Exercise 1:

Create a class called "Animal" with a method called "sound." Create two subclasses "Dog" and "Cat" that override the "sound" method to print different sounds for each animal.

public class Animal {

public void sound() {

System.out.println("Animal makes a sound");

}

}

public class Dog extends Animal {

@Override

public void sound() {

System.out.println("Dog barks");

}

}

public class Cat extends Animal {

@Override

public void sound() {

System.out.println("Cat meows");

}

}

public class Exercise1 {

public static void main(String[] args) {

Animal animal1 = new Dog();

Animal animal2 = new Cat();

animal1.sound();

animal2.sound();

}

}

Exercise 2:

Create a superclass "Shape" with a method called "area." Create two subclasses "Circle" and "Rectangle" that override the "area" method to calculate the area for each shape.

public class Shape {

public double area() {

return 0.0;

}

}

public class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI \* radius \* radius;

}

}

public class Rectangle extends Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double area() {

return length \* width;

}

}

public class Exercise2 {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Area of Circle: " + circle.area());

System.out.println("Area of Rectangle: " + rectangle.area());

}

}

Exercise 3:

Create a superclass "Vehicle" with a method called "start." Create two subclasses "Car" and "Bike" that override the "start" method to print different messages for each vehicle.

public class Vehicle {

public void start() {

System.out.println("Vehicle starts");

}

}

public class Car extends Vehicle {

@Override

public void start() {

System.out.println("Car starts with a key");

}

}

public class Bike extends Vehicle {

@Override

public void start() {

System.out.println("Bike starts with a kick");

}

}

public class Exercise3 {

public static void main(String[] args) {

Vehicle vehicle1 = new Car();

Vehicle vehicle2 = new Bike();

vehicle1.start();

vehicle2.start();

}

}

Exercise 4:

Create a superclass "Person" with attributes "name" and "age." Create a subclass "Employee" that extends "Person" and has an additional attribute "salary." Override the "toString" method in "Employee" to display all attributes.

pubic class Person {

protected String name;

protected int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

}

public class Employee extends Person {

private double salary;

public Employee(String name, int age, double salary) {

super(name, age);

this.salary = salary;

}

@Override

public String toString() {

return "Name: " + name + ", Age: " + age + ", Salary: " + salary;

}

}

public class Exercise4 {

public static void main(String[] args) {

Employee emp = new Employee("John", 30, 50000.0);

System.out.println(emp.toString());

}

}

Exercise 5:

Create a superclass "Bird" with a method called "fly." Create two subclasses "Eagle" and "Penguin" that override the "fly" method to print appropriate messages for each bird.

class Bird {

public void fly() {

System.out.println("Bird can fly");

}

}

class Eagle extends Bird {

@Override

public void fly() {

System.out.println("Eagle flies high in the sky");

}

}

class Penguin extends Bird {

@Override

public void fly() {

System.out.println("Penguin cannot fly");

}

}

public class Exercise5 {

public static void main(String[] args) {

Bird bird1 = new Eagle();

Bird bird2 = new Penguin();

bird1.fly();

bird2.fly();

}

}

Exercise 6:

Create a superclass "Account" with attributes "accountNumber" and "balance." Create two subclasses "CheckingAccount" and "SavingsAccount" that override a method called "withdraw" to handle specific withdrawal scenarios.

class Account {

protected String accountNumber;

protected double balance;

public Account(String accountNumber, double balance) {

this.accountNumber = accountNumber;

this.balance = balance;

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal successful. Remaining balance: " + balance);

} else {

System.out.println("Insufficient balance.");

}

}

}

class CheckingAccount extends Account {

public CheckingAccount(String accountNumber, double balance) {

super(accountNumber, balance);

}

@Override

public void withdraw(double amount) {

super.withdraw(amount);

System.out.println("Checking Account withdrawal charges: $2");

balance -= 2;

System.out.println("Updated balance after charges: " + balance);

}

}

class SavingsAccount extends Account {

public SavingsAccount(String accountNumber, double balance) {

super(accountNumber, balance);

