**LAB REPORT**

**OBJECT-ORIENTED PROGRAMMING**

****

|  |
| --- |
| **Submitted by** |
| Name B.PRANATHI |
| Roll No AV.SC.U4CSE24022 |
| Year/Sem/Section 1st YEAR/SEM-2/CSE-A |
| Date of Submission |

|  |
| --- |
| **Submitted to** |
| Name DR. Rajkumar |
| Department CSE |
| Designation Asst. Professor |

|  |  |
| --- | --- |
| Marks |  |

**INDEX**

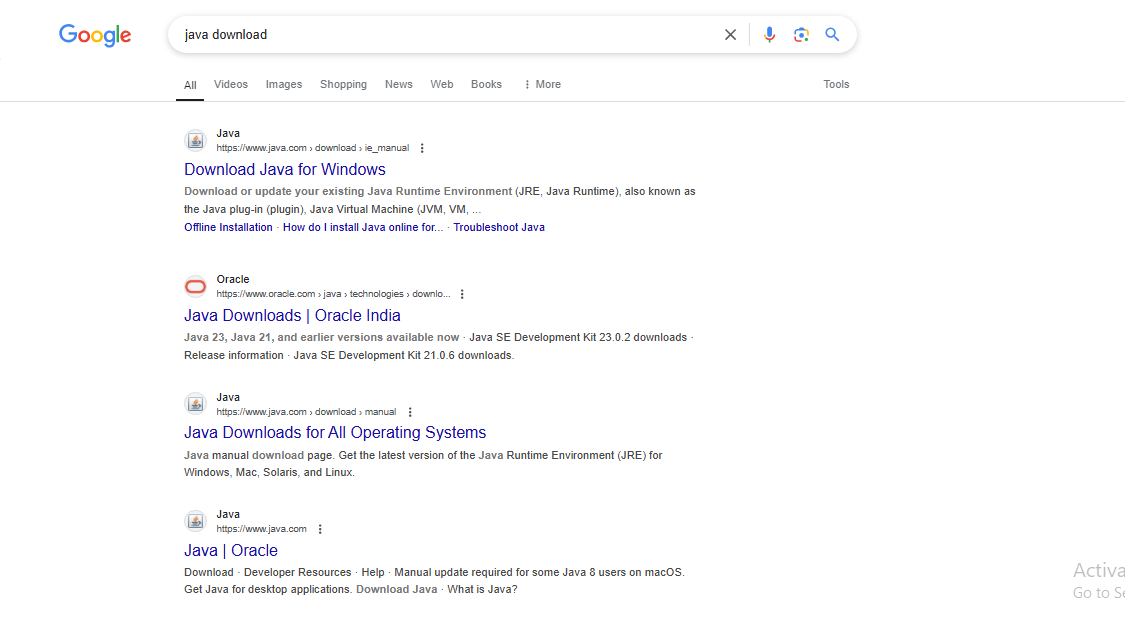
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Title** | **Date** | **Page No.** | **Signature** |
| Week 1 |  | 27-01-2025 |  |  |
| 1. | How to download and install Java Software. |  | **6** |  |
| **2.** | Write a Java Program to print the message “Welcome to Java Programming”. |  | **9** |  |
| **3.** | Write a Java program that prints: Name, Roll.no. , section of a student. |  | **10** |  |
| Week 2 |  | 10-02-2025 |  |  |
| **1.** | Write a java program to calculate the area of a rectangle. |  | **11** |  |
| **2.** | Write a java program to temperature from Celsius to Fahrenheit and vica-versa. |  | **11** |  |
| **3.** | Write a java program to calculate the simple interest. |  | **12** |  |
| **4.** | Write a java program to find the largest of three numbers, using ternary operator. |  | **13** |  |
| **5.** | Write a java program to find the factorial of a number. |  | **14** |  |
| Week 3 |  | 24-02-2025 |  |  |
| 1. | Create a java program with the following instructions.   1. Create a class with name car 2. Create four attributes named car color,carbrand,fuel type and milage.   Create three methods named start(),stop(),service(). IV. Create three objects named car1,car2,car3. |  | **16** |  |
| 2. | Create a java program with the following instructions.   1. Create a class with named bank account 2. Create two methods named withdraw(),deposit(). |  | **17** |  |
| Week 4 |  | 03-03-2025 |  |  |
| 1. | 1) Write a java program with class named “book” the class should contain various attributes such as “title”,”author”,”year\_publication”.It should also contain a constructor with parameters which initializes “title”,”author” and “year\_publication”.   * Create a method which displays the details of the book i.e.”title”,”author” and   “year\_Publication”.   * Display the details of two books by creating two objects. |  | 20 |  |
| 2. | :To create a java program with class named “My\_class” with a static variable “count” of “int” type,initialized to 0 and a constant variable “pi” of type  “double” initialized to 3.1415 as attributes of that class.Define a constructor for “My\_class” that increments the count variable each time an object of “My\_class” is created finally print the final values of “count” and “pi” values. |  | 21 |  |
| Week 5 |  | 10-03-2025 |  |  |
| 1. | create a calculator using the operations including add, sub, multi and div using multilevel inheritance and display the desired output |  | 23 |  |
| 2. | 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 |  | 25 |  |
| Week 6 |  | 29-03-2025 |  |  |
| 1. | 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). |  | 27 |  |
| 2. | 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%. |  | 28 |  |
| 3. | Create a calculator class with overloaded methods to perform addition of:  1. Add two integers  2. Add two doubles  3. Add three integers |  | 30 |  |
| 4. | 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. |  | 31 |  |
| Week 7 |  | 14-04-2025 |  |  |
| 1. | create Java program to create an abstractclass 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. |  | 33 |  |
| 2. | : 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. |  | 34 |  |
| 3. | : 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. |  | 36 |  |
| Week 8 |  | 21-04-2025 |  |  |
| 1. | Write a java program to create an interface shape with the getperimeter() method. Create three classes Rectangle,circle and triangle that implement the shape interface. Implement the getperimeter() method for each of the three classes. |  | 39 |  |
| 2. | write a java program to create an interface playable with a method play() that takes no arguments and returns void. Create three classes Football,Volley ball and Basket ball that implement the playable interface and override the playable method to play the respective sports. |  | 40 |  |
| 3. | Write a java program to implement a login system using interfaces. |  | 42 |  |
| Week 9 |  | 26-04-2025 |  |  |
| 1. | Write a java program to create a method that takes integers as parameters and throws an exception if the number is even. |  | 44 |  |
| 2. | Write a java program to create a method that reads a file and throws an exception if the file is not found. |  | 45 |  |
| 3. | Write a java program to handle arthimatic exception using try, catch and finally. |  | 46 |  |
| 4. | Write a program to stimulate a university system using inner classes.   * Create an outer class named university with a variable universityName. * Inside it, define two non-static inner classes.   1.Department- With variables like deptName, deptCode and a method to display department details.  2.Student- With variables like studentName, rollNumber and a method to display details.  3.Create an object for each class and call their methods to display their details along with their universityName. |  | 47 |  |
| Week 10 |  | 02-06-2025 |  |  |
| 1. | Write a java program to generate a password for a student using his/her initials and age. The password displayed should be the string consists of first character of first name, middle name, last name with age. |  | 50 |  |
| 2. | Design and implement a java program that will do the folleing operations to this string “Welcome! You are practicing String concepts”.   * Convert all alphabets to capital letters and print out the result. * Convert all alphabets to lowercase letters and print out the result. * Print out the length of the string. * Print out the index of the concept. |  | 51 |  |
| 3. | Implement a java program using the below array methods.   * Sorting the elements (numbers & strings) of an array. * Convert the array elements into String. * Fill the part of an array. * Copy the elementsof one array into the other. |  | 52 |  |
| 4. | Implement a java program using the below Array list method.   * Insert an element at particular index in the array list. * Modify an element in the array list. * Access an element from the array list. * Remove an element from the Array list. * Clear the elements from the array list. |  | 53 |  |

**WEEK-1**

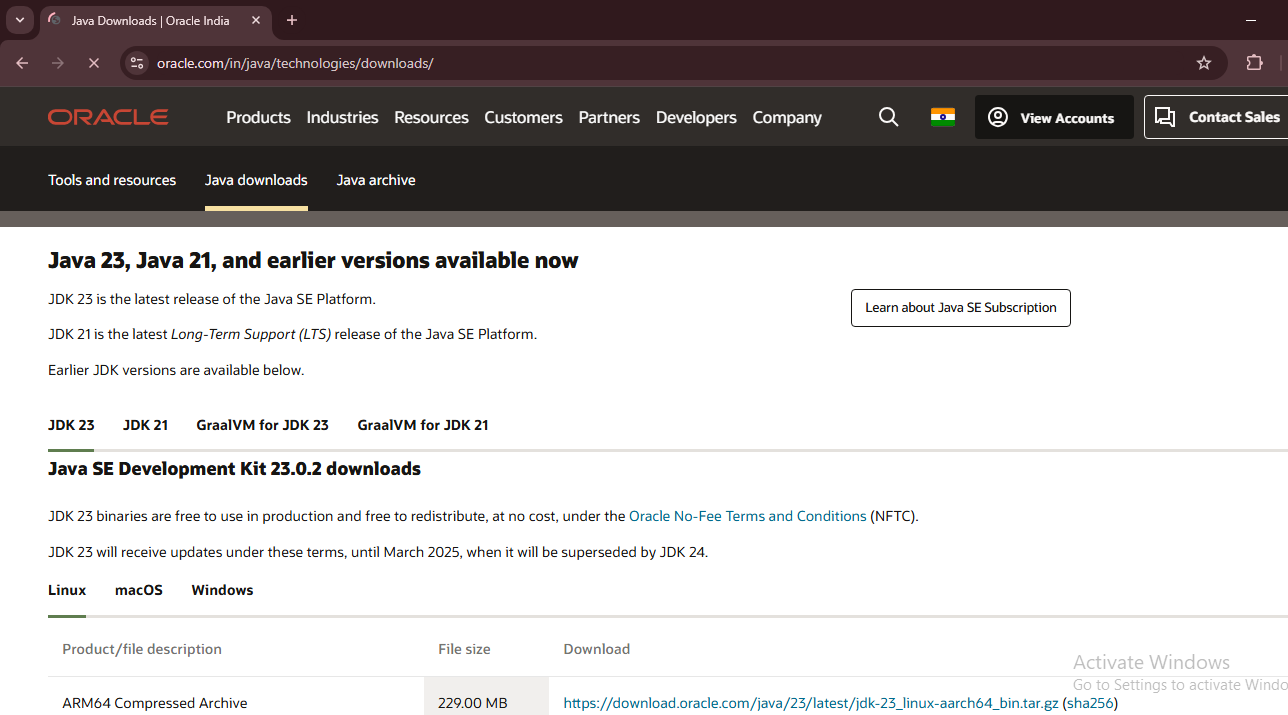
**PROGRAM-1:**

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

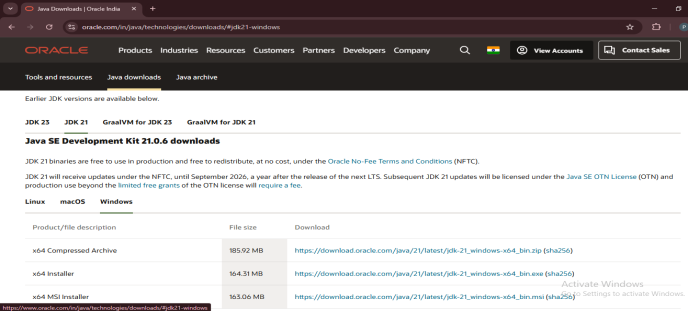
Step-1: search java download .



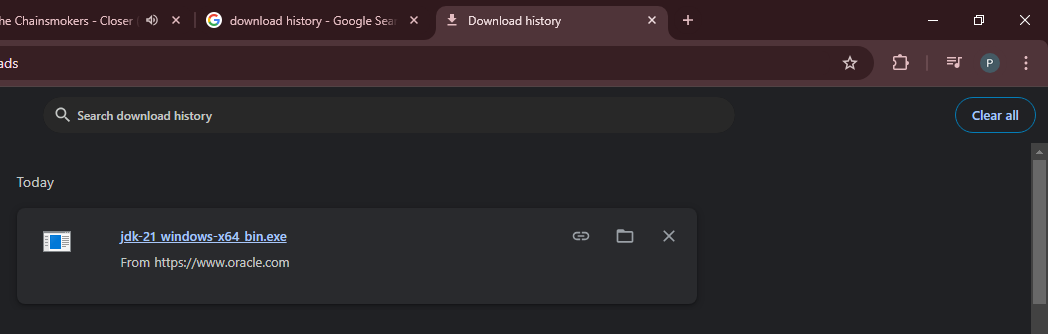
Step-2: click oracle java download (official website).



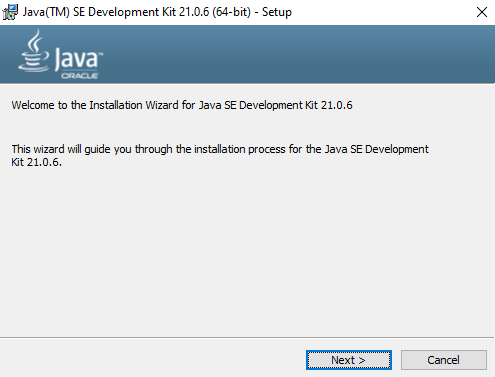
Step-3: click JDK 21, after that click windows.



