**23CSE111**

**OBJECT ORIENTED PROGRAMMING**

**ASSIGNMENT**



**Department of Computer Science Engineering**

**Amrita School of Computing**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Name:M.Pranavi**

**Verified By Roll No: 24302**

|  |  |
| --- | --- |
| **Submitted by** | |
| Name | M.Pranavi |
| Roll No | AV.SC.U4CSE243032 |
| Year/Sem/Section | 1st YEAR/SEM-2/CSE-B |
| Date of Submission |  |
| **Submitted to** | |
| Name | Dr. B Raj Kumar |
| Department | CSE |
| Designation | Asst. Professor |

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

|  |  |  |
| --- | --- | --- |
| S.NO | Q.No | PAGE No. |
| 1. | 1 |  |
| 2. | 2 |  |
| 3. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**PROGRAM-1:**

**AIM:** Write a java program with class named “book”, the class should contain various attributes such as title, author, year of publication it should also contain a constructor with parameters which initializes, title, author, and year of publication.

Create a method which displays the details of the book and display the details of two books.

**CODE:**

public class Book{

public String Title;

public String Author;

public String Year\_of\_publication;

public Book(String Title, String Author, String Year\_of\_publication){

this.Title=Title;

this.Author=Author;

this.Year\_of\_publication=Year\_of\_publication;

}

public void Bookdetails()

{

System.out.println("Title:"+Title);

System.out.println("Author:"+Author);

System.out.println("Year\_of\_publication:"+Year\_of\_publication);

}

public static void main (String[] args){

Book book1=new Book("Mahabaratam","Vyasa","3rd century");

Book book2=new Book("Ramayanam","Valmiki","5th century");

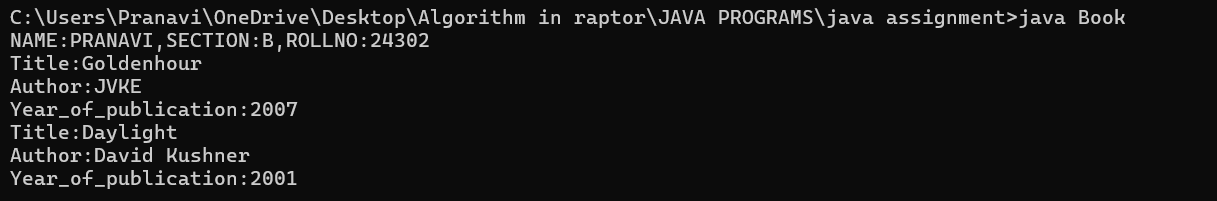
book1.Bookdetails();

book2.Bookdetails();

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Not putting the semi-colon; after calling the function. 2. After Start, Stop, Service not giving the parenthesis ( ). | 1. Put the semi-colon after the writing the code. 2. After every method, put the parenthesis ( ). |

**IMPORTANT POINTS**

1. While defining two classes for a code, we must be sure that we save both the classes in separate files.
2. While defining a method we should also define a function to call that method.

**PROGRAM-2:**

**AIM:**Write 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 3.14159 as attributes of that class. Now define a constructor for “MyClass” that increments the “count” variable each time an object of “MyClass” is created.Finally print the final values of “count” and “PI” variables.

**CODE**:

public class myclass {

static int count = 0;

final double pi = 3.14;

public myclass() {

count++;

}

public static void main(String[] args) {

System.out.println(“NAME:M.Pranavi , SECTION:B, ROLLNO:AV.SC.U4CSE24302”);

myclass a = new myclass();

myclass b = new myclass();

myclass c = new myclass();

myclass d = new myclass();

myclass e = new myclass();

System.out.println("count: " + count);

System.out.println("Value of pi:"+a.pi);

System.out.println("Value of pi:"+b.pi);

System.out.println("Value of pi:"+c.pi);