}

@Override

public void withdraw(double amount) {

super.withdraw(amount);

System.out.println("Savings Account withdrawal limit: $500");

}

}

public class Exercise6 {

public static void main(String[] args) {

CheckingAccount checkingAcc = new CheckingAccount("C1001", 1000);

SavingsAccount savingsAcc = new SavingsAccount("S2001", 1500);

checkingAcc.withdraw(200);

savingsAcc.withdraw(1000);

}

}

Exercise 7:

Create a superclass "Vehicle" with attributes "make" and "model." Create two subclasses "Car" and "Motorcycle" that override the "toString" method to display all attributes.

class Vehicle {

protected String make;

protected String model;

public Vehicle(String make, String model) {

this.make = make;

this.model = model;

}

@Override

public String toString() {

return "Make: " + make + ", Model: " + model;

}

}

class Car extends Vehicle {

public Car(String make, String model) {

super(make, model);

}

}

class Motorcycle extends Vehicle {

public Motorcycle(String make, String model) {

super(make, model);

}

}

public class Exercise7 {

public static void main(String[] args) {

Car car = new Car("Toyota", "Camry");

Motorcycle bike = new Motorcycle("Harley Davidson", "Street 750");

System.out.println("Car Details: " + car.toString());

System.out.println("Motorcycle Details: " + bike.toString());

}

}

Exercise 8:

Create a superclass "Person" with a method called "displayDetails." Create two subclasses "Student" and "Teacher" that override the "displayDetails" method to print appropriate details for each person.

public class Person {

protected String name;

protected int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public void displayDetails() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

class Student extends Person {

private String studentId;

public Student(String name, int age, String studentId) {

super(name, age);

this.studentId = studentId;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Student ID: " + studentId);

}

}

class Teacher extends Person {

private String employeeId;

public Teacher(String name, int age, String employeeId) {

super(name, age);

this.employeeId = employeeId;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Employee ID: " + employeeId);

}

}

public class Exercise8 {

public static void main(String[] args) {

Person person1 = new Student("John Doe", 20, "S12345");

Person person2 = new Teacher("Jane Smith", 35, "T98765");

person1.displayDetails();

person2.displayDetails();

}

}

Exercise 9:

Create a superclass "Shape" with attributes "color" and "filled." Create two subclasses "Circle" and "Rectangle" that override the "toString" method to display all attributes.

class Shape {

protected String color;

protected boolean filled;

public Shape(String color, boolean filled) {

this.color = color;

this.filled = filled;

}

@Override

public String toString() {

return "Color: " + color + ", Filled: " + filled;

}

}

class Circle extends Shape {

private double radius;

public Circle(String color, boolean filled, double radius) {

super(color, filled);

this.radius = radius;

}

@Override

public String toString() {

return super.toString() + ", Radius: " + radius;

}

}

class Rectangle extends Shape {

private double width;

private double length;

public Rectangle(String color, boolean filled, double width, double length) {

super(color, filled);

this.width = width;

this.length = length;

}

@Override

public String toString() {

return super.toString() + ", Width: " + width + ", Length: " + length;

}

}

public class Exercise9 {

public static void main(String[] args) {

Circle circle = new Circle("Red", true, 5.0);

Rectangle rectangle = new Rectangle("Blue", false, 4.0, 6.0);

System.out.println("Circle Details: " + circle.toString());

System.out.println("Rectangle Details: " + rectangle.toString());

}

}

Exercise 10:

Create a superclass "Animal" with a method called "sound." Create two subclasses "Dog" and "Cat" that override the "sound" method to print different sounds for each animal.

class Animal {

public void sound() {

System.out.println("Animal makes a sound");

}

}

class Dog extends Animal {

@Override

public void sound() {

System.out.println("Dog barks");

}

}

class Cat extends Animal {

@Override

public void sound() {

System.out.println("Cat meows");

}

}

public class Exercise10 {

public static void main(String[] args) {

Animal animal1 = new Dog();