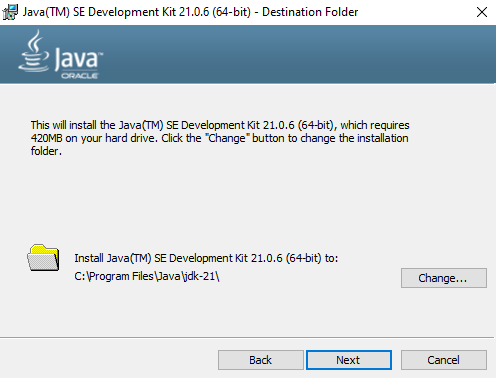
Step-4: click \*64 installer and then install.

****

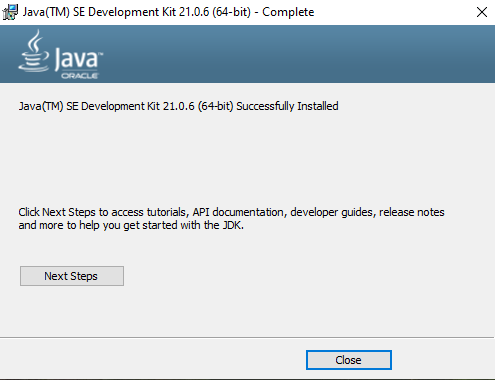
**Step-5:**

****

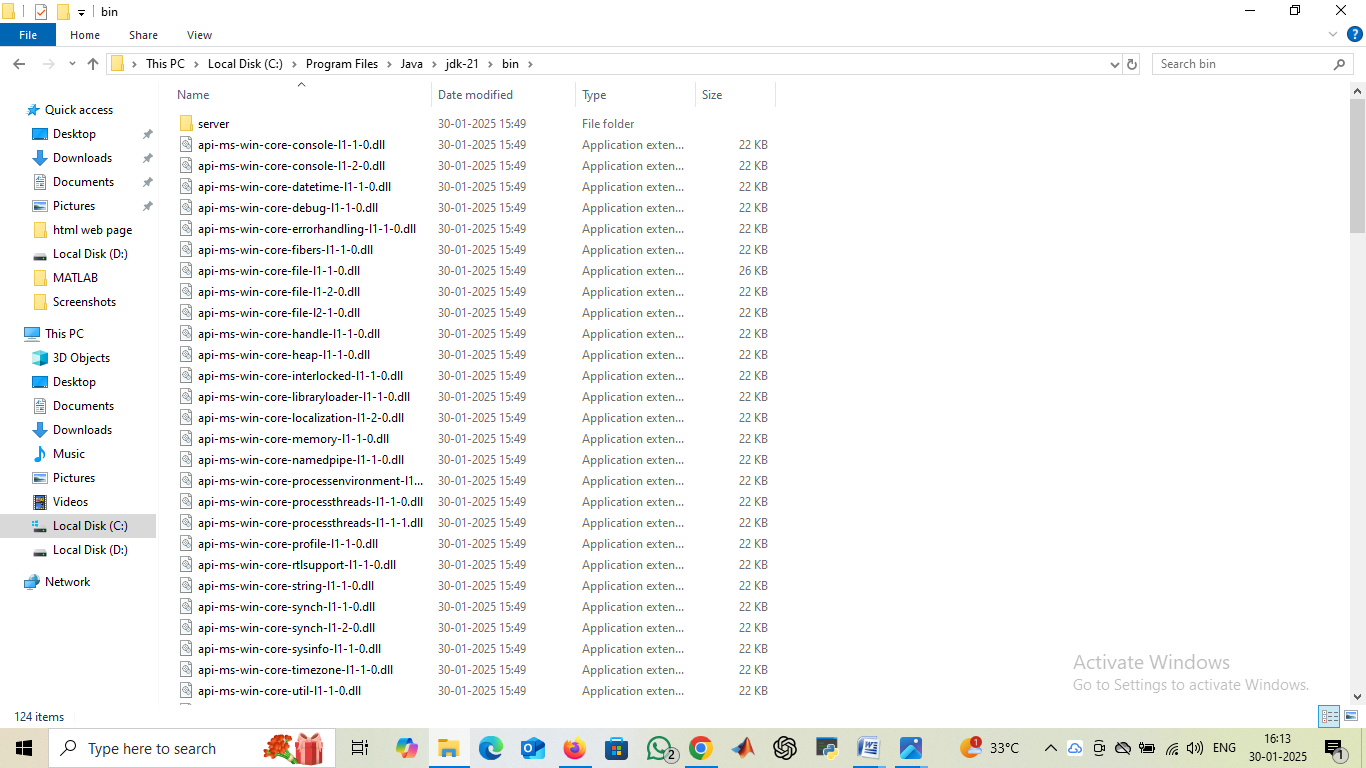
**Step-6:**

****

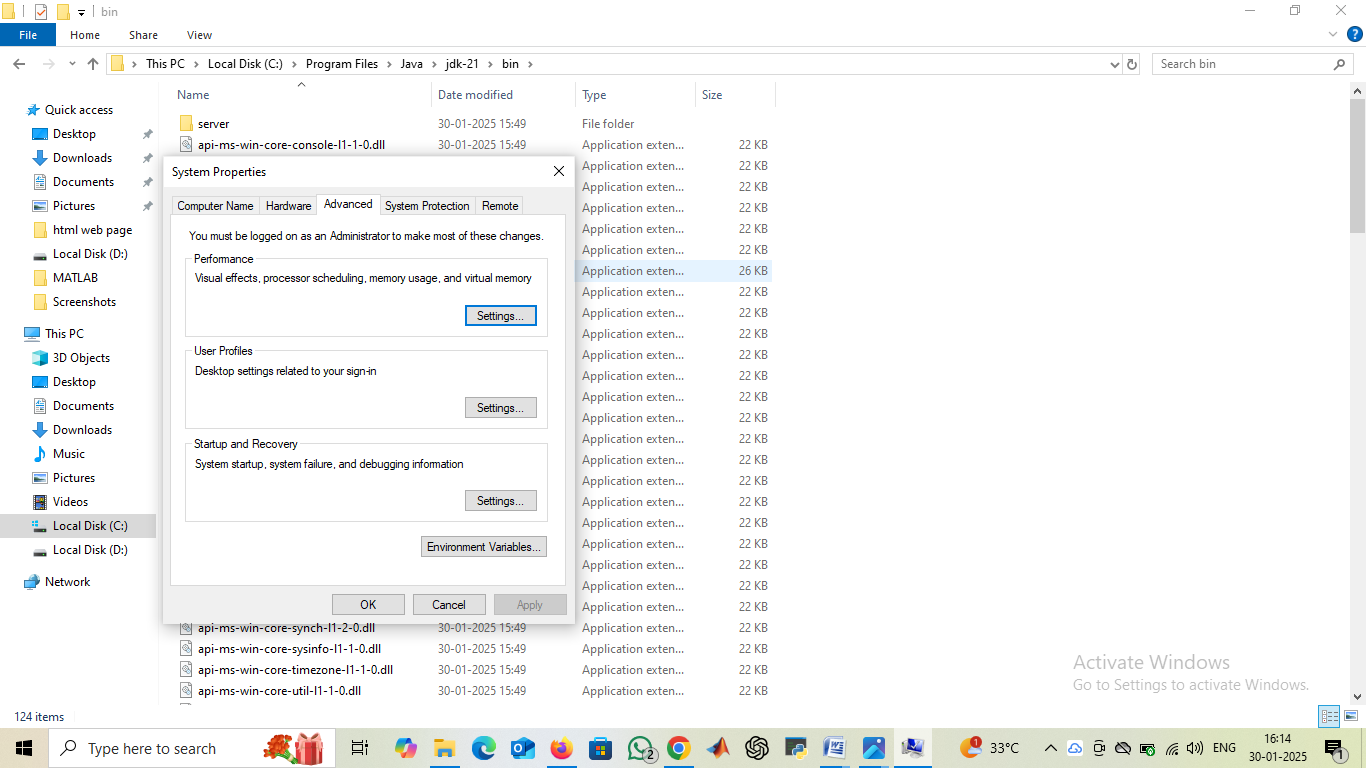
**Step-7:**

****

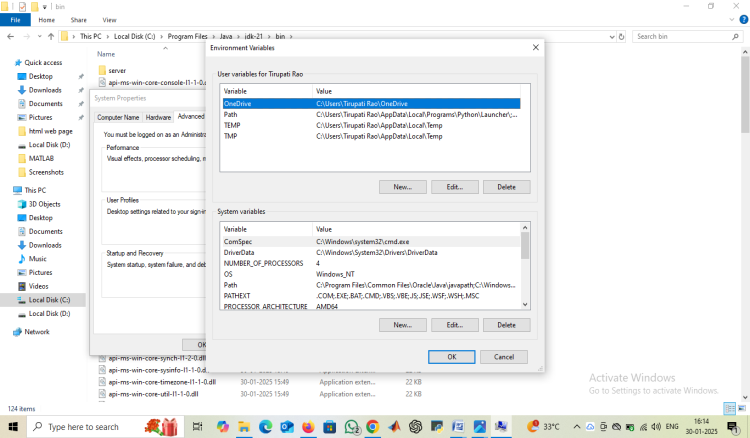
**Step-8:**

****

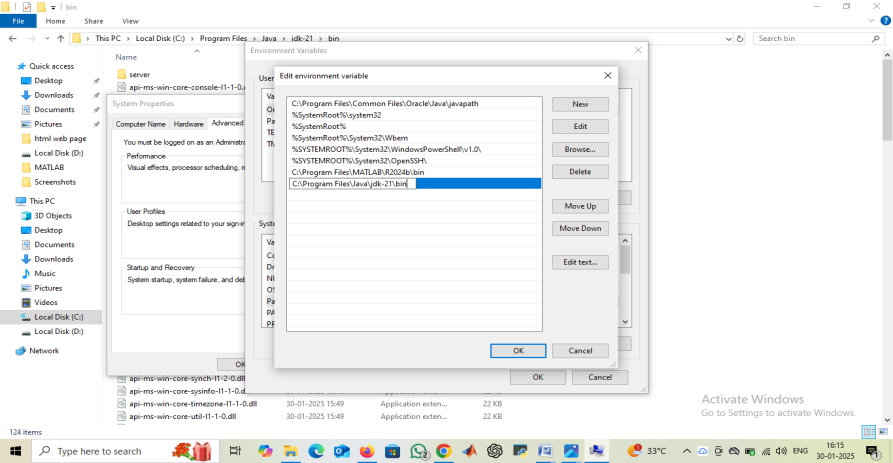
**Step-9:**

****

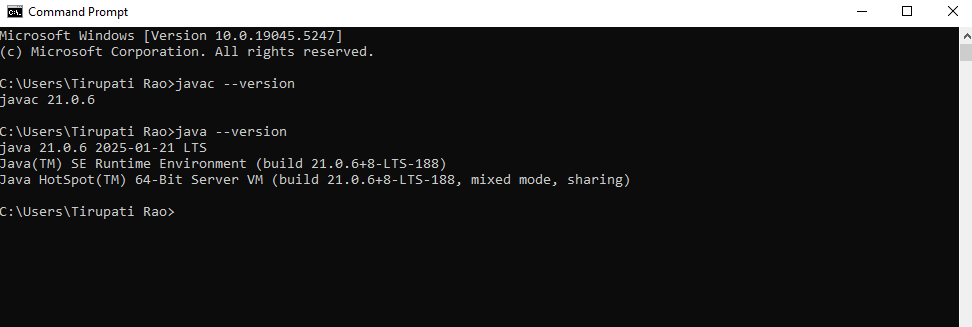
Step-10:



Step-11:



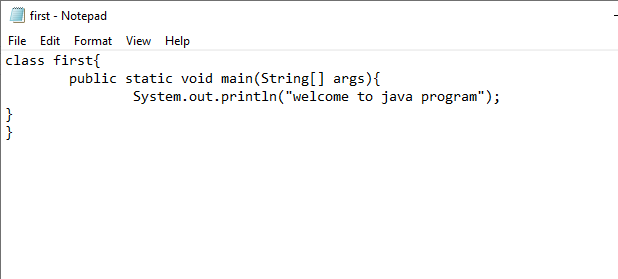
Step-13:



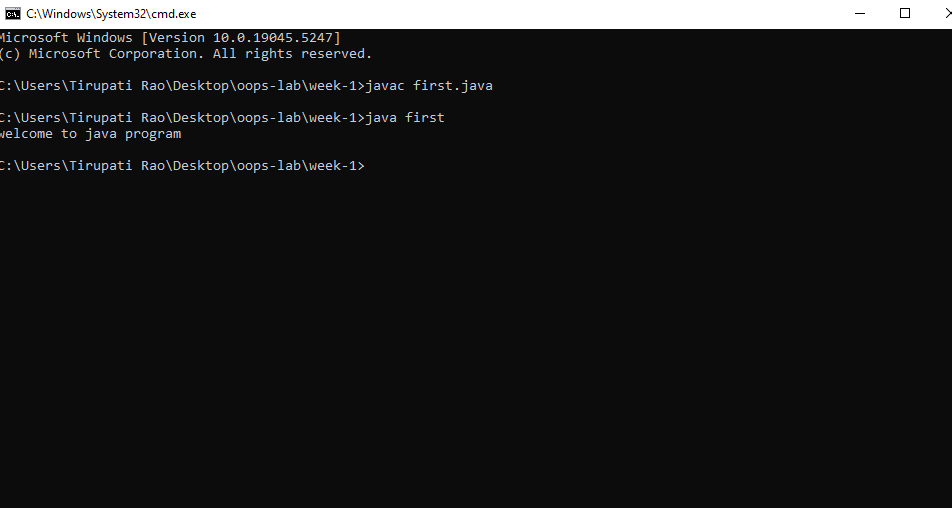
**PROGRAM-2:**

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

**Code**:



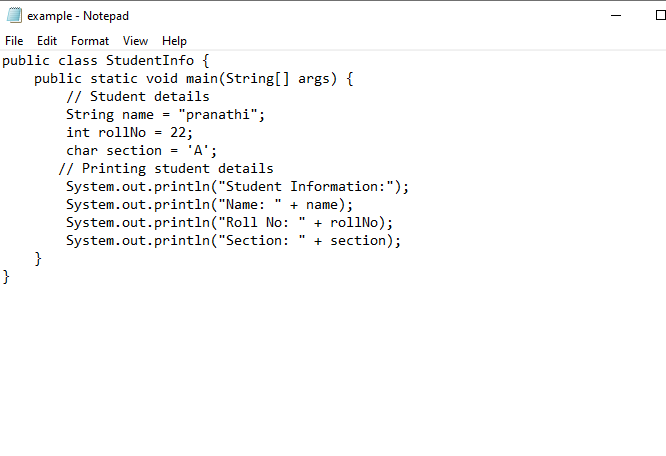
**Output:**



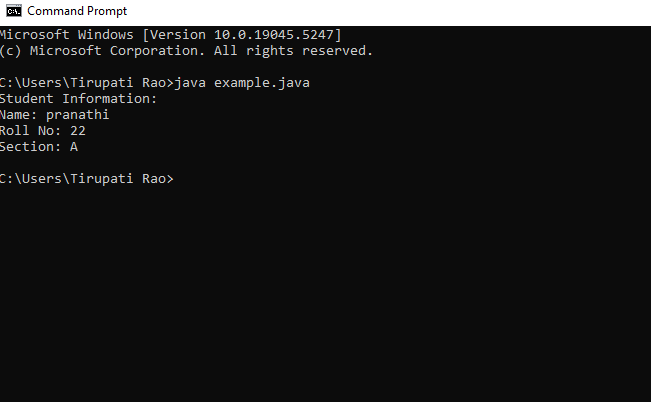
**Program-3:**

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

**CODE:**

****

**OUTPUT:**

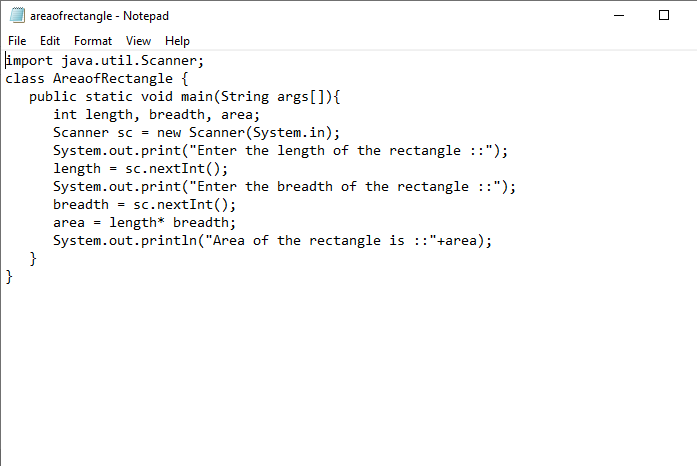


**WEEK-2**

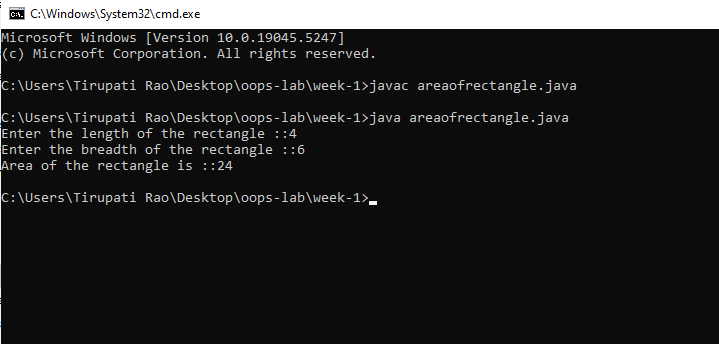
**PROGRAM-1:**

**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 .

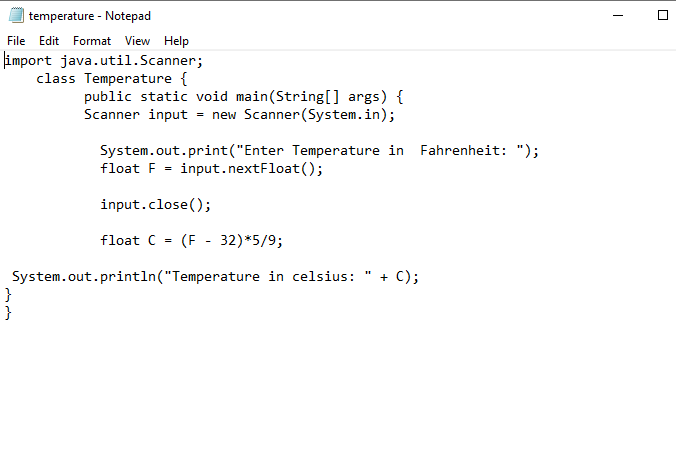
2. ”importjava.util.Scanner;”-step to import library.

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

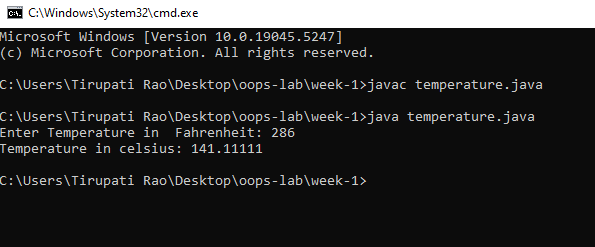
**PROGRAM-2**

**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.

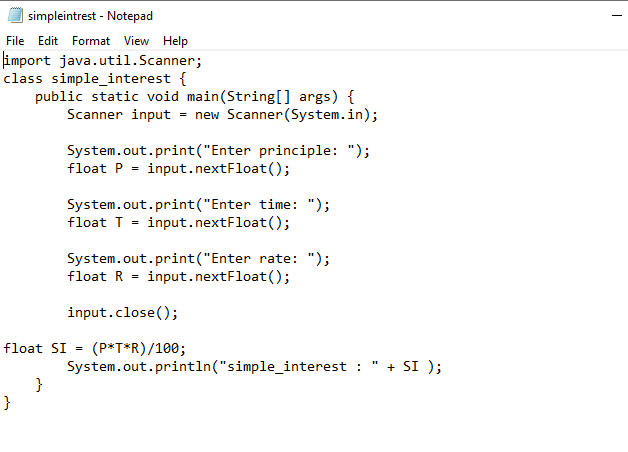
**(B).**

**CODE:**

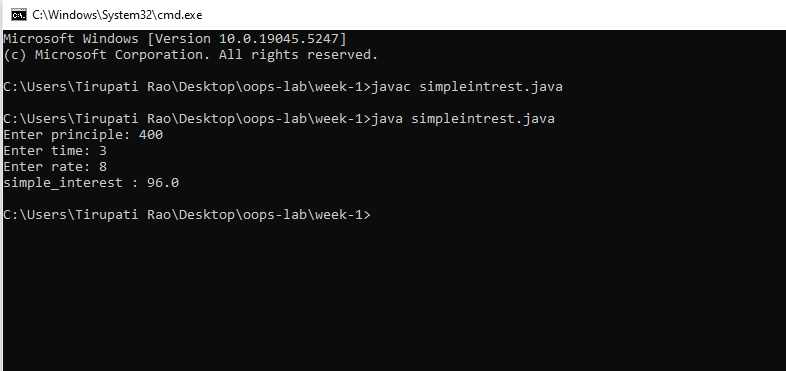
**PROGRAM-3:**

**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 |

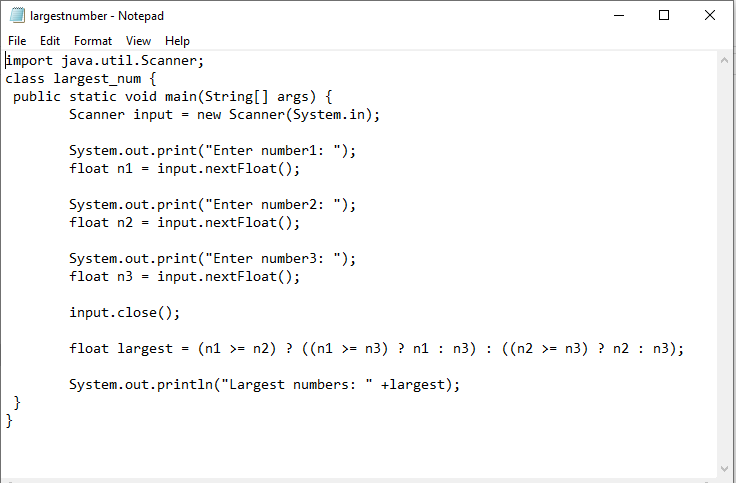
**IMPORTANT POINTS:**

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

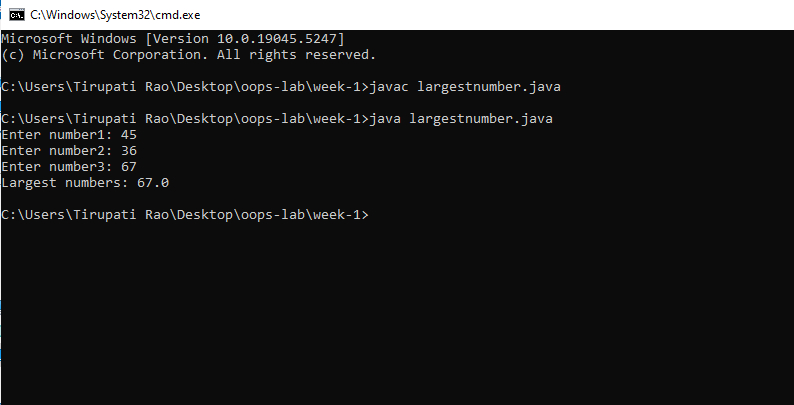
**PROGRAM-4:**

**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:**

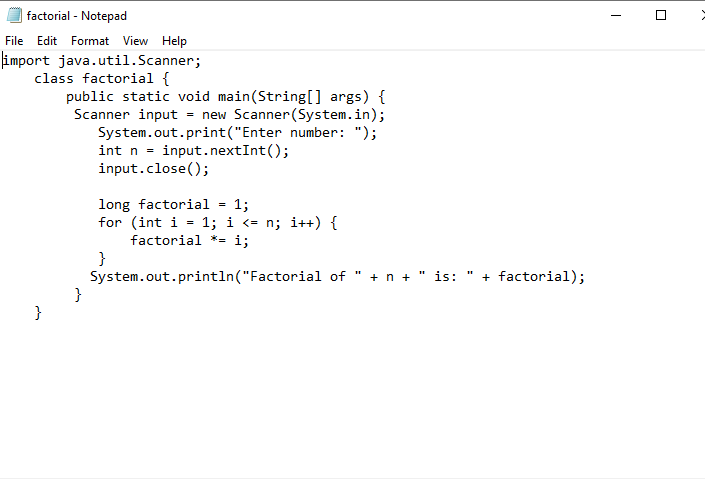
1.Ternary operators:a shorthand for the if-else statement, used to execute condition-based operations in a single line.

