23CSE111

OBJECT ORIENTED PROGRAMMING

**DOCUMENT**

****

## Department of computer science Engineering

## Amrita School of Engineering

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

### Name: Davis J.Johney

**Verified By : Roll No: CSEA-24041**

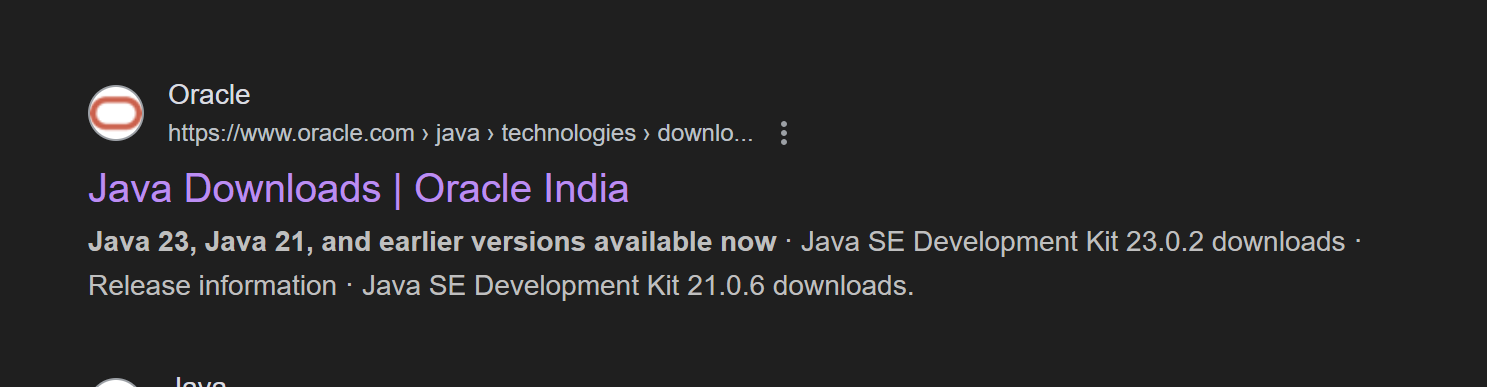
|  |  |  |
| --- | --- | --- |
|  |  |  |

**WEEK - 1**

1. **Write the steps to download and install Java.**

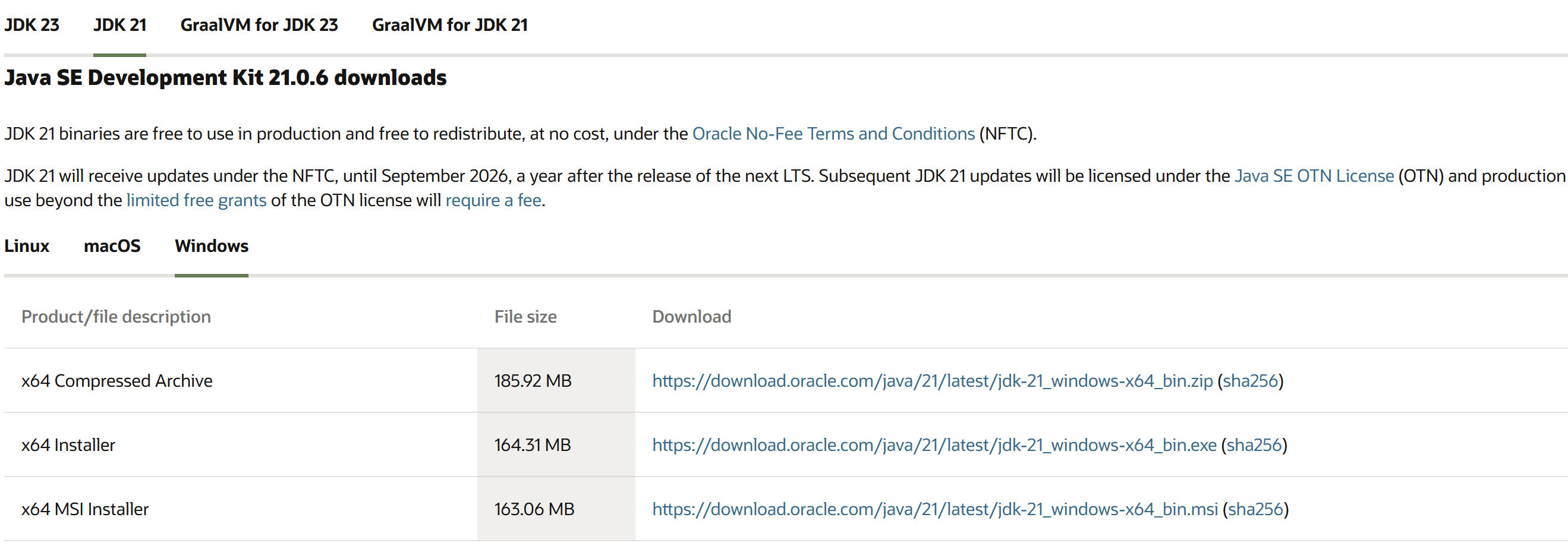
**Aim : Download and Install Java Software**