Animal animal2 = new Cat();

animal1.sound();

animal2.sound();

}

}

Exercise 11:

Create a superclass "Shape" with a method called "area." Create two subclasses "Circle" and "Rectangle" that override the "area" method to calculate the area for each shape.

class Shape {

public double area() {

return 0.0;

}

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI \* radius \* radius;

}

}

class Rectangle extends Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double area() {

return length \* width;

}

}

public class Exercise11 {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Area of Circle: " + circle.area());

System.out.println("Area of Rectangle: " + rectangle.area());

}

}

Exercise 12:

Create a superclass "Vehicle" with a method called "start." Create two subclasses "Car" and "Bike" that override the "start" method to print different messages for each vehicle.

class Vehicle {

public void start() {

System.out.println("Vehicle starts");

}

}

class Car extends Vehicle {

@Override

public void start() {

System.out.println("Car starts with a key");

}

}

class Bike extends Vehicle {

@Override

public void start() {

System.out.println("Bike starts with a kick");

}

}

public class Exercise12 {

public static void main(String[] args) {

Vehicle vehicle1 = new Car();

Vehicle vehicle2 = new Bike();

vehicle1.start();

vehicle2.start();

}

}

Exercise 13:

Create a superclass "Person" with attributes "name" and "age." Create a subclass "Employee" that extends "Person" and has an additional attribute "salary." Override the "toString" method in "Employee" to display all attributes.

class Person {

protected String name;

protected int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

}

class Employee extends Person {

private double salary;

public Employee(String name, int age, double salary) {

super(name, age);

this.salary = salary;

}

@Override

public String toString() {

return "Name: " + name + ", Age: " + age + ", Salary: " + salary;

}

}

public class Exercise13 {

public static void main(String[] args) {

Employee emp = new Employee("John", 30, 50000.0);

System.out.println(emp.toString());

}

}

Exercise 14:

Create a superclass "Bird" with a method called "fly." Create two subclasses "Eagle" and "Penguin" that override the "fly" method to print appropriate messages for each bird.

class Bird {

public void fly() {

System.out.println("Bird can fly");

}

}

class Eagle extends Bird {

@Override

public void fly() {

System.out.println("Eagle flies high in the sky");

}

}

class Penguin extends Bird {

@Override

public void fly() {

System.out.println("Penguin cannot fly");

}

}

public class Exercise14 {

public static void main(String[] args) {

Bird bird1 = new Eagle();

Bird bird2 = new Penguin();

bird1.fly();

bird2.fly();

}

}

Exercise 15:

Create a superclass "Account" with attributes "accountNumber" and "balance." Create two subclasses "CheckingAccount" and "SavingsAccount" that override a method called "withdraw" to handle specific withdrawal scenarios.

class Account {

protected String accountNumber;

protected double balance;

public Account(String accountNumber, double balance) {

this.accountNumber = accountNumber;

this.balance = balance;

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal successful. Remaining balance: " + balance);

} else {

System.out.println("Insufficient balance.");

}

}

}

class CheckingAccount extends Account {

public CheckingAccount(String accountNumber, double balance) {

super(accountNumber, balance);

}

@Override

public void withdraw(double amount) {

super.withdraw(amount);

System.out.println("Checking Account withdrawal charges: $2");

balance -= 2;

System.out.println("Updated balance after charges: " + balance);

}

}

class SavingsAccount extends Account {

public SavingsAccount(String accountNumber, double balance) {

super(accountNumber, balance);

}

@Override

public void withdraw(double amount) {

super.withdraw(amount);

System.out.println("Savings Account withdrawal limit: $500");

}

}

public class Exercise15 {

public static void main(String[] args) {

CheckingAccount checkingAcc = new CheckingAccount("C1001", 1000);

SavingsAccount savingsAcc = new SavingsAccount("S2001", 1500);

checkingAcc.withdraw(200);

savingsAcc.withdraw(1000);

}

}

Exercise 16:

Create a superclass "Vehicle" with attributes "make" and "model." Create two subclasses "Car" and "Motorcycle" that override the "toString" method to display all attributes.

class Vehicle {

protected String make;

protected String model;

public Vehicle(String make, String model) {

this.make = make;

this.model = model;

}

@Override

public String toString() {

return "Make: " + make + ", Model: " + model;

}

}

class Car extends Vehicle {

public Car(String make, String model) {

super(make, model);

}

}

class Motorcycle extends Vehicle {

public Motorcycle(String make, String model) {

super(make, model);

}

}

public class Exercise16 {

public static void main(String[] args) {

Car car = new Car("Toyota", "Camry");

Motorcycle bike = new Motorcycle("Harley Davidson", "Street 750");

System.out.println("Car Details: " + car.toString());

System.out.println("Motorcycle Details: " + bike.toString());

}

}

Exercise 17:

Create a superclass "Person" with a method called "displayDetails." Create two subclasses "Student" and "Teacher" that override the "displayDetails" method to print appropriate details for each person.

class Person {

protected String name;

protected int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public void displayDetails() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

class Student extends Person {

private String studentId;

public Student(String name, int age, String studentId) {

super(name, age);

this.studentId = studentId;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Student ID: " + studentId);

}

}

class Teacher extends Person {

private String employeeId;

public Teacher(String name, int age, String employeeId) {

super(name, age);

this.employeeId = employeeId;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Employee ID: " + employeeId);

}

}

public class Exercise17 {

public static void main(String[] args) {

Person person1 = new Student("John Doe", 20, "S12345");

Person person2 = new Teacher("Jane Smith", 35, "T98765");

person1.displayDetails();

person2.displayDetails();

}

}

Exercise 18:

Create a superclass "Shape" with attributes "color" and "filled." Create two subclasses "Circle" and "Rectangle" that override the "toString" method to display all attributes.

class Shape {

protected String color;

protected boolean filled;

public Shape(String color, boolean filled) {

this.color = color;

this.filled = filled;

}

@Override

public String toString() {

return "Color: " + color + ", Filled: " + filled;

}

}

class Circle extends Shape {

private double radius;

public Circle(String color, boolean filled, double radius) {

super(color, filled);

this.radius = radius;

}

@Override

public String toString() {

return super.toString() + ", Radius: " + radius;

}

}

class Rectangle extends Shape {

private double width;

private double length;

public Rectangle(String color, boolean filled, double width, double length) {

super(color, filled);

this.width = width;

this.length = length;

}

@Override

public String toString() {

return super.toString() + ", Width: " + width + ", Length: " + length;

}

}

public class Exercise18 {

public static void main(String[] args) {

Circle circle = new Circle("Red", true, 5.0);

Rectangle rectangle = new Rectangle("Blue", false, 4.0, 6.0);

System.out.println("Circle Details: " + circle.toString());

System.out.println("Rectangle Details: " + rectangle.toString());

}

}

Exercise 19:

Create a superclass "Animal" with a method called "sound." Create two subclasses "Dog" and "Cat" that override the "sound" method to print different sounds for each animal.

class Animal {

public void sound() {

System.out.println("Animal makes a sound");

}

}

class Dog extends Animal {

@Override

public void sound() {

System.out.println("Dog barks");

}

}

class Cat extends Animal {

@Override

public void sound() {

System.out.println("Cat meows");

}

}

public class Exercise19 {

public static void main(String[] args) {

Animal animal1 = new Dog();

Animal animal2 = new Cat();

animal1.sound();

animal2.sound();

}

}

Exercise 20:

Create a superclass "Shape" with a method called "area." Create two subclasses "Circle" and "Rectangle" that override the "area" method to calculate the area for each shape.

class Shape {

public double area() {

return 0.0;

}

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI \* radius \* radius;

}

}

class Rectangle extends Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double area() {

return length \* width;

}

}

public class Exercise20 {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Area of Circle: " + circle.area());

System.out.println("Area of Rectangle: " + rectangle.area());

}

}