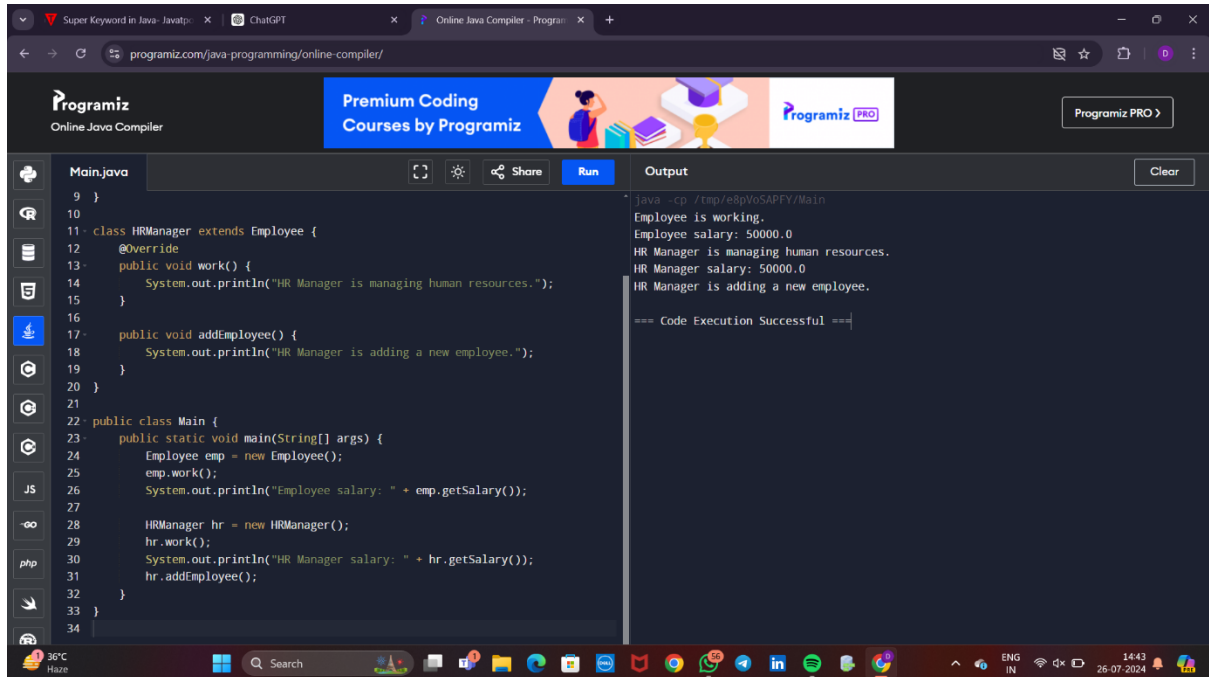


1. Write a Java program to create a class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().



The screenshot shows the Programiz Online Java Compiler interface. The code editor on the left contains the following Java code:

```
9 }
10
11 class HRManager extends Employee {
12     @Override
13     public void work() {
14         System.out.println("HR Manager is managing human resources.");
15     }
16
17     public void addEmployee() {
18         System.out.println("HR Manager is adding a new employee.");
19     }
20 }
21
22 public class Main {
23     public static void main(String[] args) {
24         Employee emp = new Employee();
25         emp.work();
26         System.out.println("Employee salary: " + emp.getSalary());
27
28         HRManager hr = new HRManager();
29         hr.work();
30         System.out.println("HR Manager salary: " + hr.getSalary());
31         hr.addEmployee();
32     }
33 }
34
```