**Step – 1 : Visit any web browser and search for java download. Select the official Oracle website.**

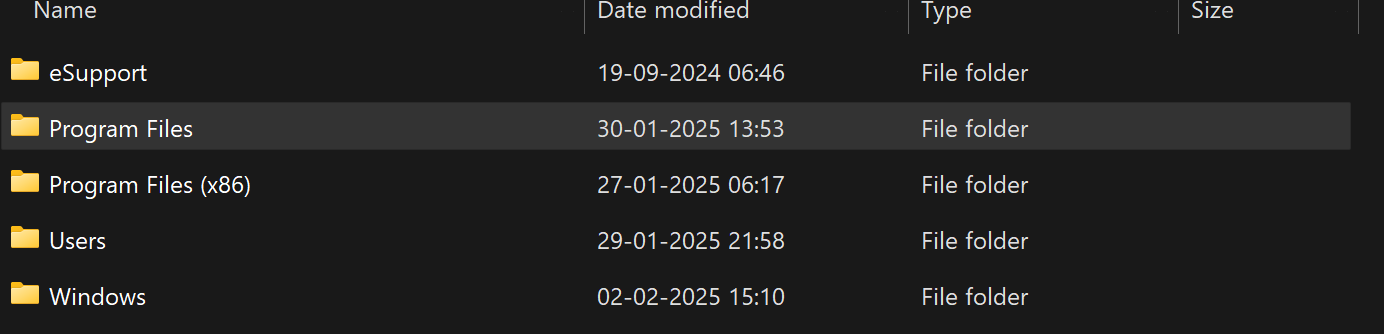


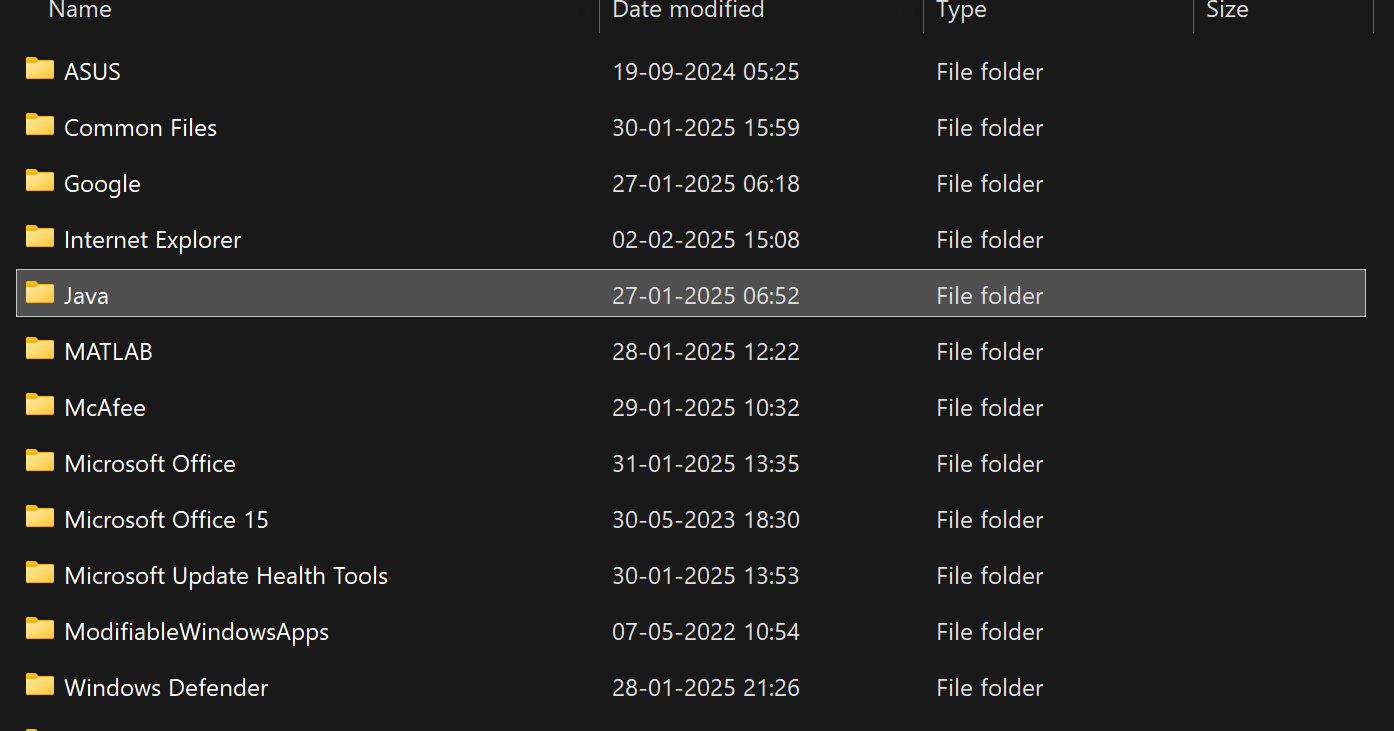
**Step – 2 : Open Oracle website and select the LTS “JDK 21 “ for Windows and select “X64 Installer” and**

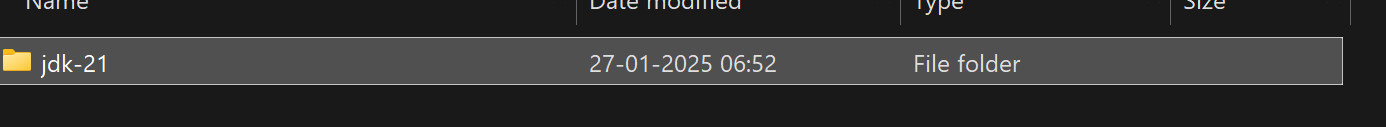
**download it.**



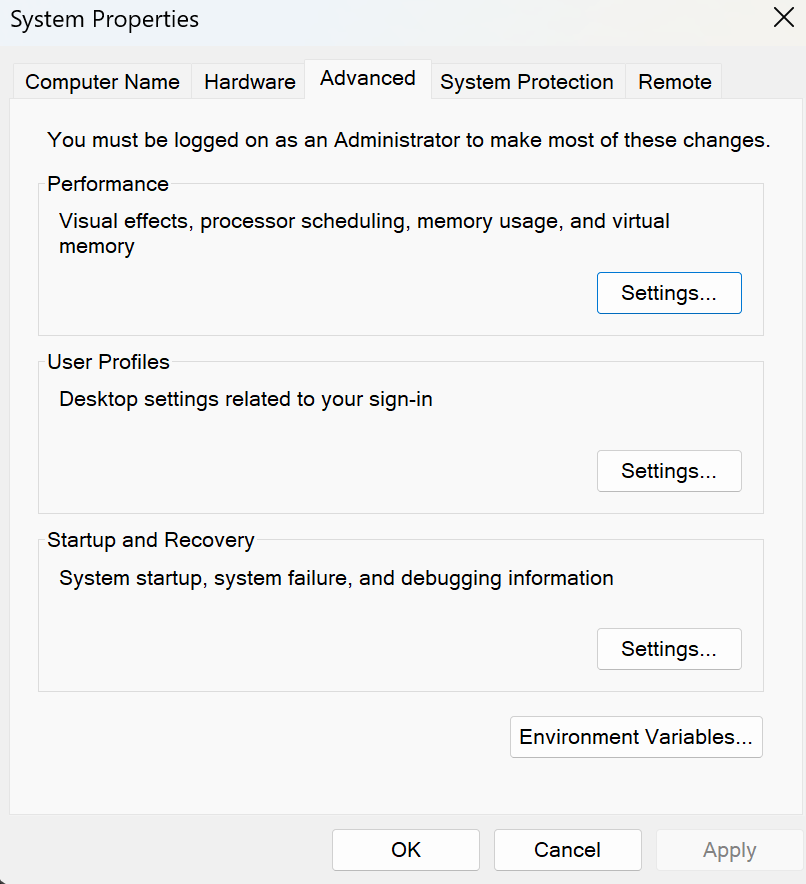
**Step – 3 : After downloading open “C-drive” on your pc and select “Program Files”, open “JDK 21”**

