

1. Write a Java program to create a class called Animal with a method called makeSound(). Create a subclass called Cat that overrides the makeSound() method to meow.
Expected Output:
Some generic animal sound
Meow
2. Write a Java program to create a class called Vehicle with a method called drive(). Create a subclass called Car that overrides the drive() method to print "Repairing a car".
Expected Output:
Driving a vehicle
Repairing a car
3. Write a Java program to create a class called Shape with a method called getArea(). Create a subclass called Rectangle that overrides the getArea() method to calculate the area of a rectangle.
Expected Output:
Calculating area in Shape class
Generic Shape Area: 0.0
Rectangle Area: 15.0
4. 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().
Expected Output:
John is working.
Salary: \$50000.0
Alice is managing HR tasks.
Alice added a new employee: Bob
Salary: \$70000.0
5. Write a Java program to create a class known as "BankAccount" with methods called deposit() and withdraw(). Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.
Expected Output:
Current balance: \$500.0
Deposited: \$200.0
Current balance: \$700.0
Withdrew: \$550.0
Current balance: \$150.0
Withdrawal denied: Balance cannot fall below \$100
Current balance: \$150.0
6. Write a Java program to create a class called Animal with a method named move(). Create a subclass called Cheetah that overrides the move() method to run.
Expected Output:
Animal is moving
Running

7. Write a Java program to create a class known as Person with methods called `getFirstName()` and `getLastName()`. Create a subclass called Employee that adds a new method named `getEmployeeId()` and overrides the `getLastName()` method to include the employee's job title.

Expected Output:

Person: Jane Doe

Employee: John Smith (Manager)

Employee ID: E456

8. Write a Java program to create a class called Shape with methods called `getPerimeter()` and `getArea()`. Create a subclass called Circle that overrides the `getPerimeter()` and `getArea()` methods to calculate the area and perimeter of a circle. Consider, Radius=5.

Expected Output:

Calculating perimeter in Shape class

Generic Shape Perimeter: 0.0

Calculating area in Shape class

Generic Shape Area: 0.0

Circle Perimeter: 31.41592653589793

Circle Area: 78.53981633974483

9. 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.

Expected Output:

=== Truck Info ===

Make: Scania, Model: R-Series, Year: 2021, Fuel: Diesel

Fuel Efficiency: 6.0 km/l

Distance for 40L: 240.0 km

Max Speed: 110.0 km/h

=== Car Info ===

Make: Honda, Model: Civic, Year: 2023, Fuel: Petrol

Fuel Efficiency: 18.0 km/l

Distance for 40L: 720.0 km

Max Speed: 190.0 km/h

=== Motorcycle Info ===

Make: Suzuki, Model: Gixxer, Year: 2022, Fuel: Petrol

Fuel Efficiency: 40.0 km/l

Distance for 10L: 400.0 km

Max Speed: 150.0 km/h

10. 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.

Expected Output:

=== Manager ===

Name: Alice

Address: 123 Business Rd

Salary: \$90000.0

Job Title: Manager

Manager Report for Alice: Excellent leadership and planning.

Alice is managing the entire project and leading the team.

Bonus: \$18000.0

=== Developer ===

Name: Bob

Address: 456 Code Ave

Salary: \$75000.0

Job Title: Developer

Developer Report for Bob: Good coding and debugging skills.

Bob is developing key modules of the project.

Bonus: \$7500.0

=== Programmer ===

Name: Charlie

Address: 789 Dev Blvd

Salary: \$65000.0

Job Title: Programmer

Programmer Report for Charlie: Efficient in writing clean code.

Charlie is writing and testing code for assigned tasks.

Bonus: \$5200.0