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

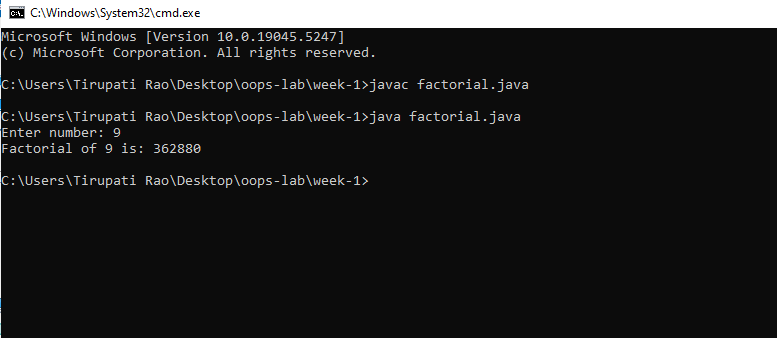
**PROGRAM-5:**

**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.

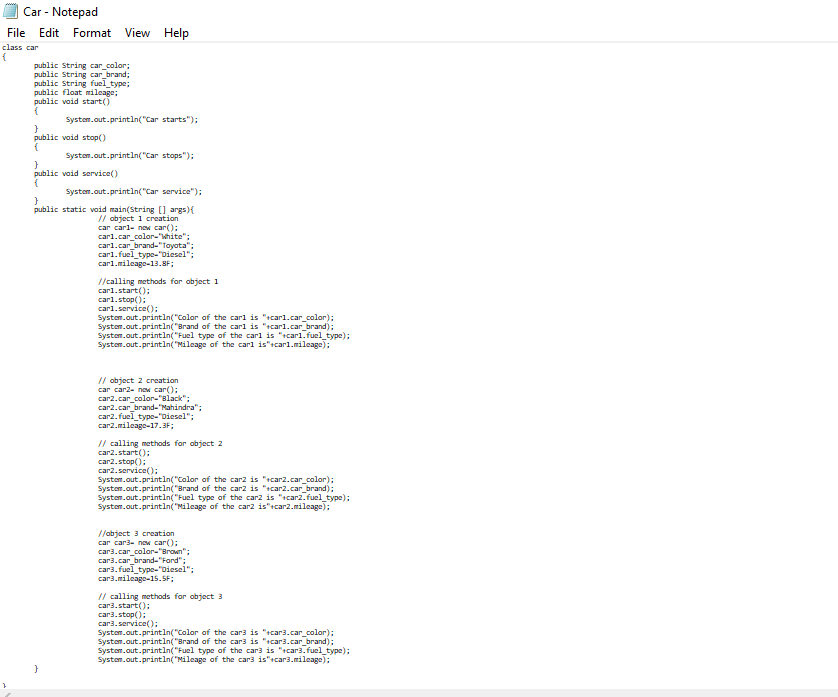
**WEEK-3**

**PROGRAM-1:**

**AIM: Create a java program with the following instructions.**

1. Create a class with name car
2. Create four attributes named car color,carbrand,fuel type and milage.
3. Create three methods named start(),stop(),service(). IV. Create three objects named car1,car2,car3.

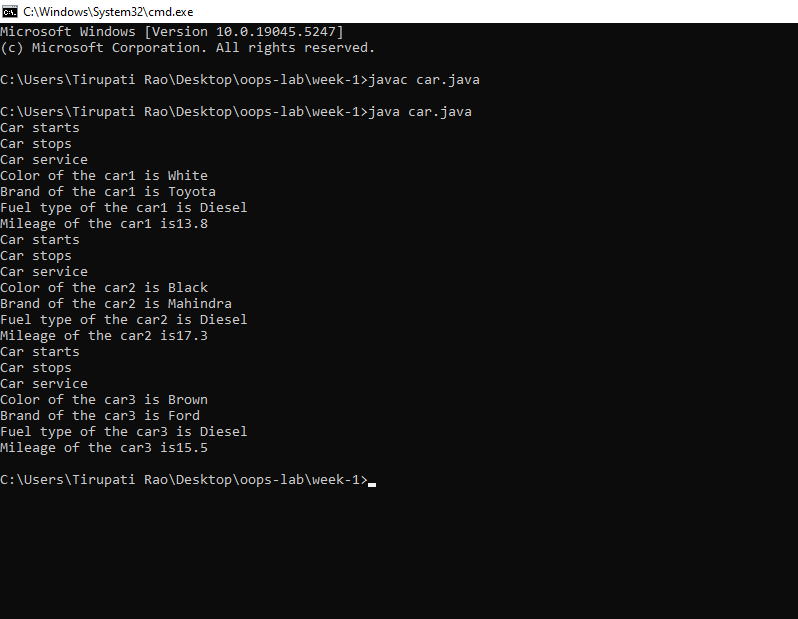
**CODE:**

****

**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.

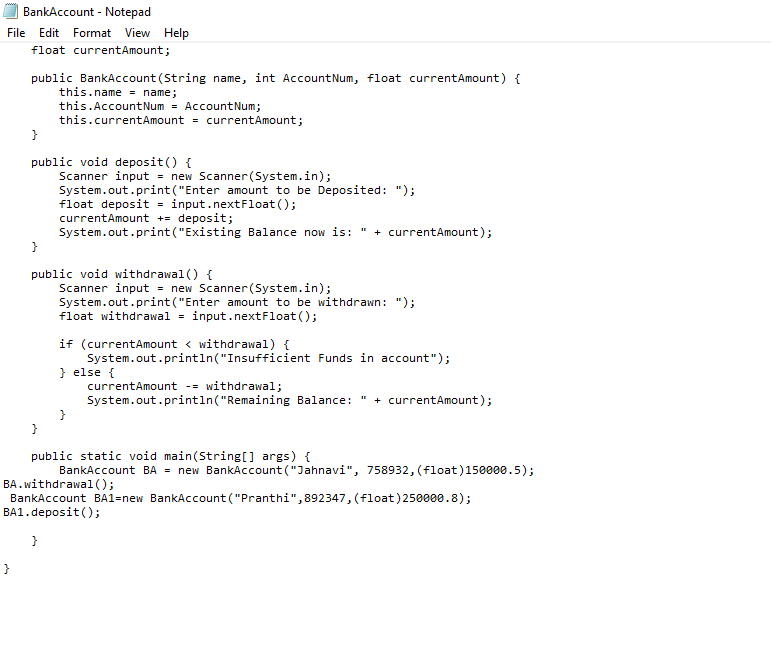
3. In Java, a method is a block of code that performs a specific task and can be invoked to execute that task. It typically consists of a method signature (name, return type, and parameters) and the body of the method, which contains the logic.

**PROGRAM-2:**

**AIM: Create a java program with the following instructions.**

1. Create a class with named bank account
2. Create two methods named withdraw(),deposit().

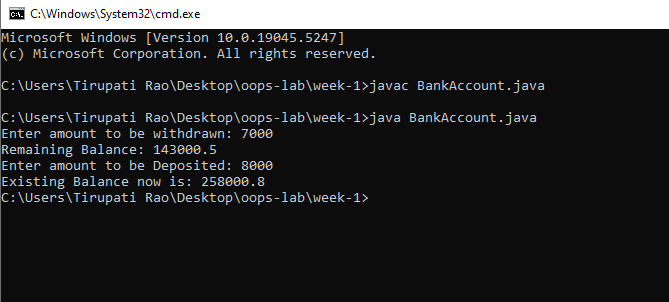
**CODE:**

****

**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:**

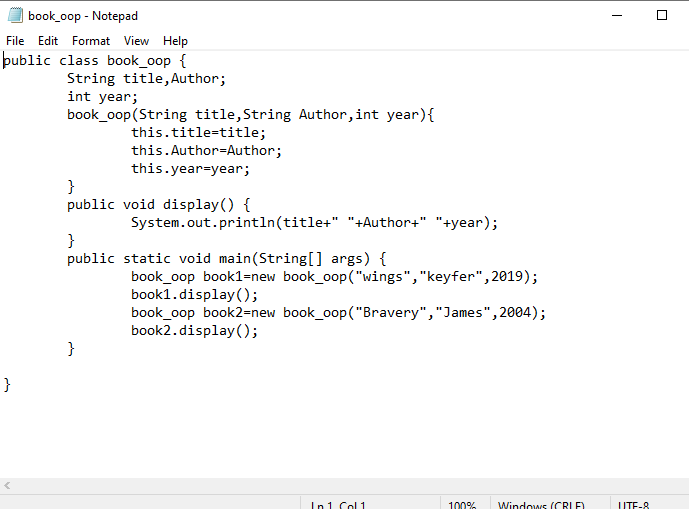
**AIM: 1) Write a java program with class named “book” the class should contain various attributes such as “title”,”author”,”year\_publication”.It should also contain a constructor with parameters which initializes “title”,”author” and “year\_publication”.**

* **Create a method which displays the details of the book i.e.”title”,”author” and**

**“year\_Publication”.**

* **Display the details of two books by creating two objects.**

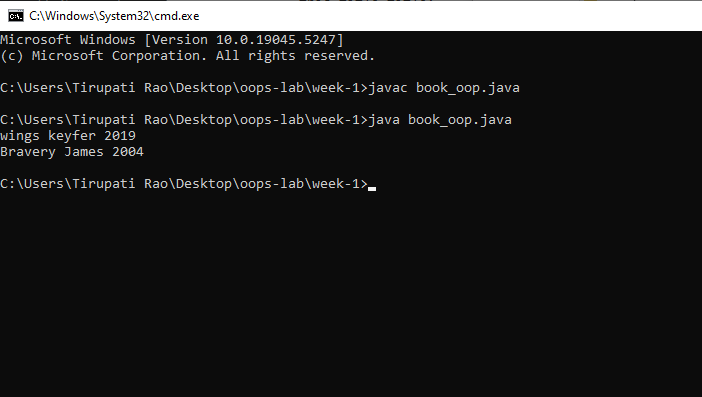
**CODE:**

****

**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. |

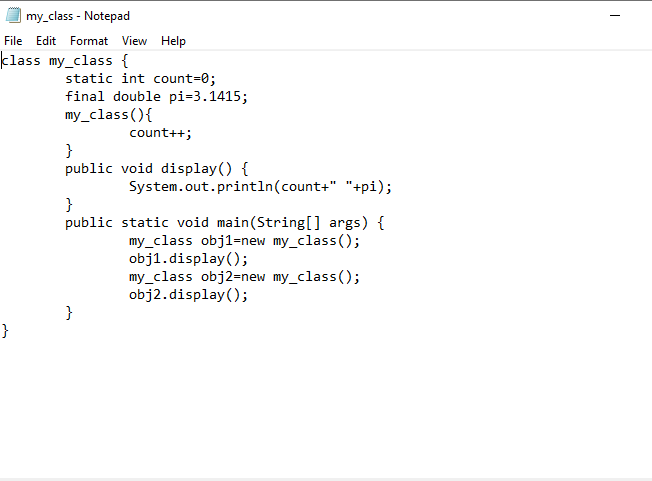
**IMPORTANT POINTS:**

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: :To create a java program with class named “My\_class” with a static variable “count” of “int” type,initialized to 0 and a constant variable “pi” of type**

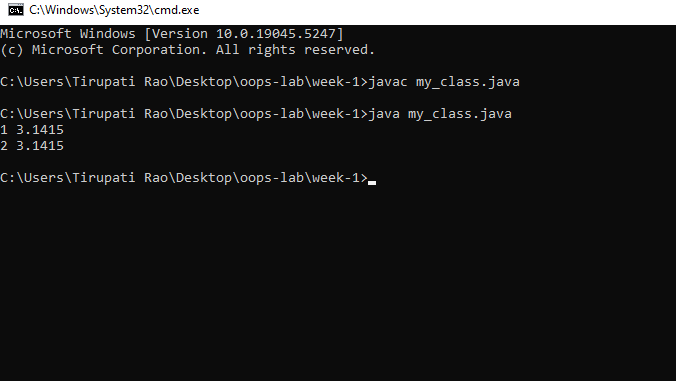
**“double” initialized to 3.1415 as attributes of that class.Define a constructor for “My\_class” that increments the count variable each time an object of “My\_class” is created finally print the final values of “count” and “pi” values.**

**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:**

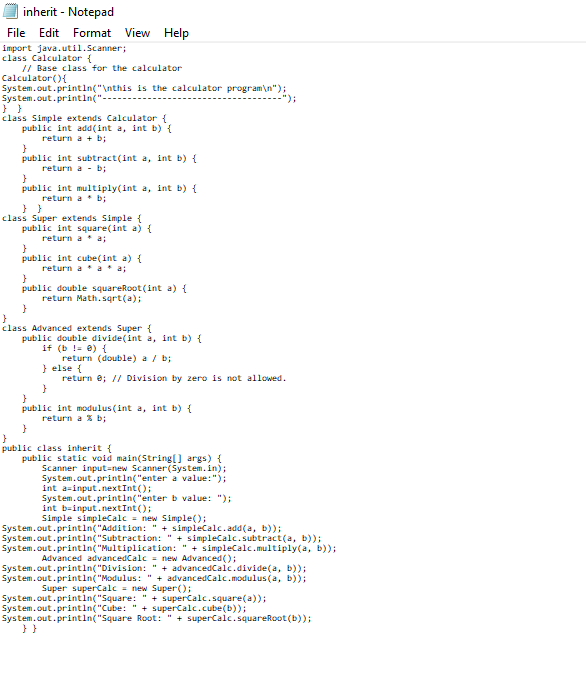
1. Java constructor is used to save the variables present in different or same class or methods.
2. In Java, the ++ operator increments a variable by 1, either as **pre-increment** (++x) or **post-increment** (x++).
3. In Java:
4. **static**: A static variable belongs to the class, not instances, meaning 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:

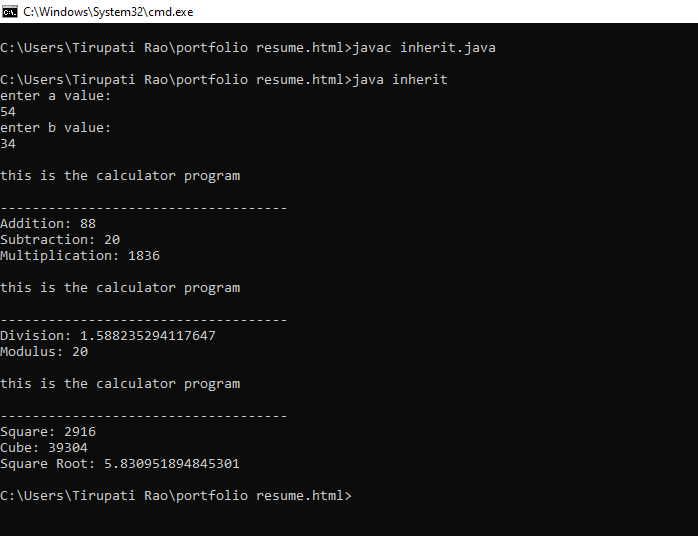
|  |
| --- |
| Calculator |
| +Calculator() |

|  |
| --- |
| 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); |

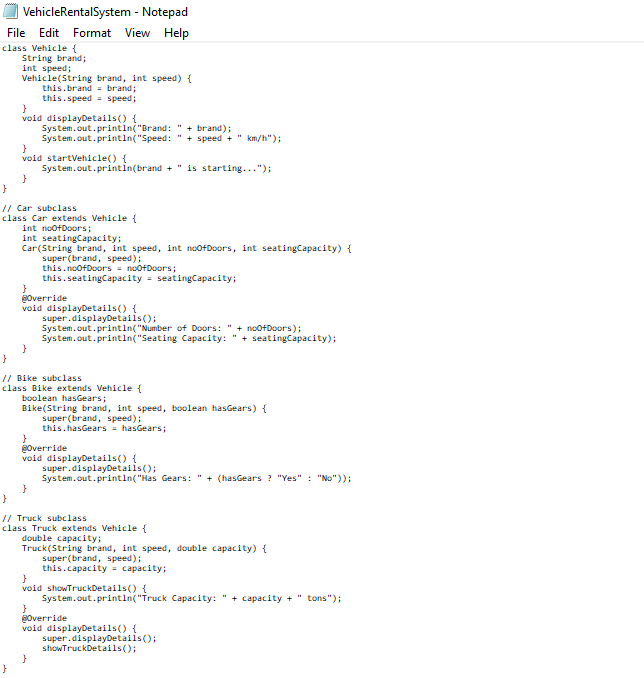
**IMPORTANT POINTS:**

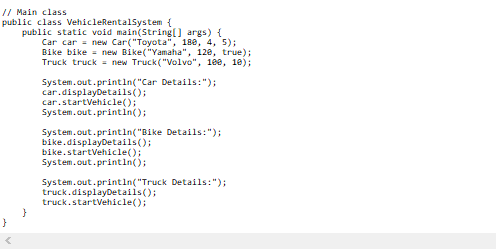
1. Multiple inheritance lets a class inherit from multiple parents, combining their features, but can cause issues like the diamond problem, resolved by MRO.
2. Math.sqrt() in Java calculates the square root of a non-negative double value and returns a double result, or NaN if the input is negative.
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**

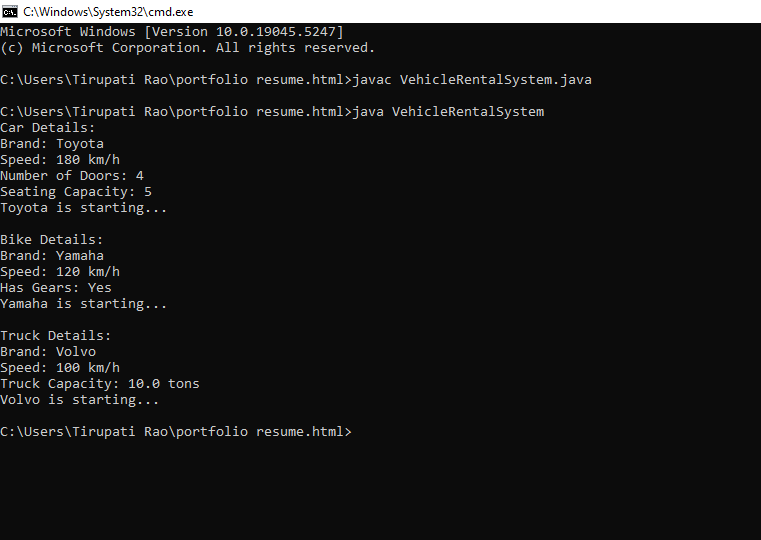
**AIM:** Create a java program of a vehicle entry company hireachical wants to develop his system that maintains information about different types of cars and bikes and they need a program to store details about each vehicle such as brand and speed

**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:**

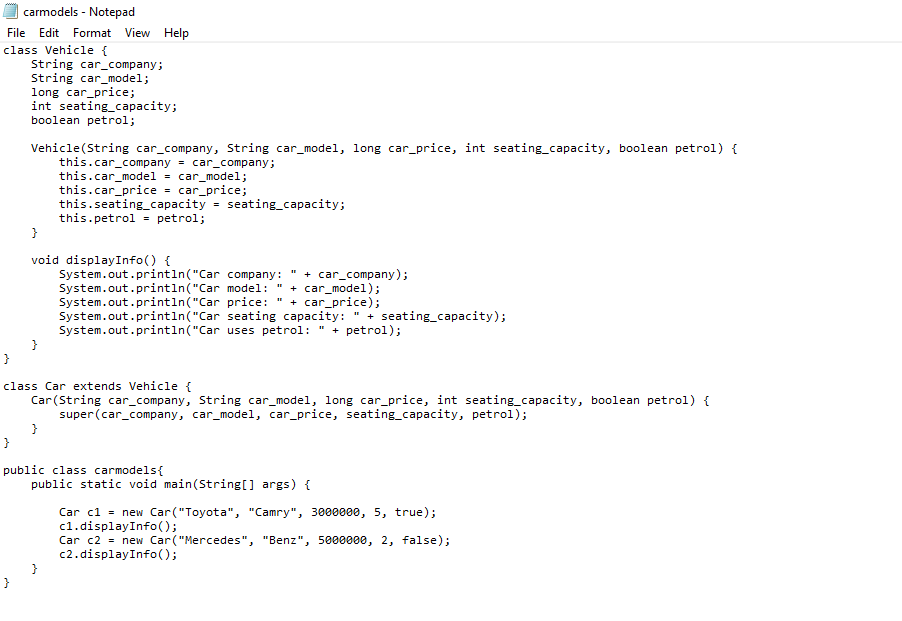
1. Hierarchical inheritance is a type of inheritance where multiple subclasses inherit from a single parent class, allowing code reuse and reducing redundancy.
2. A **constructor** is a special method in a class used to initialize new objects with default or provided values. It is automatically called 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).

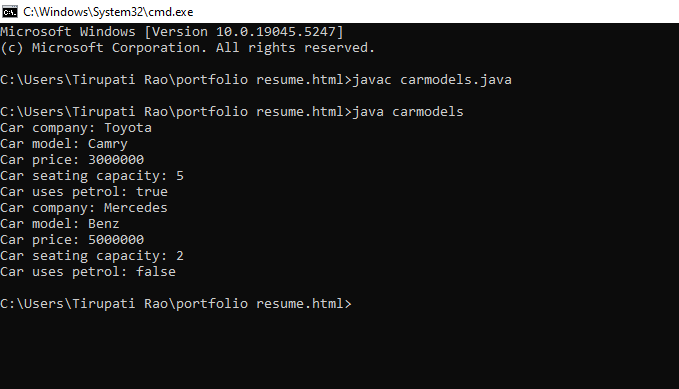
**CODE:**



**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.

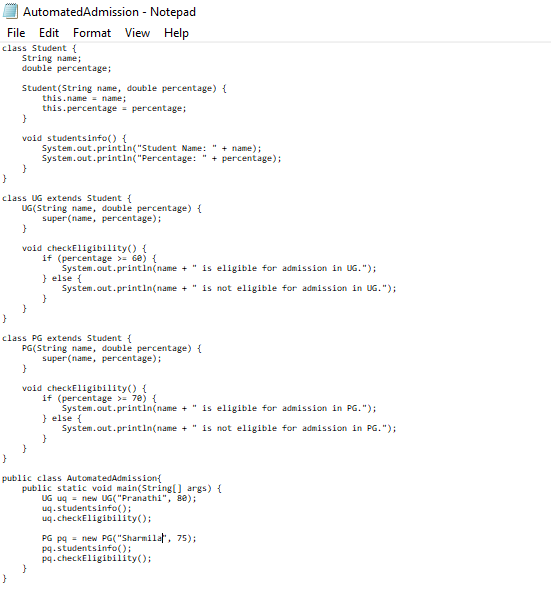
**PROGRAM-2:**

**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%.

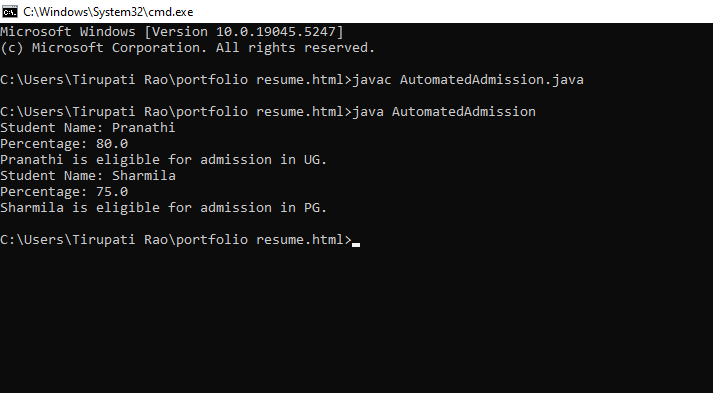
**CODE:**



**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:**

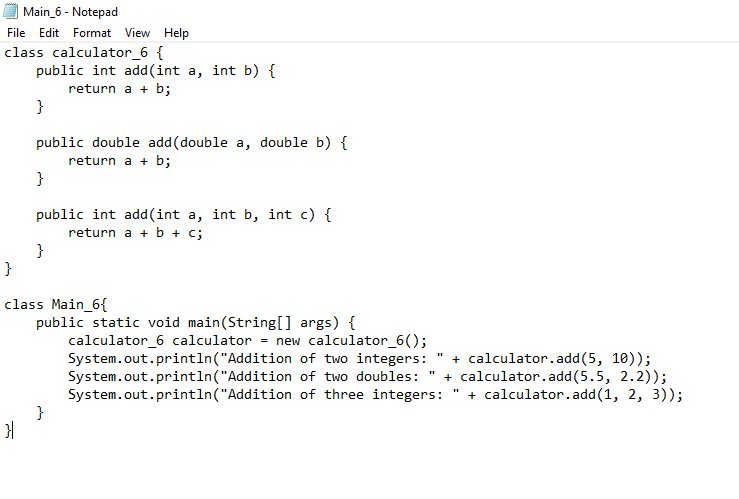
**AIM: Create a calculator class with overloaded methods to perform** addition of:

1. Add two integers

2. Add two doubles

3. Add three integers

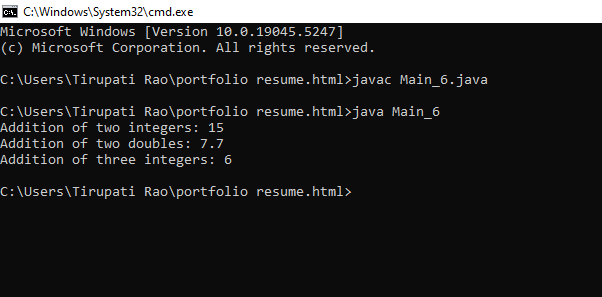
**CODE:**



**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:**

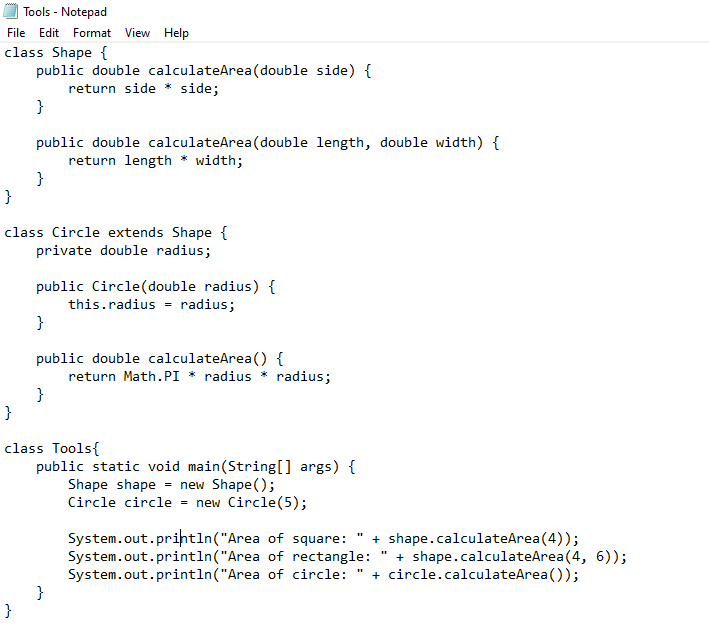
1.Method Overloading**:** The add method is overloaded with different parameter types and counts, demonstrating compile-time polymorphism.