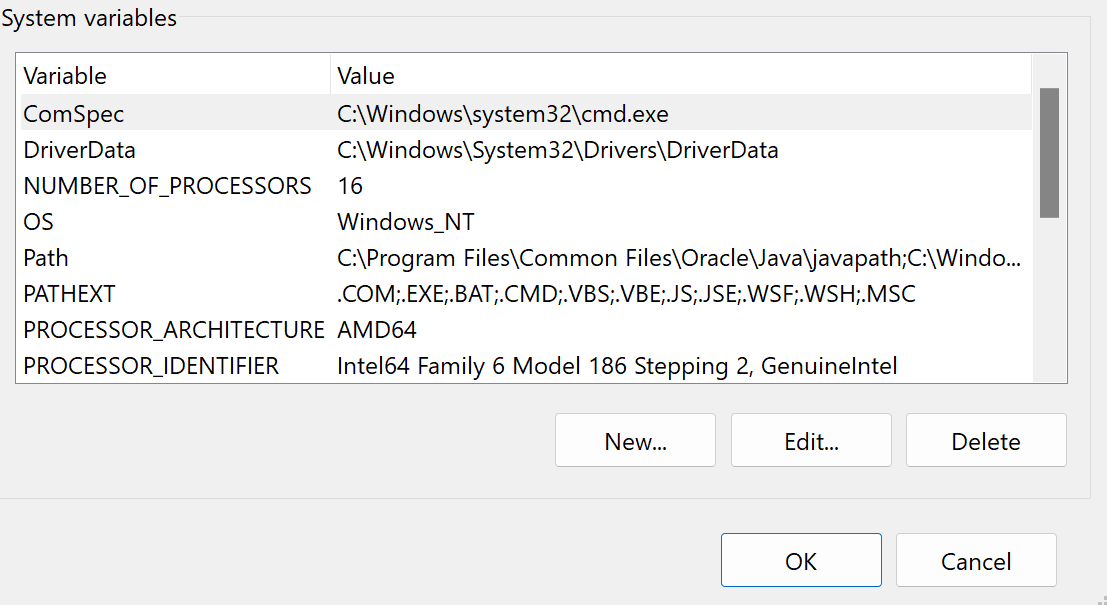




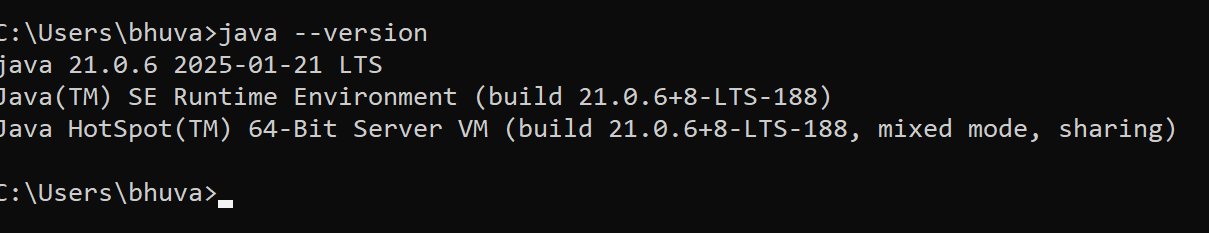


**Step – 4 : Open environmental variables and add a new file with path.**





**Step – 5 : Verify java version in command window**



1. **Write a java program to print the message “Welcome to java programming”.**

**Code:**

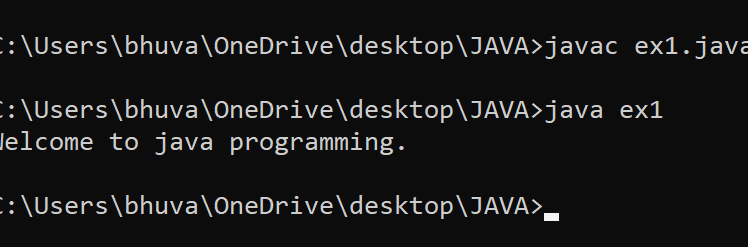
**class ex1 {**

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

**System.out.println("Welcome to java programming.");**

**}**

**}**



**Error :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **S** | **Capital S is expected for String and System.** |

1. **Write a java program to print the student information**

**Code :**

**class ex2{**

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

**System.out.println("Student Information:");**

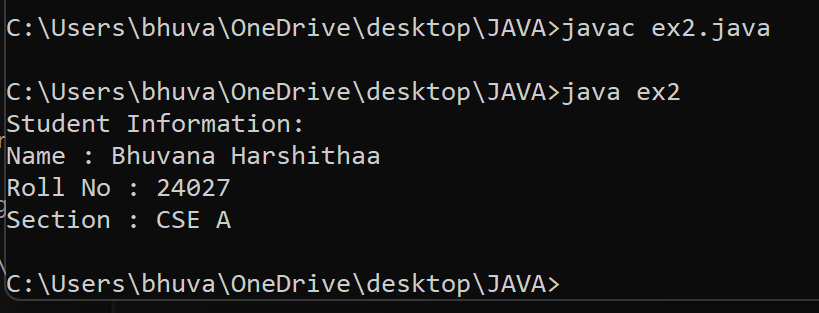
**System.out.println(“Name : Bhuvana Harshithaa”);**

**System.out.println(“Roll No : 24027”)**

**System.out.println(“Section : CSE A”)**

**}**

**}**



**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **S** | **Capital S is expected for String** |

**WEEK – 2**

1. **Write a java program to clalculate area of rectangle.**

**Code : import java.util.Scanner;**

**public class arear{**

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

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

**System.out.print("Enter a value : ");**

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

**System.out.print("Enter a value : ");**

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

**int area = b\*l;**

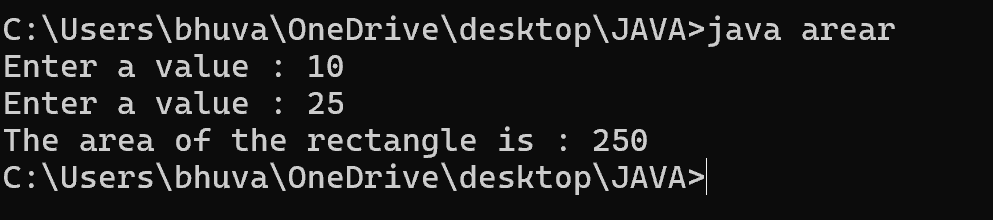
**System.out.print("The area of the rectangle is : "+ area);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **area** | **Declaration of int type variable** |

1. **Write a java program to convert temperature from Celsius to Fahrenheit and vice versa.**

**Code : import java.util.Scanner;**

**class temp{**

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

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

**System.out.print("enter the the temperature in degrees:");**

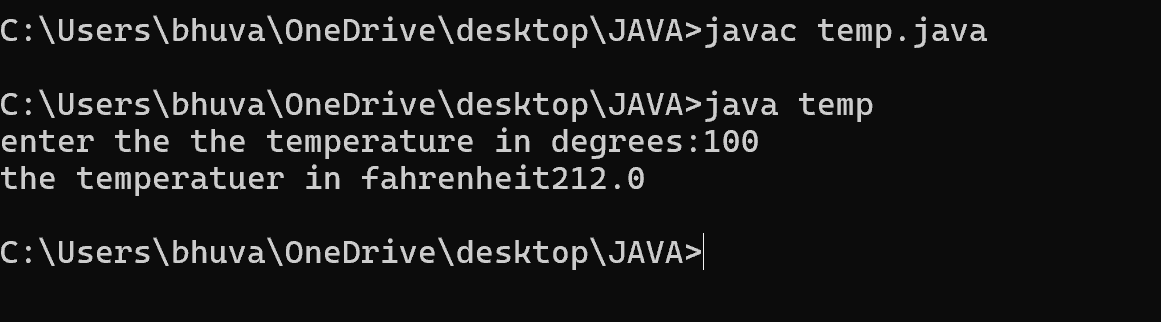
**double deg=input.nextDouble();**

**System.out.println("the temperatuer in fahrenheit"+((deg\*9/5)+32));**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **Input.close();** | **The input is expected to be closed.** |