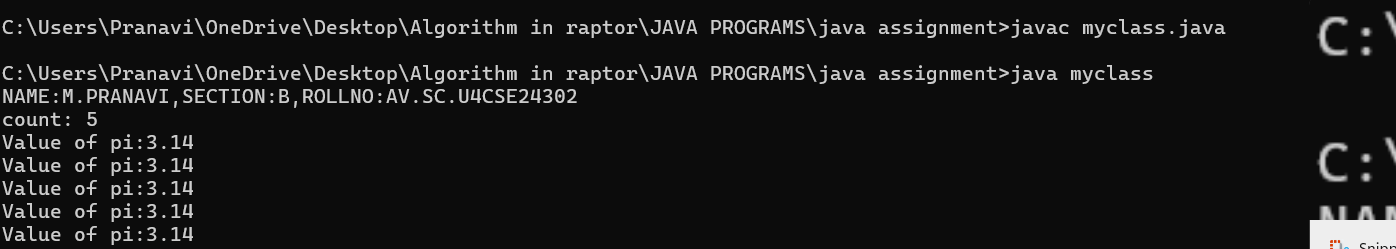
System.out.println("Value of pi:"+d.pi);

System.out.println("Value of pi:"+e.pi);

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Incorrect datatype entered. 2. Not giving the indentation properly. | 1. Enter the correct input during the run-time. 2. All the indentation must be correct to run the code correct. |

**IMPORTANT POINTS:**

1. We must declare the initial value of the variable before declaring the final one.
2. Here the main objective is to increase the count according to the number of objects we make, i.e the count increases when the no.of objects are increasing.

**PROGRAM-3:**

**AIM:**Define a Java class named **VisibilityExample** with the following attributes and methods:

Attributes:

* A public integer variable named publicVariable, initialized to 10.
* A private integer variable named privateVariable, initialized to 20.

Methods:

* A public method named publicMethod() that prints "This is a public method."
* A private method named privateMethod() that prints "This is a private method."
* In a separate Java class named **Main**, write the main method to demonstrate accessing the members of the VisibilityExample class:
* Create an object of the VisibilityExample class.
* Access and print the value of the public variable publicVariable.
* Call the public method publicMethod().

**PROGRAM-4:**

**AIM:**.Write a Java program that takes a number from the user and generates an integer between 1 and 7. It displays the weekday name (Use Conditional Statements).

**CODE**:

import java.util.Scanner;

class WeekdayConditional {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Input number (1-7): ");

int numb = sc.nextInt();

String day;

if (numb == 1) {

day = "Sunday";

} else if (numb == 2) {

day = "Monday";

} else if (numb == 3) {

day = "Tuesday";

} else if (numb == 4) {

day = "Wednesday";

} else if (numb == 5) {

day = "Thursday";

} else if (numb == 6) {

day = "Friday";

} else if (numb == 7) {

day = "Saturday";

} else {

day = "Invalid input! Please enter a number between 1 and 7.";

}

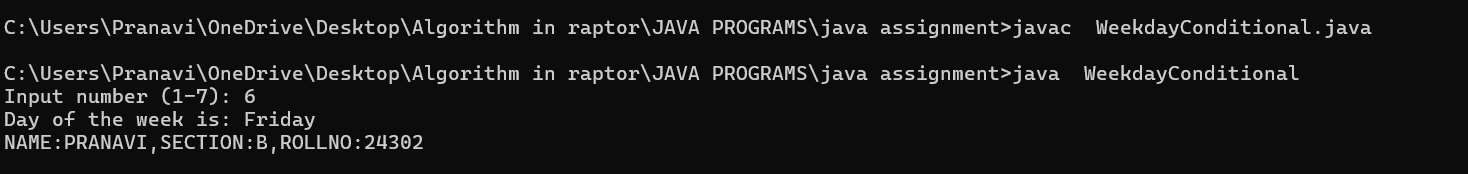
System.out.println("Day of the week is: " + day);

sc.close();

}

}

**Output:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. While printing the variable not giving + sign. 2. Not closing the scanner. | 1. We should give correct indentation. 2. Closing the scanner is must. |

**IMPORTANT POINTS:**

1. While declaring a variable, first we must initialize the variable,
2. Here, we used the Scanner package to take the inputs from the user instead of declaring it beforehand.
3. After using the package we close the scanner package.