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:**

****

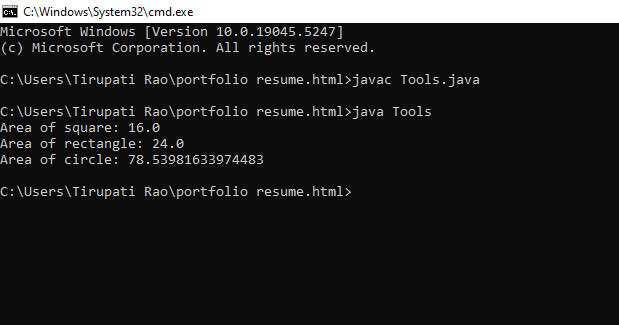
**CLASS DIAGRAM:**

|  |
| --- |
| 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.

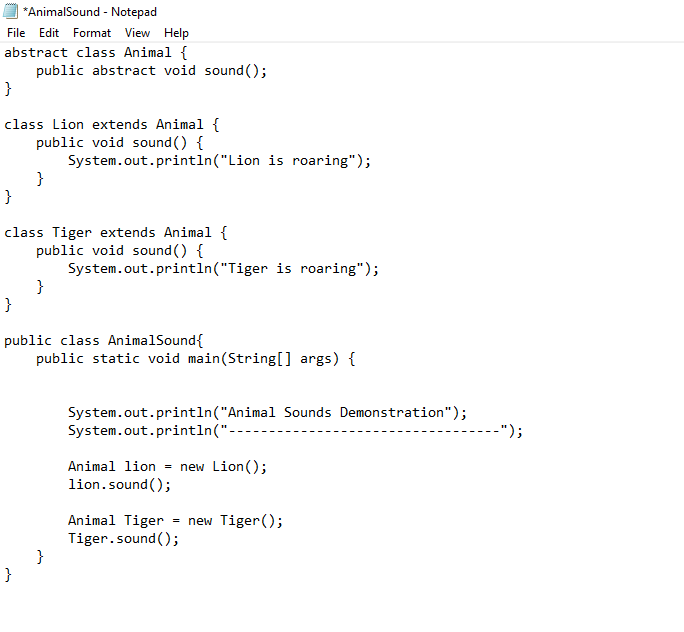
4.Polymorphism: The overridden method in Circle demonstrates runtime polymorphism.

**WEEK-7**

**PROGRAM-1:**

**AIM:** create Java program to create an abstractclass 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:**

Animal

+Sound()

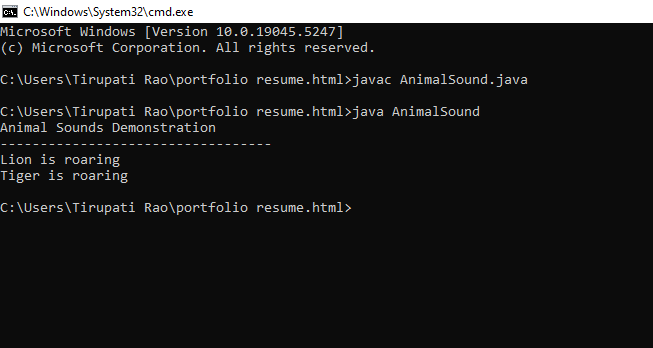
Tiger

+Sound()

Lion

+Sound()

**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. |

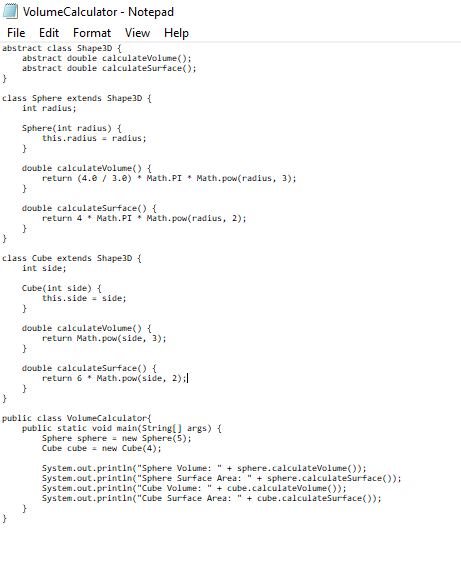
**IMPORTANT POINTS:**

* 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:**

Shape3D

+calculateVolume()

+calculateSurfaceArea()

Cube

-side: double

+calculateVolume()

+calculateSurfaceArea()

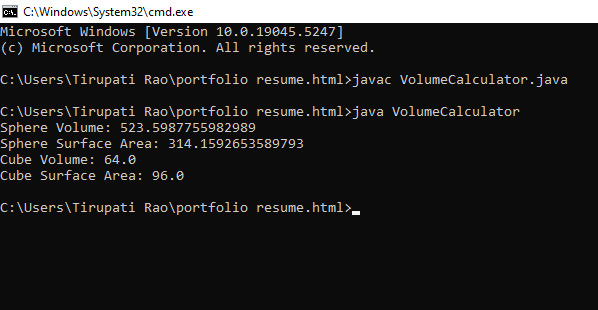
Sphere

-radius: double

+calculate Volume()

+calculateSurfaceArea()

**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

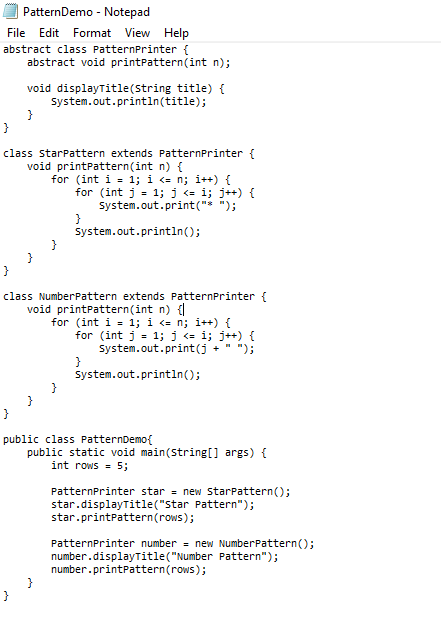
Create two subclasses:

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:**

PatternPrinter

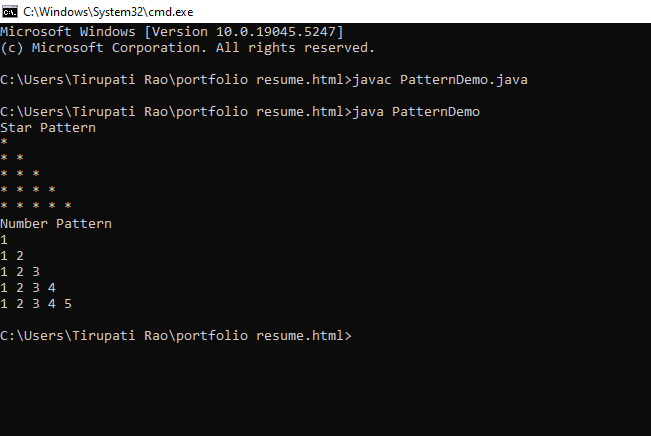
+ printPattern(int n)

+ displayTitle()

StarPattern

+ printPattern(int n)

**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:**

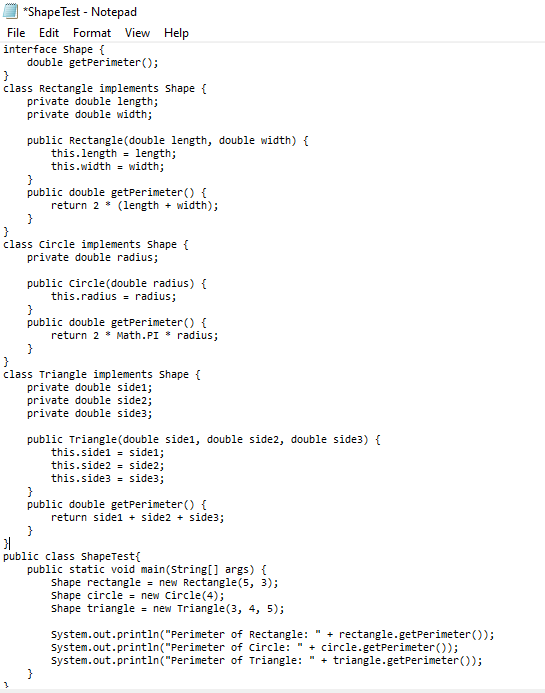
1. Abstract class PatternPrinter cannot be instantiated directly.
2. Abstract method printPattern(int n) must be implemented in all subclasses.
3. Concrete method displayTitle(String title) is reusable by both subclasses.
4. Use of inheritance: StarPattern and NumberPattern extend the abstract class.

**WEEK-8**

**PROGRAM-1:**

**AIM:**Write a javaprogram to create an interface shape with the getperimeter() method. Create three classes Rectangle,circle and triangle that implement the shape interface. Implement the getperimeter() method for each of the three classes.

**CODE:**

****

**CLASS DIAGRAM:**

Shape

+ getPerimeter()

Triangle

-side1

-side2

-side3

Rectangle

-length

-width

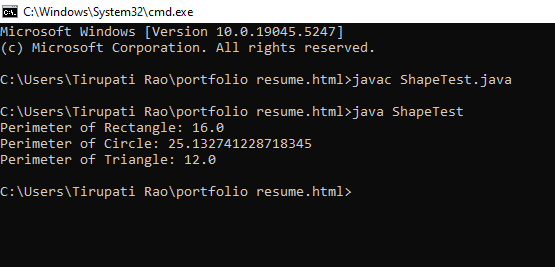
+ getPerimeter()

Circle

- radius

+ getPerimeter()

**OUTPUT:**

****

**ERROR:**

|  |  |
| --- | --- |
| CODE ERROR | ERROR RECTIFICATION |
| 1)Class name "Shapes" is inconsistently used (should be consistent capitalization) | 1) Change to consistent capitalization (either all "Shapes" or all "Shapes") |
| 2)Base class method area() returns 0 by default - better to make it abstract | 2)Consider making Shapes abstract with abstract area() method |

**IMPORTANT POINTS:**

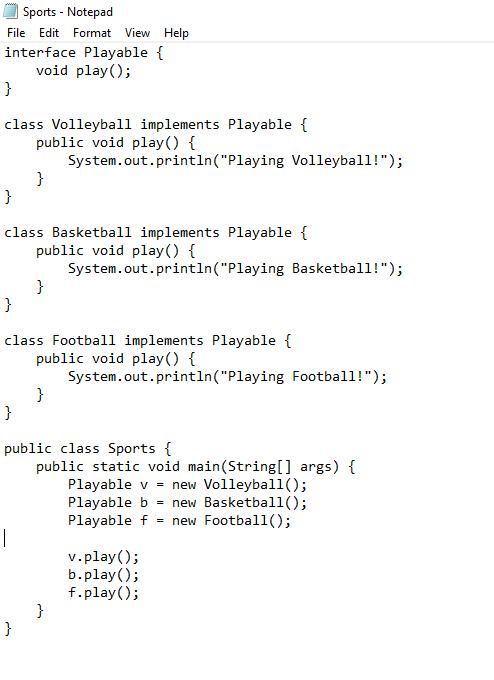
1.Inheritance Hierarchy: The Traingle, Circle and Rectangle classes all inherit from the base Shapes class (note: class name is misspelled as "Shapes" in some places and "Shapes" in others).

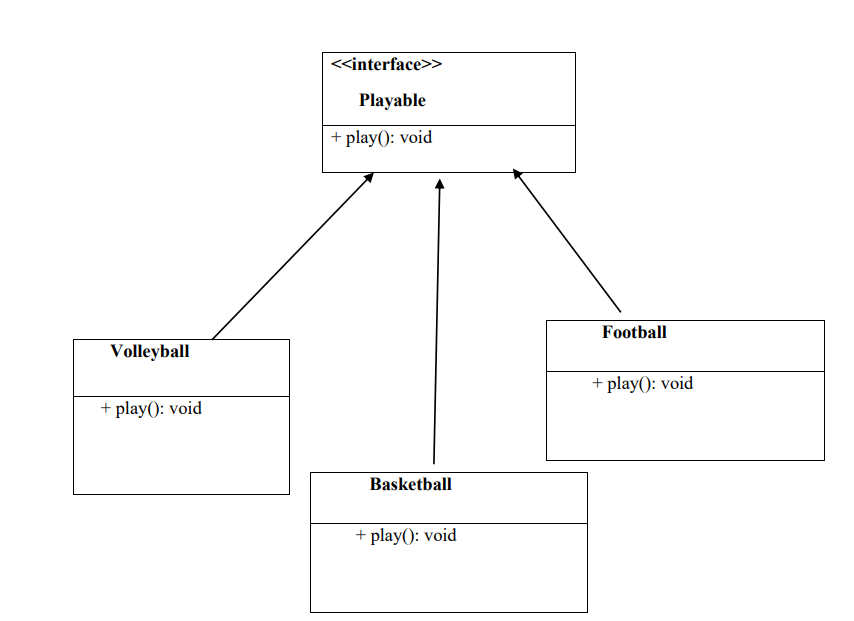
2.Polymorphism: Each subclass overrides the area() method to provide its own implementation, demonstrating polymorphic behavior. 3.Encapsulation: All shape classes properly encapsulate their attributes (base, height, radius, length, width) as private fields. 4.Method Overriding: The area() method is overridden in each subclass with the appropriate calculation formula for that shape. 5.Main Class: The ShapeArea class demonstrates the use of these shapes by creating instances and calling their area() methods.