1. **Write a java program to calculate the simple interest.**

**Code : import java.util.Scanner;**

**public class si{**

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

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

**System.out.print("Enter principal amount : ");**

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

**System.out.print("Enter rate of interest : ");**

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

**System.out.print("Enter the time period : ");**

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

**int SI = p\*r\*t/100;**

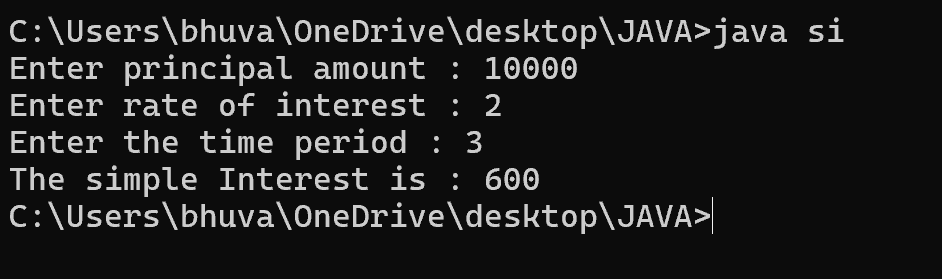
**System.out.print("The simple Interest is : " + SI);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **;** | **; is expected at end** |
| **2** | **Int t** | **Without declaring t the compiler cannot execute the program.** |

1. **Write a java program to find the largest of three numbers using ternary operation.**

**Code : import java.util.Scanner;**

**public class largest{**

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

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

**System.out.print("Enter number a : ");**

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

**System.out.print("Enter number b : ");**

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

**System.out.print("Enter number c : ");**

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

**int largest = (a>=b) ? ((a>=c ) ? a : c) : ((b >=c) ? b : c);**

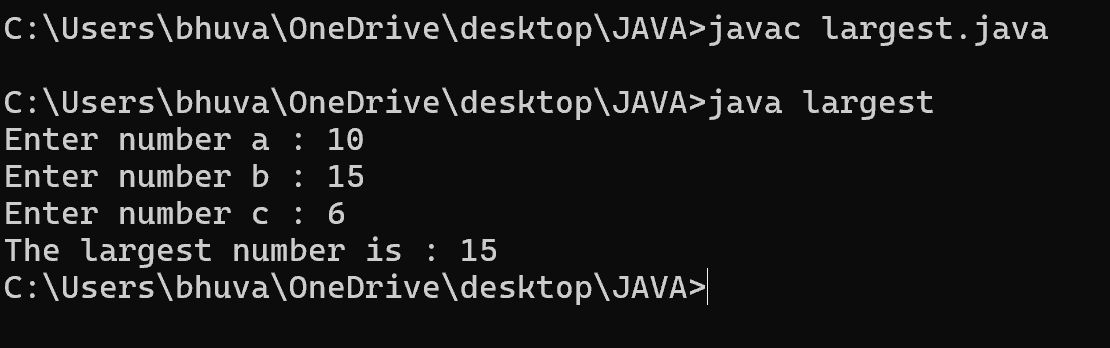
**System.out.print("The largest number is : " + largest);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **?** | **Checks the condition** |
| **2** | **:** | **Comparing between two variables** |

1. **Write a java program to find the factorial of a number**

**Code : import java.util.Scanner;**

**public class fac{**

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

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

**System.out.print("Enter the number n : ");**

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

**int fac = 1;**

**for(int i = 2; i<=n;i++){**

**fac \*= i;**

**}**

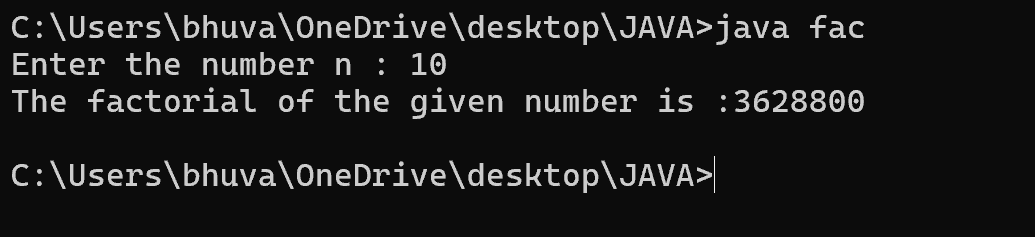
**System.out.println( "The factorial of the given number is :" + fac);**

**input.close();**

**}**

**}**

**Output :**

****

**ERRORS :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **To close for loop** |
| **2** | **System.out.print();** | **If we place the print statement inside the for loop it will print the each i value everytime but to print only the final value we must place it outside the for loop.** |

**WEEK – 3**

1. **Create the java program with the following instructions**
2. **Create a class with name Car**
3. **Create 4 attributes named Car\_Color , Car\_brand, fuel\_type, mileage**
4. **Create 3 method named Start( ) , Stop( ), Service( )**
5. **Create 3 objects Car1 , Car2 , Car3**
6. **Create a constructor which should print “Welcome to Car Garage”**

**Code:** **public class Car{**

**public String carColor;**

**private String carBrand;**

**private String fuelType;**

**public int mileage;**

**Car(String carColor , String carBrand , String fuelType , int mileage){**

**this.carColor = carColor;**

**this.carBrand = carBrand;**

**this.fuelType = fuelType;**

**this.mileage = mileage;**

**System.out.println(carColor + " " + carBrand + " " + fuelType + " " + mileage);**

**}**

**public void Start(){**

**System.out.println("The car has just started");**

**}**

**public void Stop(){**

**System.out.println("The car has just stopped");**

**}**

**public void Service(){**

**System.out.println("The car is in good condition");**

**}**

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

**Car Car1 = new Car("Black","Hyundai","Petrol",100);**

**Car Car2 = new Car("White","Suzuki","Diesel",150);**

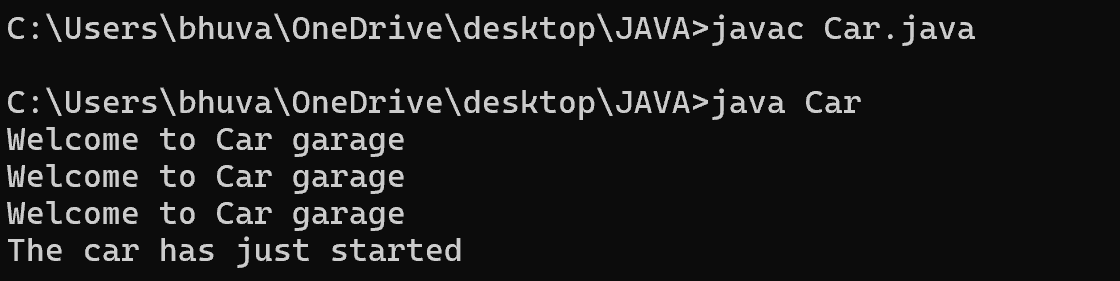
**Car Car3 = new Car("Red","Benz","Petrol",200);**

**Car1.Start();**

**}**

**}**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **} is expected at end of the class** |
| **2** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |

**Class Diagram :**

|  |
| --- |
| **Car** |
| **+ carColor : String**  **- carBrand : String**  **- fuelType : String**  **+ mileage : int** |
| **+ Car( ) : void**  **+ Start( ) : void**  **+ Stop( ) : void**  **+ Service( ) : void** |