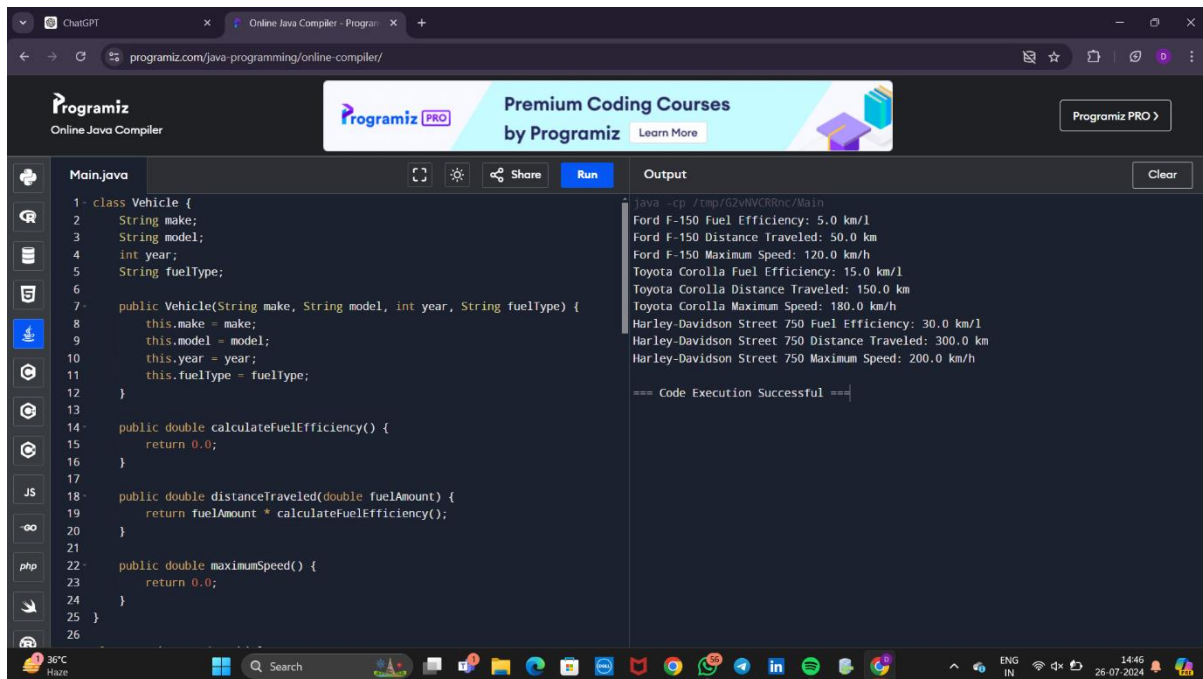
The output window on the right shows the following output:

```
java -cp /tmp/e8pVoSAPFY/Main
Employee is working.
Employee salary: 50000.0
HR Manager is managing human resources.
HR Manager salary: 50000.0
HR Manager is adding a new employee.

=== Code Execution Successful ===
```

The browser's address bar shows the URL `programiz.com/java-programming/online-compiler/`. The Windows taskbar at the bottom shows the date and time as 14:43 on 26-07-2024.

2. 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 traveled, and maximum speed.



The screenshot shows a web browser window with the URL `programiz.com/java-programming/online-compiler/`. The page features a header for "Programiz Online Java Compiler" and a banner for "Premium Coding Courses by Programiz". The main area is divided into two panels: "Main.java" on the left and "Output" on the right.

The "Main.java" panel contains the following Java code:

```
1 class Vehicle {
2     String make;
3     String model;
4     int year;
5     String fuelType;
6
7     public Vehicle(String make, String model, int year, String fuelType) {
8         this.make = make;
9         this.model = model;
10        this.year = year;
11        this.fuelType = fuelType;
12    }
13
14    public double calculateFuelEfficiency() {
15        return 0.0;
16    }
17
18    public double distanceTraveled(double fuelAmount) {
19        return fuelAmount * calculateFuelEfficiency();
20    }
21
22    public double maximumSpeed() {
23        return 0.0;
24    }
25 }
26
```