**PROGRAM-2:**

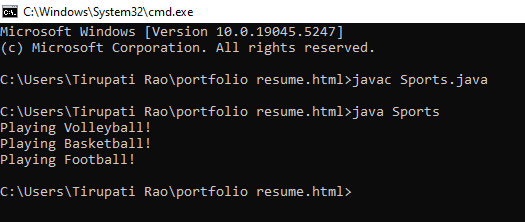
**AIM:** write a java program to create an interface playable with a method play() that takes no arguments and returns void. Create three classes Football,Volley ball and Basket ball that implement the playable interface and override the playable method to play the respective sports.

**CODE:**

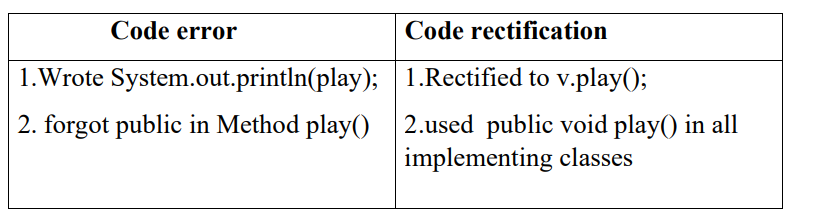


**CLASS DIAGRAM:  
**

**OUTPUT:**

****

**ERRORS:**

****

**IMPORTANT POINTS:**

**1**. Playable is an interface, so it only has abstract methods (by default).

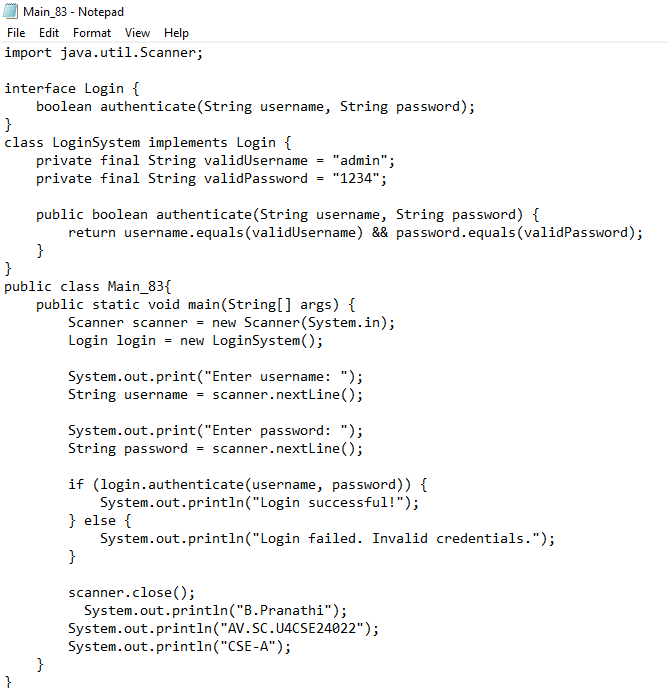
**2.** All classes implement the Playable interface.

**3.** Each class overrides the play() method with its own message.

**PROGRAM-3:**

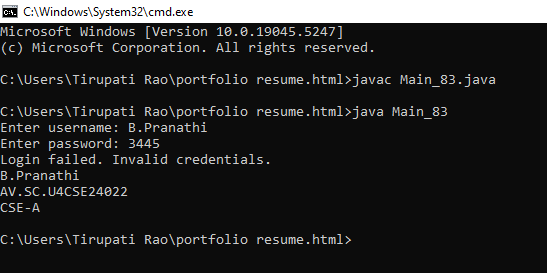
**AIM:** Write a java program to implement a login system using interfaces.

**CODE:**

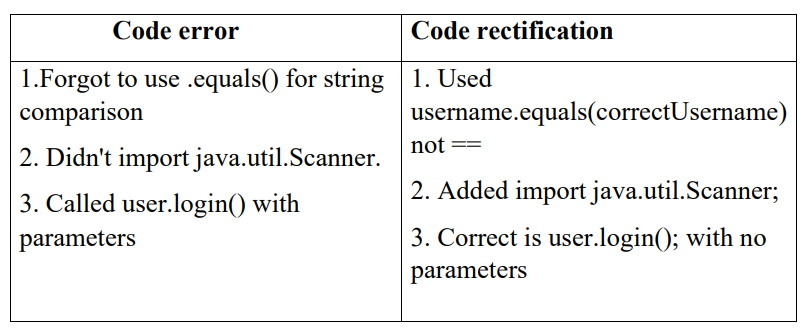


**CLASS DIAGRAM:**

**OUTPUT:**

****

**ERROR:**

****

**IMPORTANT POINTS:**

1.Interface Implementation: University portal correctly implements Login System interface uses override annotation for the login() method 2. Authentication Logic: Hardcoded credentials: id=”Students123”,password=”pass02” Returns Boolean and prints appropriate message. 3.Main Method: Demonstrate both successful and failed login attempts includes student information print statement.

**WEEK-9**

**PROGRAM-1:**

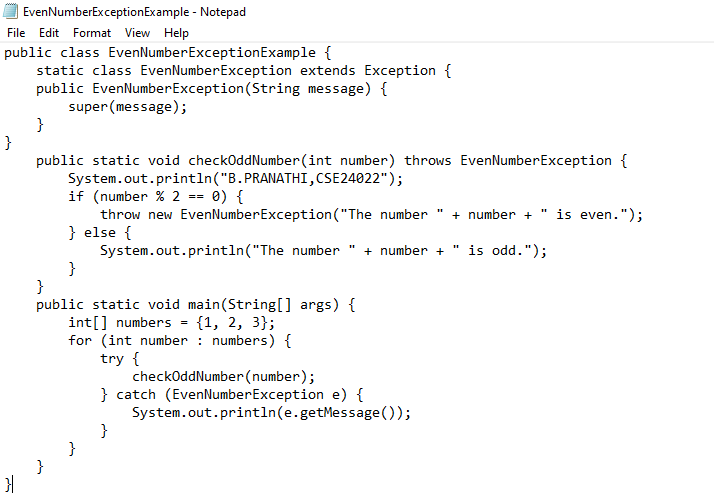
**AIM:**Write a java program to create a method that takes integers as parameters and throws an exception if the number is even.

**CLASS DIAGRAM:**

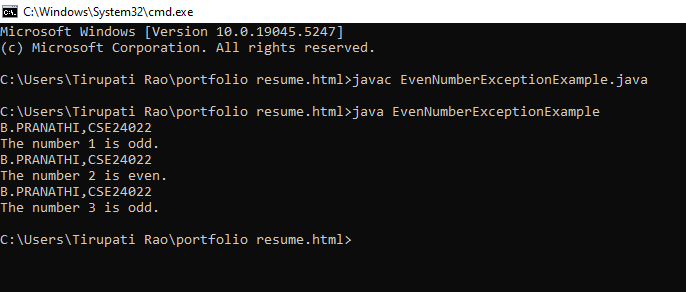
|  |
| --- |
| **EvenNumberExceptionDemo** |
| +checkoddNumber(int number): void+main(String[]args):void |

|  |
| --- |
| **EvenNumberException**  (extends Exception) |
| + EvenNumberException() |

**CODE:**



**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. Unhandeled exception type EvennumberException. 2. Syntax errors 3. Compilation error. | 1. If you checkoddNumber() without using try-catch or without declaring throws. 2. If missing curly barces or wrong method syntax. 3. If constructor of EvenNumberException is missing or incorrectly defined. |

**IMPORTANT POINTS:**

1) Created a custom exception by extending the Exception class.

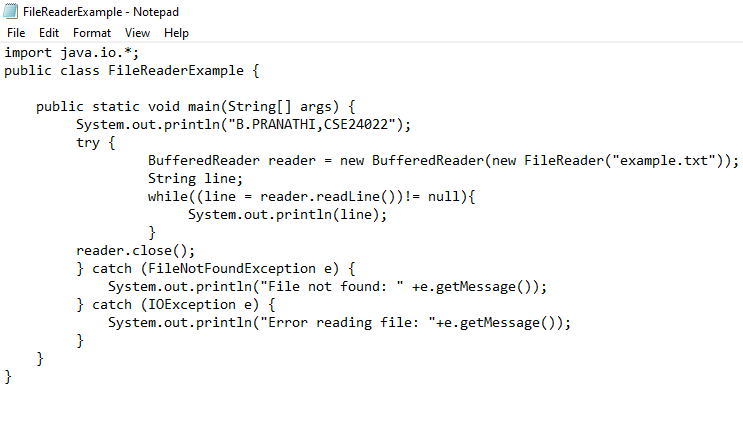
2) Used throw keyword to manually throw the custom exception if the number is even.

3)Handled the exception using a try-catch block inside main() method.

**PROGRAM-2**

**AIM:** Write a java program to create a method that reads a file and throws an exception if the file is not found.

**CODE:**

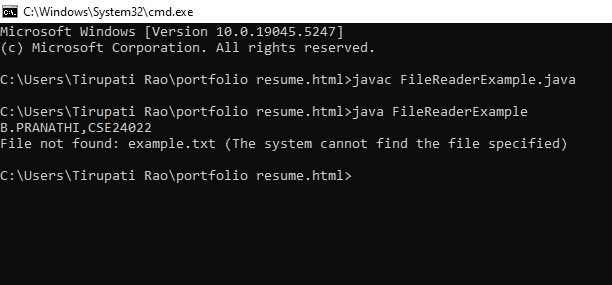


**CLASS DIAGRAM:**

|  |
| --- |
| **FileReadExample** |
| + main(String[] args) : void |

|  |
| --- |
| **Uses** |
| -BuefferdReader  -FileReader  -FileNotFound  -IOException |

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. File Not Found 2. IOException 3. Syntax Error | 1. Occurs if the specified fiole path is wrong or file does not exist. 2. Occurs while reading file if an input/output error happens. 3. If missing semicolon; wronf try-catch syntax |

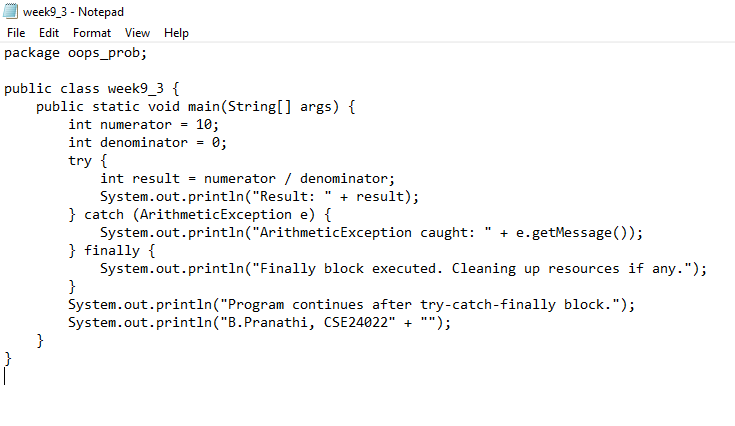
**IMPORTANT POINTS:**

1. Used BufferedReader and FileReader to read text files.
2. FileNotFoundException occurs if the file is missing.
3. IOException occurs for input/output errors during file reading.
4. try-catch block is used for proper exception handling.

**PROGRAM-3**

**AIM:** Write a java program to handle arthimatic exception using try, catch and finally.

**CODE:**

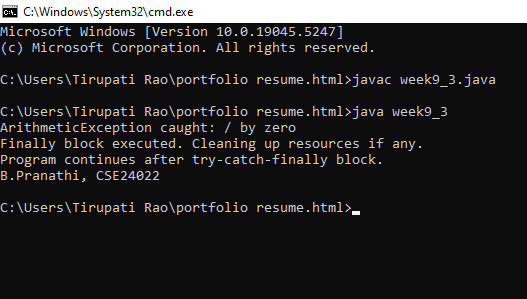
****

**CLASS DIAGRAM:**

|  |
| --- |
| **FileReadExample** |
| +main(String[]args): void |
|  |

|  |
| --- |
| **Uses** |
| **-**BuefferdReader  -FileReader  -FileNotFound  -IOException |

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. File not found 2. IOException 3. Syntax error | 1. Occurs if the specified file path is wrong or file does not exist 2. Occurs while reading file if an input/output error happens 3. If missing semicolon ; wrong try-catch block syntax |

**IMPORTANT POINTS:**

1. Used BufferedReader and FileReader to read text files.
2. FileNotFoundException occurs if the file is missing.
3. IOException occurs for input/output errors during file reading.
4. try-catch block is used for proper exception handling.
5. Always close the reader after reading the file (reader.close()).