1. **Write a java program to create a class BackAccount with two methods deposit( ) and withdraw( )**
2. **In deposit( ) whenever an amount is deposited it has to be updated with current amount**
3. **In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”.**

**Code : public class BankAccount{**

**private String Name;**

**private int AccNo, CurrBal ;**

**BankAccount(String Name, int AccNo, int CurrBal){**

**this.Name = Name;**

**this.AccNo = AccNo;**

**this.CurrBal = CurrBal;**

**System.out.println("The customers are : " + this.Name + " ");**

**}**

**public int deposit(int dAmt){**

**CurrBal = CurrBal + dAmt ;**

**return CurrBal;**

**}**

**public void withdraw(int wAmount){**

**if(wAmount < CurrBal){**

**CurrBal = CurrBal - wAmount ;**

**System.out.println(CurrBal);**

**}**

**else{**

**System.out.println("Insufficient funds");**

**}**

**}**

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

**BankAccount Bhuvana = new BankAccount("Bhuvana",1500,10000);**

**Bhuvana.withdraw(13000);**

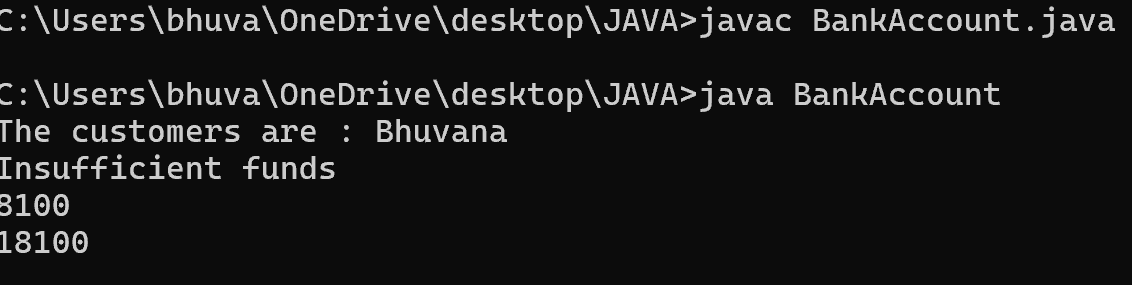
**Bhuvana.withdraw(1900);**

**int FinalAmount = Bhuvana.deposit(10000);**

**System.out.println(FinalAmount);**

**}**

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Expected Error** | **Reason** |
| **1** | **}** | **} is expected at end of the class** |
| **2** | **Setting the parameters inside the constructor** | **We cannot pass the values inside constructor without setting them first** |

**Class Diagram :**

|  |
| --- |
| **BankAccount** |
| **- Name : String**  **- AccNo : String**  **- CurrBal : String** |
| **+ BankAccount( ) : void**  **+ deposit( ) : int**  **+ withdraw( ) : void** |

**WEEK – 4**

1. **Write a java program with class named “Book”. The class should contain various attributes such as**

**“Title of the book , author , year of publication “. It should also contain a constructor with parameters**

**which initializes “ Title of the book, author, year of publication”. Create a method which displays the**

**details of the book. i.e. “ Title of the book, author and year of publication”. Display the details of two**

**books by creating two objects.**

**Code : class Book{**

**// beginning of the class book**

**public String Title;**

**private String author;**

**public int yearOfPublication;**

**// beginning of constructor**

**Book(String Title , String author , int yearOfPublication){**

**this.Title = Title;**

**this.author = author;**

**this.yearOfPublication = yearOfPublication;**

**}**

**//constructor ends here**

**// methos display starts here**

**public void display(){**

**System.out.println("Title of the book is : " + Title + "The name of the author is : " + author +**

**“The year of publication is : " + yearOfPublication );**

**}**

**// method display ends here**

**// creating objects**

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

**Book Book1 = new Book("Harry Potter" , "J.K.Rowling" ,1993);**

**Book Book2 = new Book("Someone Like You" , "Nikitha Singh" , 2010);**

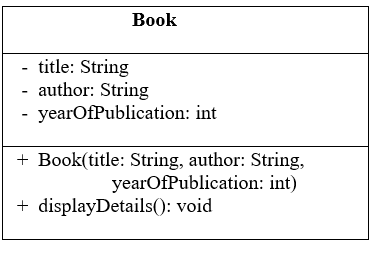
**Book1.display();**

**Book2.display();**

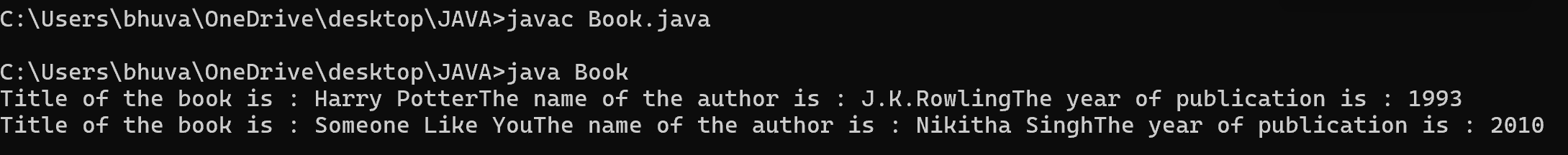
**}**

**}**

**// class ends here**

****

**Output :**

****

**Errors :**

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

1. **To create a java program with class named Myclass 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**

**Now, define a constructor for “Myclass” that increments the “Count” variable each that an object of**

**Myclass is created. Finally , print the final values of “Count” and “pi” variables .**

**Code :**

**class Myclass{**

**// class starts here**

**static int Count = 0;**

**final double pi = 3.1415;**

**// the constructor starts here**

**Myclass(){**

**Count++;**

**}**

**// the constructor ends here**

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

**Myclass c1 = new Myclass();**

**Myclass c2 = new Myclass();**

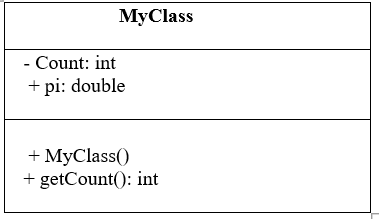
**System.out.println("Count : " + c1.Count);**

**System.out.println("Pi : " + c1.pi);**

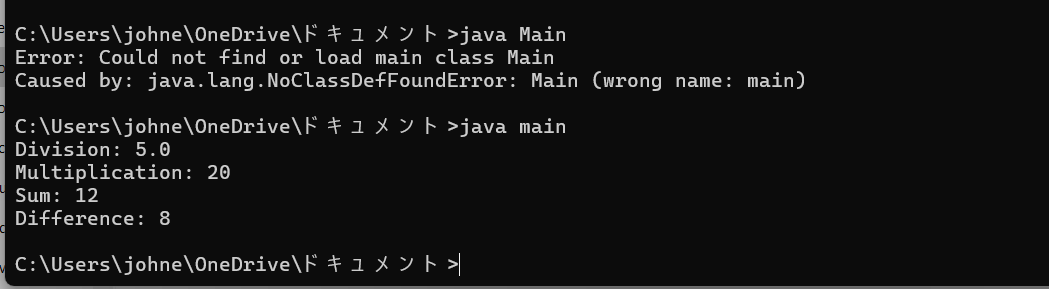
**}**

**}**

**// class ends here**

****

**Output :**

****

**Errors :**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Expected Error** | **Reason** |
| **1** | **.variable** | **We must mention variable name to call the variable** |
| **2** | **static** | **Static variables contain only one value** |