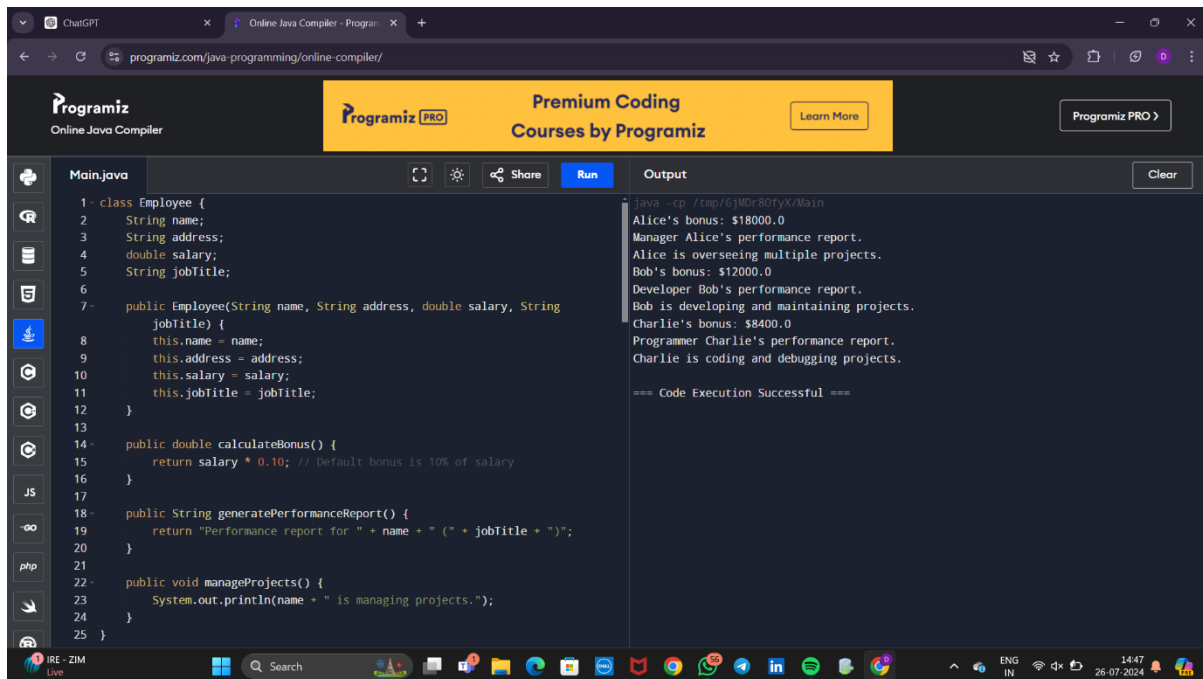
The "Output" panel shows the execution results of the code:

```
java -cp /tmp/G2vWVCR0nc/Main
Ford F-150 Fuel Efficiency: 5.0 km/l
Ford F-150 Distance Traveled: 50.0 km
Ford F-150 Maximum Speed: 120.0 km/h
Toyota Corolla Fuel Efficiency: 15.0 km/l
Toyota Corolla Distance Traveled: 150.0 km
Toyota Corolla Maximum Speed: 180.0 km/h
Harley-Davidson Street 750 Fuel Efficiency: 30.0 km/l
Harley-Davidson Street 750 Distance Traveled: 300.0 km
Harley-Davidson Street 750 Maximum Speed: 200.0 km/h

=== Code Execution Successful ===
```

The bottom of the image shows a Windows taskbar with the date and time as 26-07-2024, 14:46.

3. Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager, Developer, and Programmer. Each subclass should have properties such as name, address, salary, and job title. Implement methods for calculating bonuses, generating performance reports, and managing projects.



The screenshot shows a web browser window with the URL `programiz.com/java-programming/online-compiler/`. The page features a dark theme and a sidebar on the left with icons for various programming languages. The main area is divided into two panels: a code editor on the left and an output window on the right.

