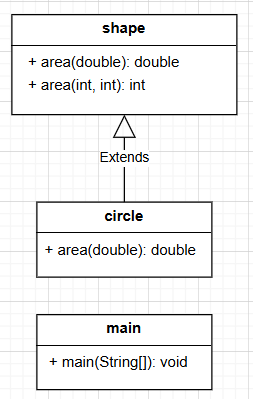
* Here we added the two numbers with different data types overloading concept is used here.
* Used the same method with different parameters. Then printed the result.

**Errors:**

|  |  |  |
| --- | --- | --- |
| **Error type** | **Incorrect error** | **Corrected error** |
| **Java naming convention violation** | class name Calci is not standerd | class calculator |

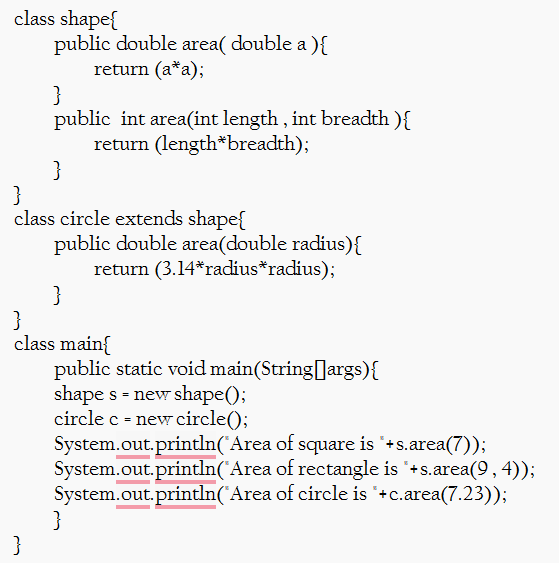
**Aim:** (iv)Create a shape class with a method-calculated area that is overloaded for different shapes. Ex square, rectangle…Then create a subclass circle that overrides the Calculated Area() method for a circle.

**Class diagram:**

**s**

**Procedure:**

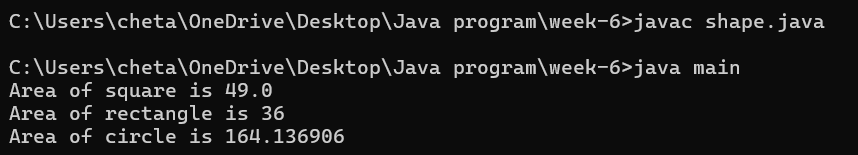
**Code:**



**Explanation or Important points :**

* Here the shape class is used for calculating the area.
* That the circle is extended from the shape class and the same method area of overloading to find the area of the circle.

**Output:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| **Error Type** | **Incorrect Error** | **Corrected Error** |
| **Syntax Error** | Forget to end the print statement with ; | Ended the statement with ; |