**WEEK 5**

**a) Create a calculator using the operations including addition using subtraction**

**multiplication and division using multilateral inheritance and display the desired output.**

**Code**-

class calculator {

int a, b;

int sum, diff;

bcalc(int a, int b) {

this.a = a;

this.b = b;

}

public void add() {

diff = a - b;

sum = a + b;

System.out.println("Difference: " + diff);

System.out.println("Sum: " + sum);

}

}

class acalc extends calculator {

int mul;

acalc(int a, int b) {

super(a, b);

}

public void mult() {

mul = a \* b;

System.out.println("Multiplication: " + mul);

}

}

class aacalc extends acalc {

float div;

aacalc(int a, int b) {

super(a, b);

}

public void divi() {

if (b != 0) { // Check to avoid division by zero

div = (float) a / b;

System.out.println("Division: " + div);

}

else {

System.out.println("Division by zero error!");

}

}

}

class main {

public static void main(String[] args) {

aacalc c = new aacalc(10, 2);

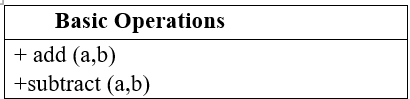
c.divi();

c.mult();

c.add();

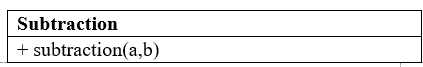
}

}

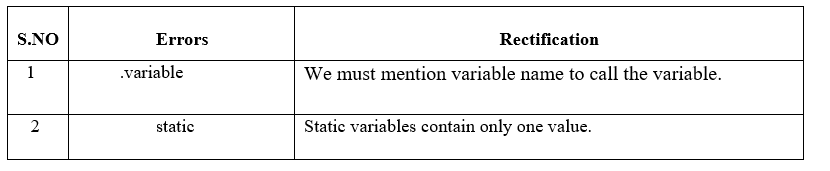
****

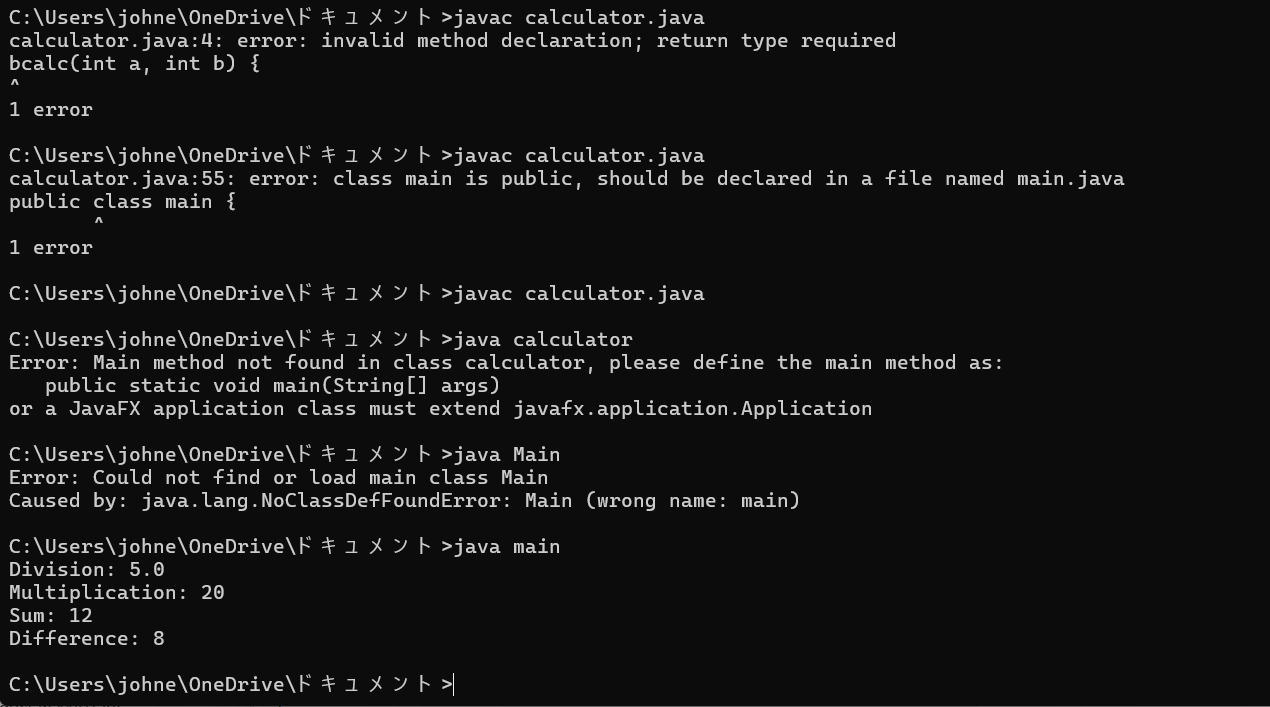
****

****

****

****

****

****

**b) A Vechile rental company wants to develop a system ,that maintains information about**

**different types of vehicles available for rent.The company rents out cars and bikes and**

**they a need a program to store details about each vehicle such as brand and speed .**

**Cars should have an additional properties .**

**“Number of doors “ seating capacity.**

**Bikes should have a property indicating whether they have gears are not ?**

**The system should also include a fuction to display details about each vehicle and indicate when a vechicle is starting .**

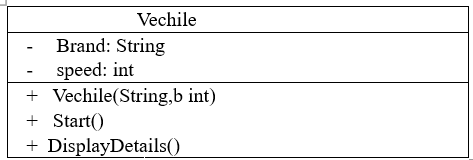
**If the company describes to add a new type of vechile ‘truck’ how would you modify above program.**

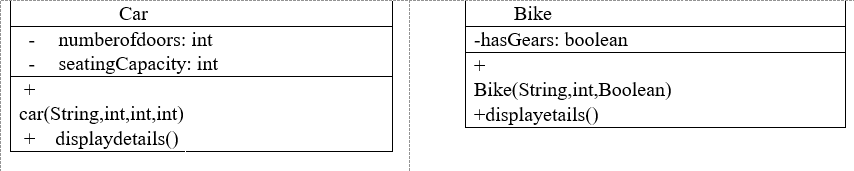
**Truck should include an addition property capacity ‘in tons’.**

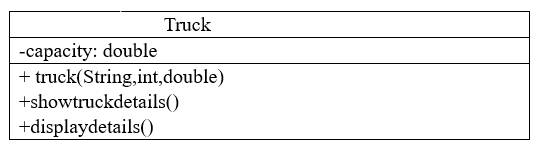
**Create a show truck details method to display the trucks capacity.**

**Write a constructor for truck that initializes all properties.**

**Implement the truck class and update the main method to create a truck object and also create an object and also create an object car and bike subclass find display it details.**

****

****

****

**Code-**

// Base class for Vehicle

class Vehicle {

protected String brand;

protected int speed;

public Vehicle(String brand, int speed) {

this.brand = brand;

this.speed = speed;