**PROGRAM-5:**

**AIM:**Write a Java program to display the multiplication table of a given integer.

Input the number (Table to be calculated) : Input number of terms : 5

**CODE**:

import java.util.Scanner;

public class MultiplicationTable {

public static void main(String[] args) {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

Scanner scanner = new Scanner(System.in);

System.out.print("Input the number (Table to be calculated): ");

int number = scanner.nextInt();

System.out.print("Input number of terms: ");

int terms = scanner.nextInt();

for (int i = 0; i <= terms; i++) {

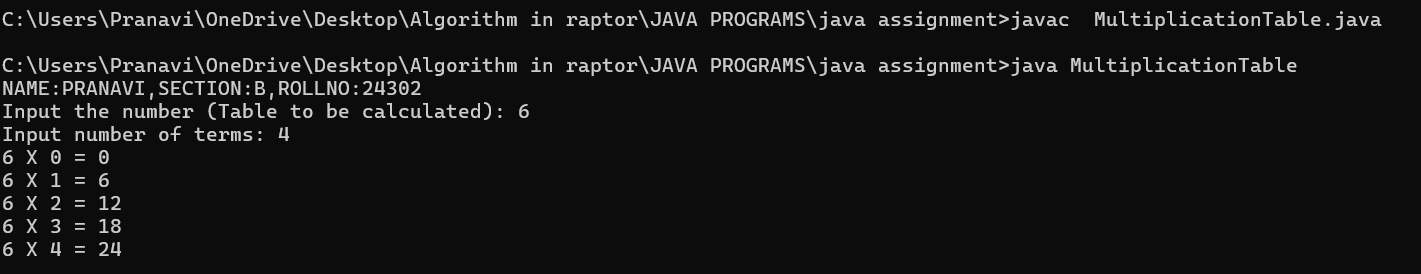
System.out.println(number + " X " + i + " = " + (number \* i));

}

scanner.close();

}

}

**OUTPUT:**  


**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Error while printing the variables. 2. Incorrect declaration of integer. | 1. Give the plus sign while printing. 2. Give input.nextInt(), where I should be capital. |

**IMPORTANT POINTS:**

1. Here, we used the Scanner package to take the inputs from the user instead of declaring it beforehand.
2. After using the package we close the scanner package.

**PROGRAM-6:**

**AIM:**Write a Java program that reads two floating-point numbers and tests whether they are the same up to three decimal places (Use Conditional Statements).

**CODE**:

import java.util.Scanner;

public class CompareFloatNumbers {

public static void main(String[] args) {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

Scanner scanner = new Scanner(System.in);

System.out.print("Input floating-point number: ");

double num1 = scanner.nextDouble();

System.out.print("Input floating-point another number: ");

double num2 = scanner.nextDouble();

num1 = Math.round(num1 \* 1000) / 1000.0;

num2 = Math.round(num2 \* 1000) / 1000.0;

if (num1 == num2) {

System.out.println("They are the same up to three decimal places.");

} else {

System.out.println("They are different.");

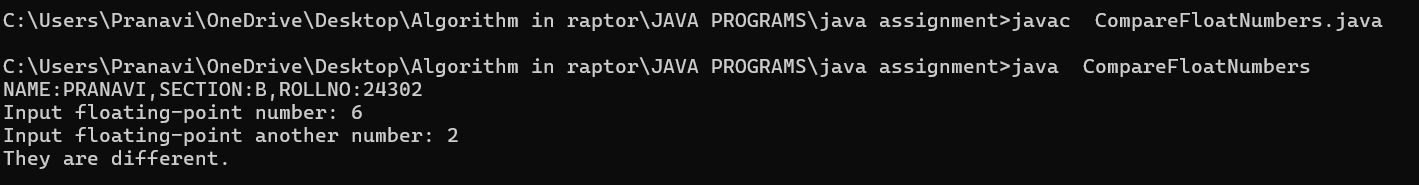
}

scanner.close();

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Invalid datatype. 2. The print statement should be closed at last. | 1. Decalre double instead of int. 2. Add the ‘;’ after each and every statement. |

**IMPORTANT POINTS:**

1. The ‘double’ data type in java states that it can calculate upto any decimal places.

Here, the if and else statement is used as if both the floating-point numbers are equal it prints the if statement or else print

**PROGRAM-7:**

**AIM:**Write a program that accepts three numbers from the user and prints "increasing" if the numbers are in increasing order, "decreasing" if the numbers are in decreasing order, and "Neither increasing or decreasing order" otherwise (Use Conditional Statements).