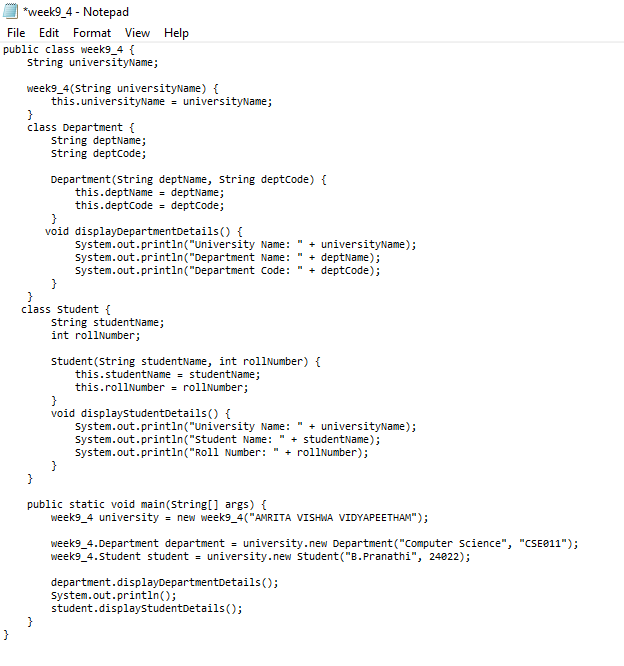
**PROGRAM-4**

**AIM:** Write a program to stimulate a university system using inner classes.

* Create an outer class named university with a variable universityName.
* Inside it, define two non-static inner classes.

1. Department- With variables like deptName, deptCode and a method to display department details.
2. Student- With variables like studentName, rollNumber and a method to display details.
3. Create an object for each class and call their methods to display their details along with their universityName.

**CODE:**



**CLASS DIAGRAM:**

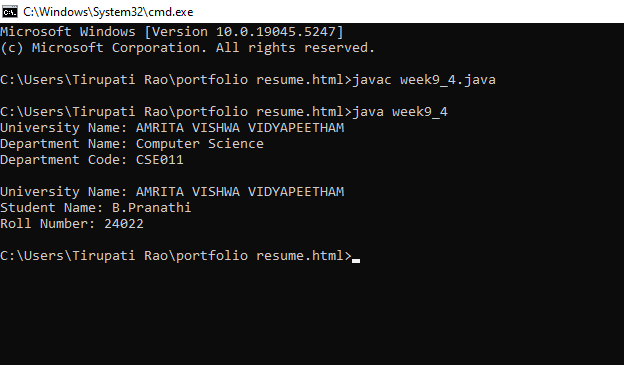
|  |
| --- |
| **University** |
| -universityName: String  + University(String name) |

|  |
| --- |
| **Innerclass** |
| - Department - deptName: String- deptCode: String  + displayDepartment(): void |

|  |
| --- |
| - Student  - studentName: String  - rollNumber: int + displayStudent(): void |

|  |
| --- |
| + main(String[] args): void |

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. Syntax Error 2. Compilation Error 3. Runtime Error | 1.Wrong object creation for inner class  2.Accessing outerclass members wrongly  3.Null pointerException if outer object missing |

**IMPORTANT POINTS:**

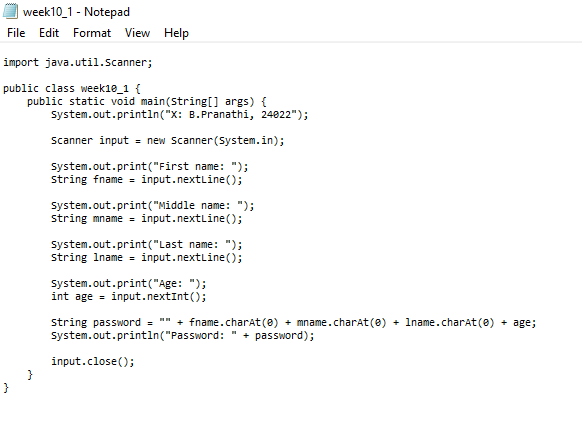
1. Demonstrate inner class usage.
2. Inner classes access outer class members easily.
3. Separate objects for Department and Student.Good example of encapsulation

**WEEK-10**

**PROGRAM-1:**

**AIM:**Write a java program to generate a password for a student using his/her initials and age. The password displayed should be the string consists of first character of first name, middle name, last name with age.

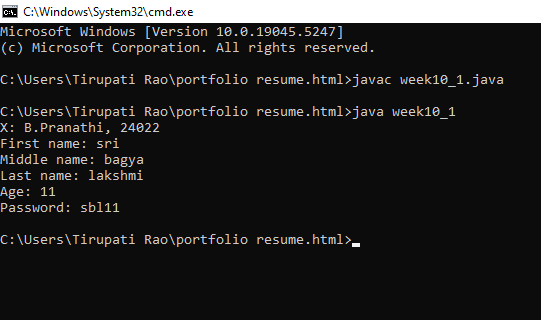
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| **Password** |
| +first name: String  +middle name: String  +last name: String  +age: int |
| +password(): String |
| +main(String[]args): void |

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. Identifier expected after the token | Give a suitable identifier that will give the output |

**IMPORTANT POINTS:**

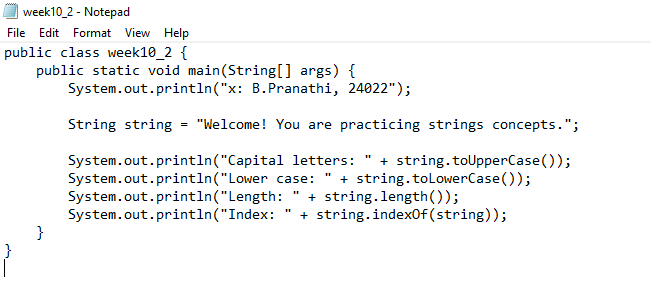
1. The program takes the first, middle, lastnames and age as input.
2. It creates a password using the first letter of each name plus the age.
3. The logic is organized using a class and method.
4. It uses the Scanner class to read input from the user.
5. The program follows object-oriented principles with clean structure and reusability.

**PROGRAM-2**

**AIM:** Design and implement a java program that will do the folleing operations to this string “Welcome! You are practicing String concepts”.

* Convert all alphabets to capital letters and print out the result.
* Convert all alphabets to lowercase letters and print out the result.
* Print out the length of the string.
* Print out the index of the concept.

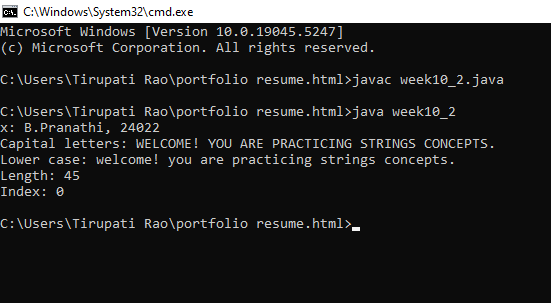
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| **Strings** |
| +string: public  +string.uppercase()  +string.lowercase()  +string.length()  +string.index() |
| +void main(String[]args): void |

**OUTPUT:**



**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. ToUpper is undefined for the type string. | * + - 1. Type toUpperCase() instead of to uppercase() |

**IMPORTANT POINTS:**

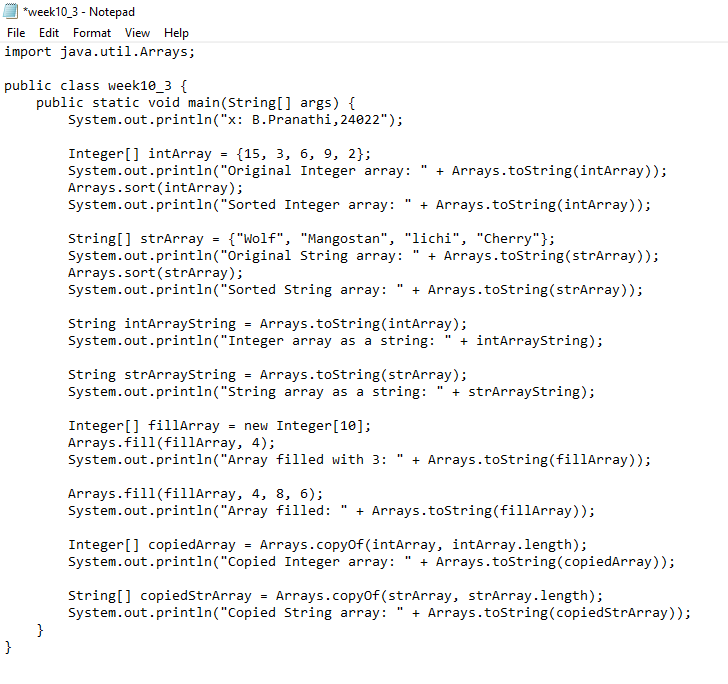
1. The program defines a string message and displays it.
2. It uses touppercase() and tolowercase() to change the case of the string.
3. Length() is used to find the length of the string.
4. Indexof() returns 0, because the string is searching for itself at index().

**PROGRAM-3:**

**AIM**: Implement a java program using the below array methods.

* Sorting the elements (numbers & strings) of an array.
* Convert the array elements into String.
* Fill the part of an array.
* Copy the elementsof one array into the other.

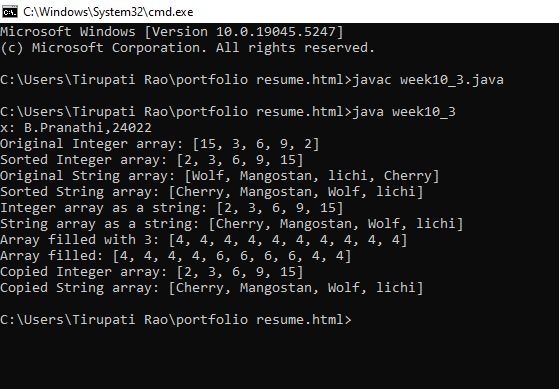
**CODE:**



**CLASS DIAGRAM:**

|  |
| --- |
| **Arrays** |
| +sort(arr: T[]): void  +toString(arr: T[]): String  +fill(arr: T[], value: T): void  +fill(arr: T[], fromindex: int, toindex: int, value: T): void  +copyof(arr: T[], newLength: int): T[] |

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. 1. The method toString() in the object is not applicable for the arguments(integer[]) | * + - 1. Declare Array instead of array. |

**IMPORTANT POINTS:**

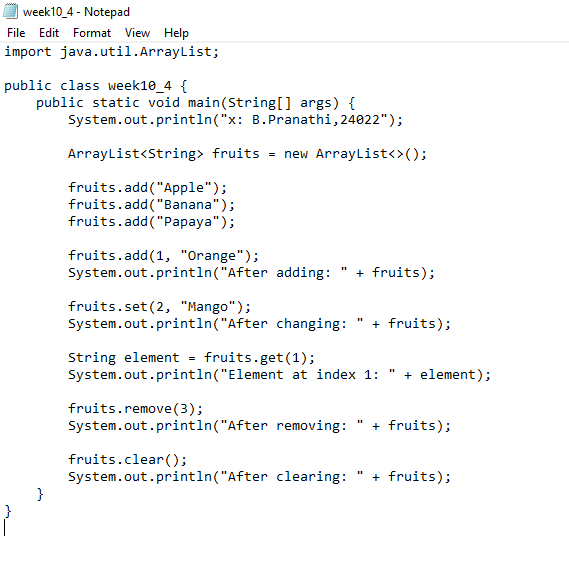
1. The code demonstrates sorting arrays to both integers and strings using Arrays.sort().
2. It converts arrays to string representations with Arrays.toString().
3. It uses Arrays.fill() to fill entire arrays or parts of arrays with specific values.
4. The code copies arrays using Arrays.copyof() to create new arrays with the same elements.

**PROGRAM-4:**

**AIM:** Implement a java program using the below Array list method.

* Insert an element at particular index in the array list.
* Modify an element in the array list.
* Access an element from the array list.
* Remove an element from the Array list.
* Clear the elements from the array list.

**CODE:**

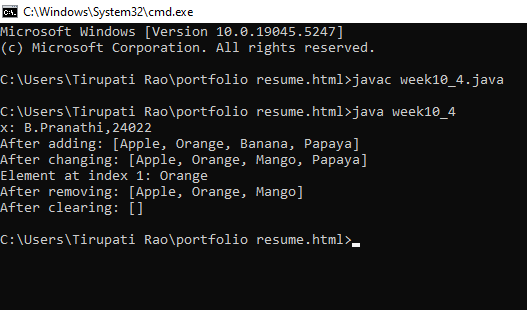
****

**CLASS DIAGRAM:**

|  |
| --- |
| **ArrayListExamaple** |
| +fruits: ArrayList<String>  +main(args: String[]): void |

|  |
| --- |
| **ArrayList<T>** |
| +add(E e): Boolean  +add(int index, E element): void  +set(int index, E element): E  +get(int index): E  +remove(int index): E  +clear(): void |

**OUTPUT:**

****

**ERRORS:**

|  |  |
| --- | --- |
| **Code Error** | **Code Rectification** |
| 1. 1. Array list not defined 2. 2. Exception main.java.lang.error; | * + - 1. Define the array list class.   2. Insert() to computer class instance creation expression. |

**IMPORTANT POINTS:**

1. The program demonstrates how to create and manipulate an arraylist in java.
2. Elements are instead at specific positons using the add(index, element) method.
3. The set(index, element) method is used to modify existing elements.
4. The entire listcan be cleared using the clear() method to reamove all elements.