}

public void start() {

System.out.println(brand + " is starting.");

}

public void displayDetails() {

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

System.out.println("Speed: " + speed + " km/h");

}

}

// Car class that extends Vehicle

class Car extends Vehicle {

private int numberOfDoors;

private int seatingCapacity;

public Car(String brand, int speed, int numberOfDoors, int seatingCapacity) {

super(brand, speed);

this.numberOfDoors = numberOfDoors;

this.seatingCapacity = seatingCapacity;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Number of Doors: " + numberOfDoors);

System.out.println("Seating Capacity: " + seatingCapacity);

}

}

// Bike class that extends Vehicle

class Bike extends Vehicle {

private boolean hasGears;

public Bike(String brand, int speed, boolean hasGears) {

super(brand, speed);

this.hasGears = hasGears;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Has Gears: " + (hasGears ? "Yes" : "No"));

}

}

// Truck class that extends Vehicle

class Truck extends Vehicle {

private double capacity; // in tons

public Truck(String brand, int speed, double capacity) {

super(brand, speed);

this.capacity = capacity;

}

public void showTruckDetails() {

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

}

@Override

public void displayDetails() {

super.displayDetails();

showTruckDetails();

}

}

// Main class to test the implementation

public class Main {

public static void main(String[] args) {

// Create a Car object

Car car = new Car("Toyota", 180, 4, 5);

car.start();

car.displayDetails();

System.out.println();

// Create a Bike object

Bike bike = new Bike("Yamaha", 120, true);

bike.start();

bike.displayDetails();

System.out.println();

// Create a Truck object

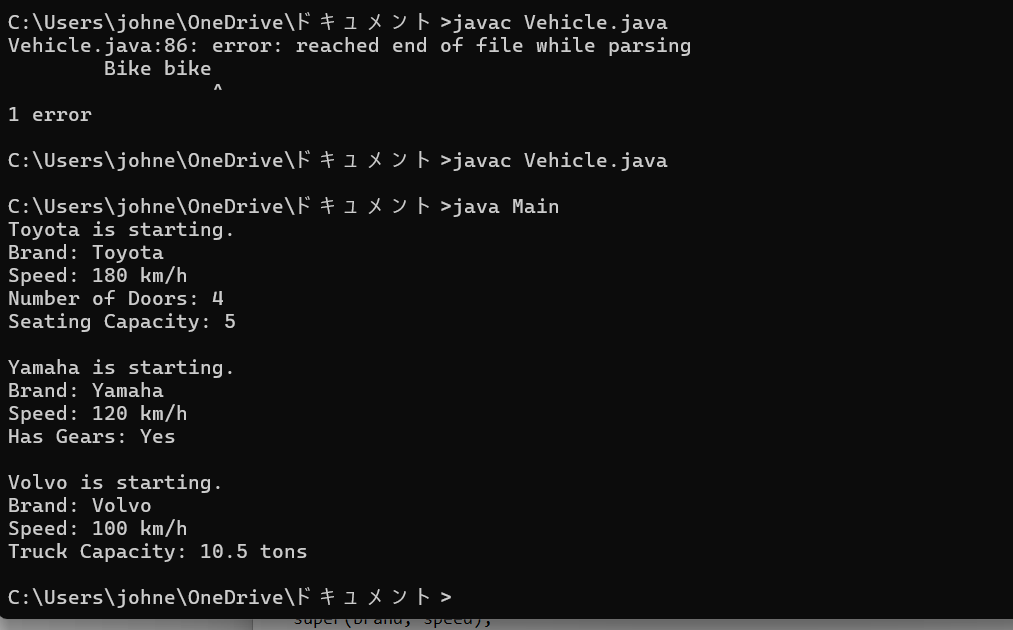
Truck truck = new Truck("Volvo", 100, 10.5);

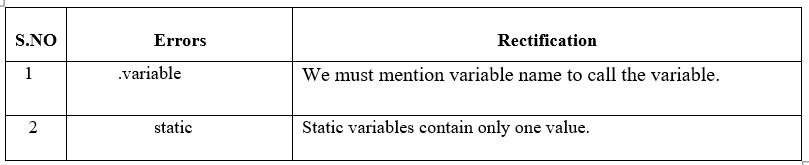
truck.start();

truck.displayDetails();

}

}

****

****

**WEEK-6**

* **Program : 1**

**Q)Write a java program to create a vehicle class with a method override**

**this method in the class subclass to provide. Specific information about**

**the cars that is “car company , model price ,seating capacity, petrol or**

**diesel, true or false”**

**Class Diagram:**

|  |
| --- |
| **Vehicle** |
|  |
| * **displayInfo()** |

|  |
| --- |
| **Car** |
| **- carCompany: String**  **- model: String**  **- price: double**  **- seatingCapacity: int**  **- fuelType: String**  **- isAvailable: boolean** |
| **+ displayInfo()**  **+ Car(company: String, model: String, price: double, seatingCapacity: int, fuelType: String, isAvailable: boolean)** |

**Program:**

**// Base class**

**class Vehicle {**

**// Method to be overridden**

**public void displayInfo() {**

**System.out.println("This is a vehicle.");**

**}**

**}**

**// Subclass**

**class Car extends Vehicle {**

**private String company;**

**private String model;**

**private double price;**

**private int seatingCapacity;**

**private String fuelType; // "Petrol" or "Diesel"**

**private boolean isAvailable; // true or false**

**// Constructor**

**public Car(String company, String model, double price, int seatingCapacity, String fuelType, boolean isAvailable) {**

**this.company = company;**

**this.model = model;**

**this.price = price;**

**this.seatingCapacity = seatingCapacity;**

**this.fuelType = fuelType;**

**this.isAvailable = isAvailable;**

**}**

**// Overriding the displayInfo method**

**@Override**

**public void displayInfo() {**

**System.out.println("Car Company: " + company);**

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

**System.out.println("Price: $" + price);**

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

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

**System.out.println("Available: " + (isAvailable ? "Yes" : "No"));**

**}**

**}**

**// Main class to test the functionality**

**public class VehicleMain {**

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

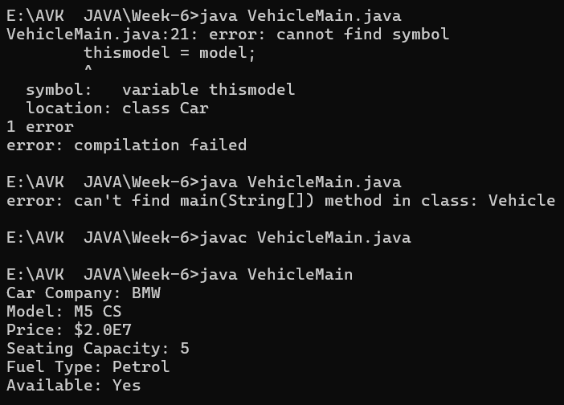
**Vehicle myCar = new Car("BMW", "M5 CS", 20000000, 5, "Petrol", true);**