**CODE**:

import java.util.Scanner;

public class NumberOrder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

System.out.print("Input first number: ");

int num1 = scanner.nextInt();

System.out.print("Input second number: ");

int num2 = scanner.nextInt();

System.out.print("Input third number: ");

int num3 = scanner.nextInt();

if (num1 < num2 && num2 < num3) {

System.out.println("Increasing order");

} else if (num1 > num2 && num2 > num3) {

System.out.println("Decreasing order");

} else {

System.out.println("Neither increasing nor decreasing order");

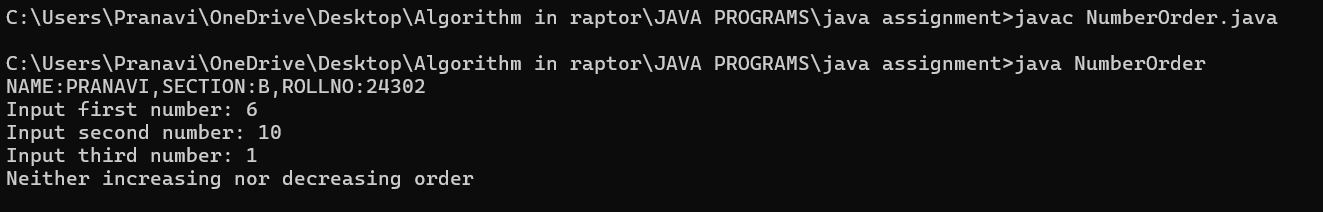
}

scanner.close();

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Incorrect data type. 2. Didn’t close the input. | 1. Declare int instead of Stirng. 2. Close the input at last like ‘input.close();’ |

**IMPORTANT POINTS**:

1. Here, we are finding whether the given numbers are in the increasing order or decreasing order.
2. We do it by using the if and else statements,.
3. According to the given condition in the question, if a<b and b<c, then it prints in increasing order or it prints in decreasing order or it prints neither increasing nor decreasing.

**PROGRAM-8:**

**AIM:**Write a Java program that reads a positive integer and count the number of digits the number (less than ten billion) has (Use Conditional Statements).

**CODE:**

import java.util.Scanner;

public class DigitCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

System.out.print("Input an integer number less than ten billion: ");

long number = scanner.nextLong();

if (number < 0) {

System.out.println("Please enter a positive number.");

} else if (number >= 10\_000\_000\_000L) {

System.out.println("Number should be less than ten billion.");

} else {

int digitCount = 0;

if (number == 0) {

digitCount = 1;

} else {

long temp = number;

while (temp != 0) {

temp /= 10;

digitCount++;

}

}

System.out.println("Number of digits in the number: " + digitCount);

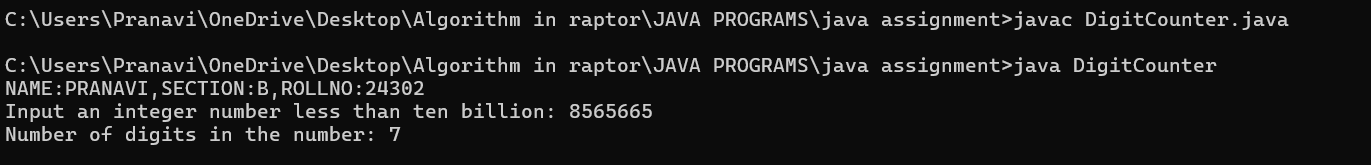
}

scanner.close();

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Variable I not found. 2. Not giving the value of a to b. | 1. Initialize variable I to 0. 2. Giving the value of a to b prints the correct output. |

**IMPORTANT POINTS:**

1. Here, we are using the while loop to execute our program.
2. In the while loop, we are dividing the number by 10 and then increasing the count number.
3. The loop continues until the value of a becomes zero.

**PROGRAM-9:**

**AIM:**Write a Java program to display Pascal's triangle.

**CODE:**

import java.util.Scanner;

public class PascalTriangle {

public static void main(String[] args) {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

Scanner scanner = new Scanner(System.in);

System.out.print("Input number of rows: ");

int rows = scanner.nextInt();

for (int i = 0; i < rows; i++) {

int number = 1;

System.out.format("%" + (rows - i) \* 2 + "s", "");

for (int j = 0; j <= i; j++) {

System.out.format("%4d", number);

number = number \* (i - j) / (j + 1);

}

System.out.println();