The code editor contains the following Java code:

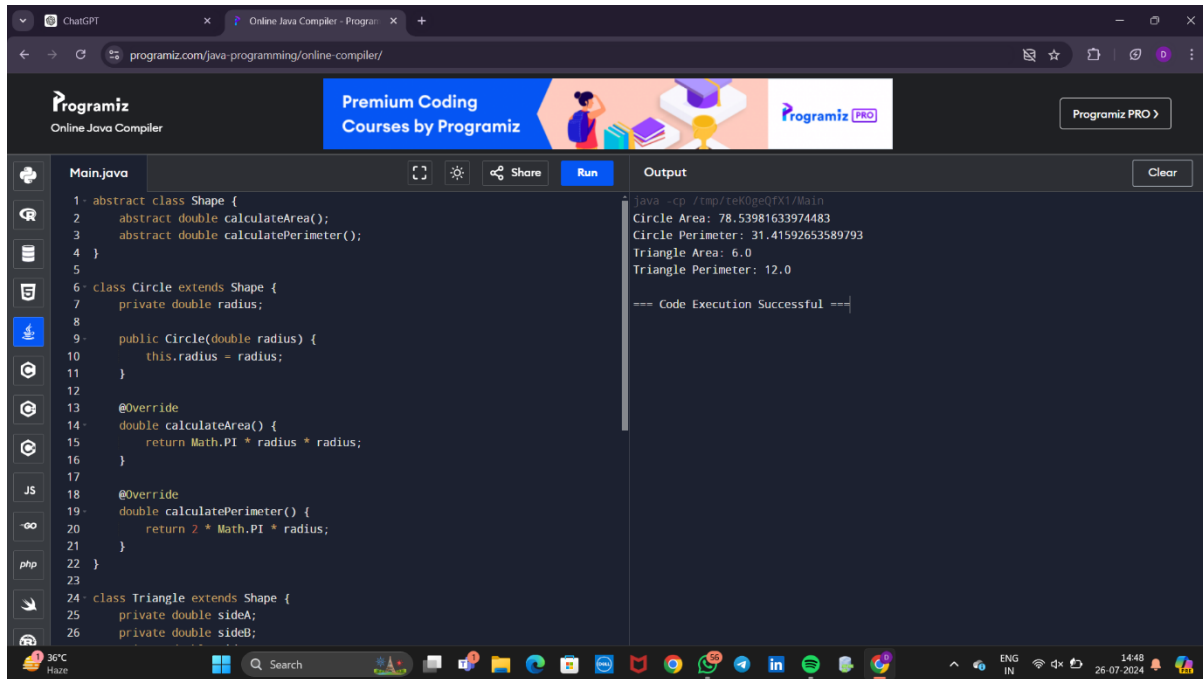
```
1 class Employee {
2     String name;
3     String address;
4     double salary;
5     String jobTitle;
6
7     public Employee(String name, String address, double salary, String
      jobTitle) {
8         this.name = name;
9         this.address = address;
10        this.salary = salary;
11        this.jobTitle = jobTitle;
12    }
13
14    public double calculateBonus() {
15        return salary * 0.10; // Default bonus is 10% of salary
16    }
17
18    public String generatePerformanceReport() {
19        return "Performance report for " + name + " (" + jobTitle + ")";
20    }
21
22    public void manageProjects() {
23        System.out.println(name + " is managing projects.");
24    }
25 }
```

The output window displays the following text:

```
java -cp /tmp/6jWdr80fyX/Main
Alice's bonus: $18000.0
Manager Alice's performance report.
Alice is overseeing multiple projects.
Bob's bonus: $12000.0
Developer Bob's performance report.
Bob is developing and maintaining projects.
Charlie's bonus: $8400.0
Programmer Charlie's performance report.
Charlie is coding and debugging projects.

=== Code Execution Successful ===
```

4. Write a Java program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape



```
1 abstract class Shape {
2     abstract double calculateArea();
3     abstract double calculatePerimeter();
4 }
5
6 class Circle extends Shape {
7     private double radius;
8
9     public Circle(double radius) {
10         this.radius = radius;
11     }
12
13     @Override
14     double calculateArea() {
15         return Math.PI * radius * radius;
16     }
17
18     @Override
19     double calculatePerimeter() {
20         return 2 * Math.PI * radius;
21     }
22 }
23
24 class Triangle extends Shape {
25     private double sideA;
26     private double sideB;
```

Output

```
java -cp /tmp/teK0geQfX1/Main
Circle Area: 78.53981633974483
Circle Perimeter: 31.41592653589793
Triangle Area: 6.0
Triangle Perimeter: 12.0

=== Code Execution Successful ===
```