**myCar.displayInfo();**

**}**

**}**

**Output:**

****

**Error Table:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Errors** | **Rectification** |
| **1** | **thismodel = model;** | **this.model = model;** |
| **2** | **System.out.println(" " + model);** | **System.out.println(" Model" + model);** |

* **Program : 2**

**Q)Write a java program that a Collage is developing a automated**

**admission system that verifies student eligibility for UG and PG**

**programs each program has a different eligibility criteria based on the**

**students percentage in the previous qualification.**

* **UG admission require minimum 60%**
* **PG admission require minimum 70%**

**Class Diagram:**

|  |
| --- |
| **Admission** |
| **+eligibility():void** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **UG** | | **+ eligible(): void** | | |  | | --- | | **PG** | | **+ eligible(): void** | |

**Program:**

**import java.util.Scanner;**

**class College {**

**String name;**

**int qualification;**

**int percentage;**

**// Constructor**

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

**this.name = name;**

**this.qualification = qualification;**

**this.percentage = percentage;**

**}**

**// Default Eligibility method**

**public void Eligibility() {**

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

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

**}**

**}**

**class UG extends College {**

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

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

**}**

**@Override**

**public void Eligibility() {**

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

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

**}**

**}**

**class PG extends College {**

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

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

**}**

**@Override**

**public void Eligibility() {**

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

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

**}**

**}**

**public class CollageMain {**

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

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

**// Taking inputs**

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

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

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

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

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

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

**// Close scanner**

**input.close();**

**// Logic to check eligibility**

**College candidate;**

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

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

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

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

**} else {**

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

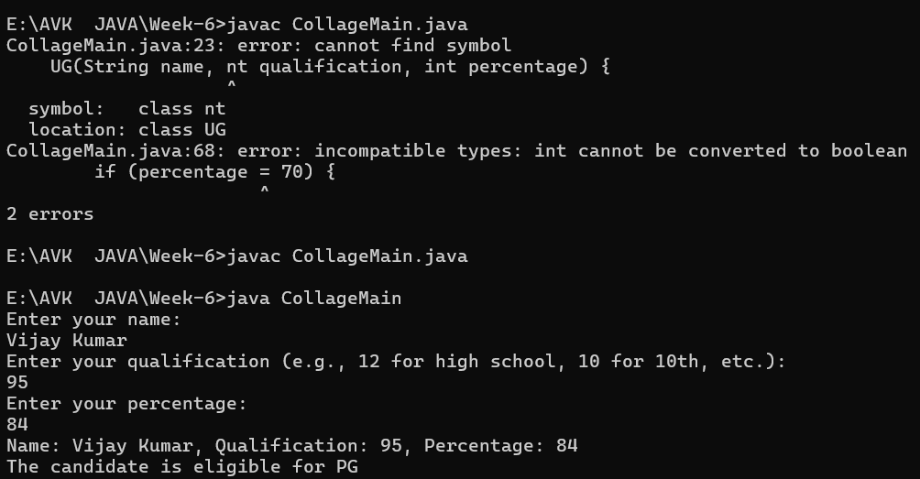
**}**

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

**}**

**}**

**Output:**

****

**Error Table:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Errors** | **Rectification** |
| **1** | **if (percentage = 70) {** | **if (percentage > = 70) {** |
| **2** | **UG(String name, nt qualification, int percentage) {** | **UG(String name, int qualification, int percentage) {** |

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

* **Add two doubles**
* **Add two doubles**
* **Add three Integers**

**Class Diagram:**

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

**Program:**

**public class Calculator{**

**// Method to add two integers**

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

**return a + b;**

**}**

**// Method to add two tuples (represented as arrays)**

**public int add(int[] tuple1, int[] tuple2) {**

**int sum = 0;**

**for (int i = 0; i < tuple1.length; i++) {**

**sum += tuple1[i] + tuple2[i];**

**}**

**return sum;**

**}**

**// Method to add three integers**

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

**return a + b + c;**

**}**

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

**Calculator calc = new Calculator();**

**// Adding two integers**

**int result1 = calc.add(10, 20);**

**System.out.println("Addition of two integers: " + result1);**

**// Adding two tuples (arrays)**

**int[] tuple1 = {1, 2};**

**int[] tuple2 = {3, 4};**

**int result2 = calc.add(tuple1, tuple2);**

**System.out.println("Addition of two tuples: " + result2);**

**// Adding three integers**

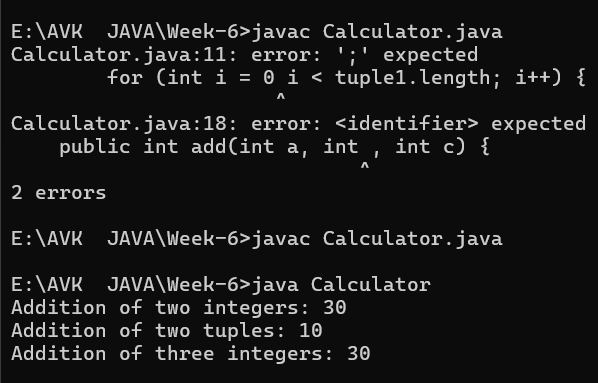
**int result3 = calc.add(5, 10, 15);**

**System.out.println("Addition of three integers: " + result3);**

**}**

**}**

**Output:**

****

**Error Table:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Errors** | **Rectification** |
| **1** | **for (int i = 0 i < tuple1.length; i++) {** | **for (int i = 0; i < tuple1.length; i++) {** |
| **2** | **public int add(int a, int , int c) {** | **public int add(int a, int b, int c) {** |

**Q)Write a java program create a shape class with a method calculate area**

**that is overloaded for different shapes Square,Rectangle then create a**

**sub class circle that overerides the calculate area methods for a circle.**

**Class Diagram:**

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

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

**Program:**

**public class Shape {**

**public double areaOfSquare(double side) {**

**return side \* side;**

**}**

**public double areaOfRectangle(double length, double width) {**

**return length \* width;**

**}**

**public double areaOfCircle(double radius) {**

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

**}**

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

**Shape shape = new Shape();**

**double square = shape.areaOfSquare(5);**

**System.out.println("Area of a square: " + square);**

**double rectangle = shape.areaOfRectangle(10, 20);**

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

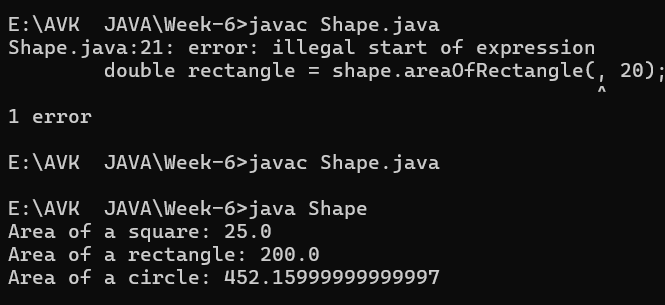
**double circle = shape.areaOfCircle(12);**

**System.out.println("Area of a circle: " + circle);**

**}**

**}**

**Output:**

****

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Errors** | **Rectification** |
| **1** | **Shape shape = Shape();** | **Shape shape = new Shape();** |
| **2** | **double rectangle = shape.areaOfRectangle(, 20);** | **double rectangle = shape.areaOfRectangle(10, 20);** |