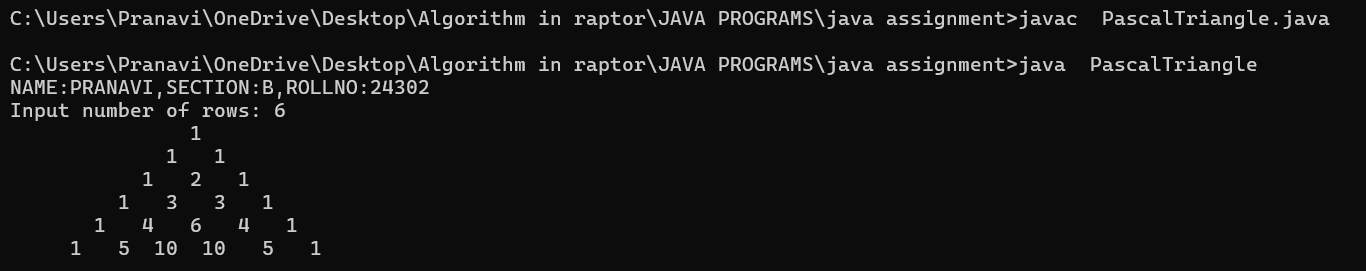
}

scanner.close();

}

}

**OUTPUT:**



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Wrong comma in the for loop. 2. Using println instead of using print. | 1. Put the ‘;’ istead of ‘,’ in the for loop. 2. Thought it runs the code it doesn’t give the right output. |

**IMPORTANT POINTS:**

1. The first inner loop controls the spaces before the numbers in each row to ensure the triangle is properly aligned.
2. R – I calculates the number of spaces needed for the current row.
3. The formula n = n\*(i-j)/(j+1) is a way of calculating the binomial coefficient c(i,j) which represents the value at the jth position in the ith row.

**PROGRAM-10:**

**AIM:**Write a Java program to display the following character rhombus structure.

**CODE:**

import java.util.Scanner;

public class DiamondPattern {

public static void main(String[] args) {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

Scanner sc = new Scanner(System.in);

System.out.print("Input the number: ");

int n = sc.nextInt();

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

for (int j = n - i; j > 0; j--) {

System.out.print(" ");

}

for (int j = 0; j < i; j++) {

System.out.print((char)('A' + j));

}

for (int j = i - 2; j >= 0; j--) {

System.out.print((char)('A' + j));

}

System.out.println();

}

for (int i = n - 1; i >= 1; i--) {

for (int j = n - i; j > 0; j--) {

System.out.print(" ");

}

for (int j = 0; j < i; j++) {

System.out.print((char)('A' + j));

}

for (int j = i - 2; j >= 0; j--) {

System.out.print((char)('A' + j));

}

System.out.println();

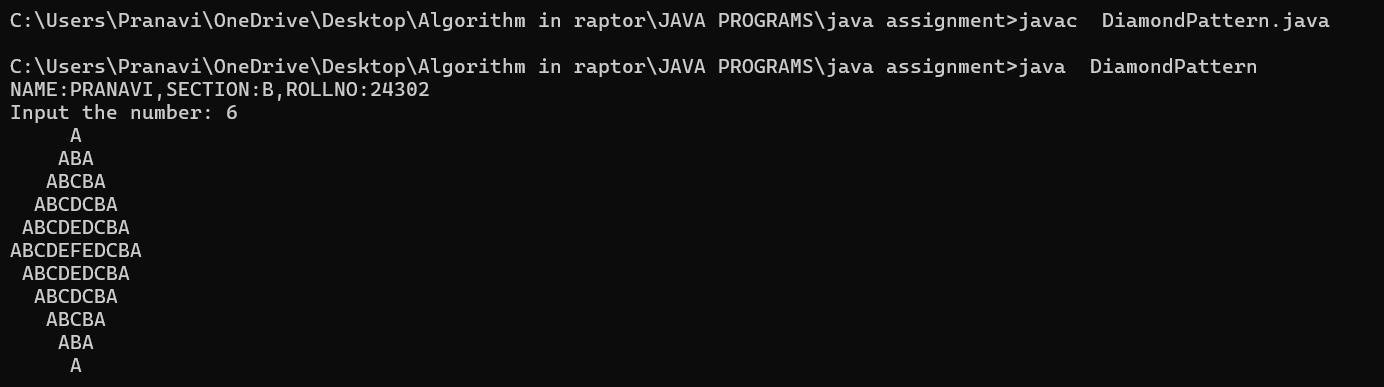
}

sc.close();

}

}

OUTPUT:



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Declaring the int type instead of char. 2. Sometimes printing ‘println’ instead of print. | 1. Declare char instead of int type. 2. Though it gives the output, it gives the wrong answer. |

**IMPORTANT POINTS:**

1. The program starts by taking an integer n as input, which represents the size of the rhombus.
2. The first for loop iterates from 1 to n. It prints the upper half of the rhombus.
3. Second and third loops handle the printing of the alphabet pattern. The second loop prints from A to the i-th letter, and the third loop prints the decreasing sequence of letters to completer the row.

**PROGRAM-11:**

**AIM:**Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance travelled, and maximum speed.

**CODE:**

import java.util.Scanner;

class Vehicle {

public String make;

public String model;

public int year;

public String fuelType;

public Vehicle(String make, String model, int year, String fuelType) {

this.make = make;

this.model = model;

this.year = year;

this.fuelType = fuelType;

}

public void displayInfo() {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

System.out.println("Make: " + make + ", Model: " + model + ", Year: " + year + ", Fuel Type: " + fuelType);

}

}

class Truck extends Vehicle {

public Truck(String make, String model, int year, String fuelType) {

super(make, model, year, fuelType);

}

public void fuelEfficiency() {

System.out.println("Trucks generally have lower fuel efficiency.");

}

}

class Car extends Vehicle {

public Car(String make, String model, int year, String fuelType) {

super(make, model, year, fuelType);

}

public void distanceTravelled(int speed, int time) {

int distance = speed \* time;

System.out.println("Distance traveled: " + distance + " km");

}

}

class Motorcycle extends Vehicle {

public Motorcycle(String make, String model, int year, String fuelType) {

super(make, model, year, fuelType);

}

public void maximumSpeed(int maxSpeed) {

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

}

}

public class VehicleTest {

public static void main(String[] args) {

System.out.println("NAME:PRANAVI,SECTION:B,ROLLNO:24302");

Truck truck = new Truck("TATA", "F-150", 2000, "Diesel");

Car car = new Car("HYUNDAI", "Corolla", 2022, "Petrol");

Motorcycle motorcycle = new Motorcycle("DUKE", "R1", 2019, "Petrol");

truck.displayInfo();

truck.fuelEfficiency();

car.displayInfo();

car.distanceTravelled(80, 2);

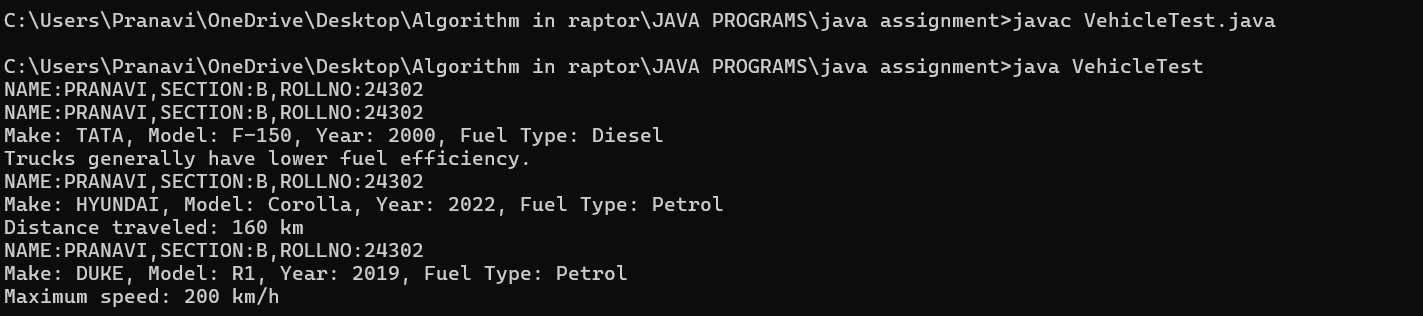
motorcycle.displayInfo();

motorcycle.maximumSpeed(200);

}

}

OUTPUT:



**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Error while printing the print statement. 2. no values in the constructor. | 1. Give + sign when you want to print methods in the print statement. 2. Initialize the value of variables in the constructor. |

**IMPORTANT POINTS:**

1. Here, we declared the class abstract because there are some methods which do not have any return method.
2. Generally, abstract classes are used to define a common structure and behavior for a group of related classes.
3. We created a constructor in the super class and linked it with the subclasses so that the details of the subclasses can also be printed.
4. For that, we use the ‘super’ keyword to access the constructor of the super class.
5. We also override certain methods to access the information.
6. Here, even after initializing values in the constructor, we still declared gettermethods and give a return type for it to access them individually.