23MCA1030 **Vinayak Kumar Singh**

Exercise 5a Inheritance

1. Define a Vehicle class with attributes like make, model, and methods like start() and stop(). Then create subclasses like Car, Bicycle, and Motorcycle inheriting from Vehicle and add specific attributes and methods to each subclass.

Task of start()

Print the following statement
“Starting the” +make+ model

Example
“Starting the” Toyota Camry

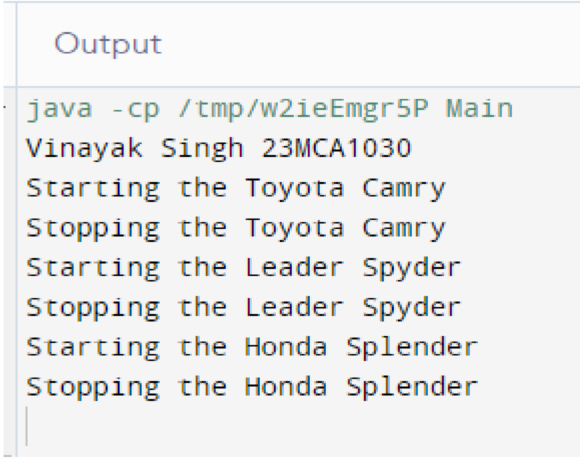
Task of stop()

Print the following statement
Stopping the Toyota Camry

Code:

```
class Vehicle {
    String make;
    String model;

    public Vehicle(String make, String model) {
        this.make = make;
        this.model = model;
    }

    public void start() {
        System.out.println("Starting the " + make + " " + model);
    }

    public void stop() {
        System.out.println("Stopping the " + make + " " + model);
    }
}
class Car extends Vehicle {
    public Car(String make, String model) {
        super(make, model);
    }
}
class Bicycle extends Vehicle {
    public Bicycle(String make, String model) {
        super(make, model);
    }
}
class Motorcycle extends Vehicle {
    public Motorcycle(String make, String model) {
        super(make, model);
    }
}
class Main {
    public static void main(String[] args) {
        System.out.println("Vinayak Singh 23MCA1030");
        Car car = new Car("Toyota", "Camry");
        car.start();
        car.stop();

        Bicycle bicycle = new Bicycle("Leader", "Spyder");
        bicycle.start();
        bicycle.stop();

        Motorcycle motorcycle = new Motorcycle("Honda", "Splender");
        motorcycle.start();
        motorcycle.stop();
    }
}
```

Output:



```
java -cp /tmp/w2ieEmgr5P Main
Vinayak Singh 23MCA1030
Starting the Toyota Camry
Stopping the Toyota Camry
Starting the Leader Spyder
Stopping the Leader Spyder
Starting the Honda Splender
Stopping the Honda Splender
```

2. Create a Shape class with attributes like color and methods like getArea(). Then create subclasses like Circle, Rectangle, and Triangle inheriting from Shape and implement the getArea() method for each subclass.

Expected Output:

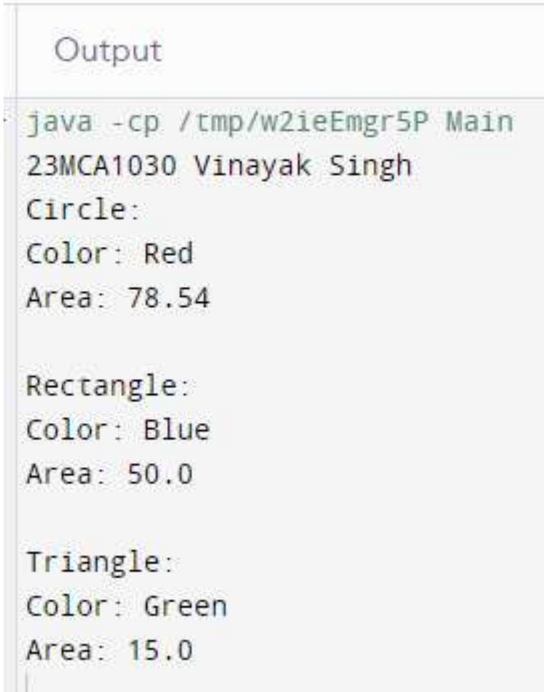
```
Circle:
Color: Red
Area: 78.54

Rectangle:
Color: Blue
Area: 50.0

Triangle:
Color: Green
Area: 15.0
```

Code:

```
class Shape {
    String color;
    public Shape(String color) {
        this.color = color;
    }
    public double getArea() {
        return 0;
    }
}
class Circle extends Shape {
    double radius;
    public Circle(String color, double radius) {
        super(color);
        this.radius = radius;
    }
    @Override
    public double getArea() {
        return Math.PI * radius * radius;
    }
}
```

```
class Rectangle extends Shape {
    double length;
    double width;
    public Rectangle(String color, double length, double width) {
        super(color);
        this.length = length;
        this.width = width;
    }
    @Override
    public double getArea() {
        return length * width;
    }
}
class Triangle extends Shape {
    double base;
    double height;
    public Triangle(String color, double base, double height) {
        super(color);
        this.base = base;
        this.height = height;
    }
    @Override
    public double getArea() {
        return 0.5 * base * height;
    }
}
class Main {
    public static void main(String[] args) {
        System.out.println("23MCA1030 Vinayak Singh");
        Circle circle = new Circle("Red", 5);
        System.out.println("Circle:");
        System.out.println("Color: " + circle.color);
        System.out.printf("Area: %.2f%n", circle.getArea());
        System.out.println();
        Rectangle rectangle = new Rectangle("Blue", 10, 5);
        System.out.println("Rectangle:");
        System.out.println("Color: " + rectangle.color);
        System.out.println("Area: " + rectangle.getArea());
        System.out.println();
        Triangle triangle = new Triangle("Green", 6, 5);
        System.out.println("Triangle:");
        System.out.println("Color: " + triangle.color);
        System.out.println("Area: " + triangle.getArea());
    }
}
```

Output:



```
java -cp /tmp/w2ieEmgr5P Main
23MCA1030 Vinayak Singh
Circle:
Color: Red
Area: 78.54

Rectangle:
Color: Blue
Area: 50.0

Triangle:
Color: Green
Area: 15.0
```

3. Create a base class Employee with attributes like name, id, and salary. Then create subclasses like Manager and Engineer inheriting from Employee and add specific attributes and methods to each subclass.

Expected Output:
Manager Details:
Name: John Doe
ID: M123
Salary: $80000
Department: Sales

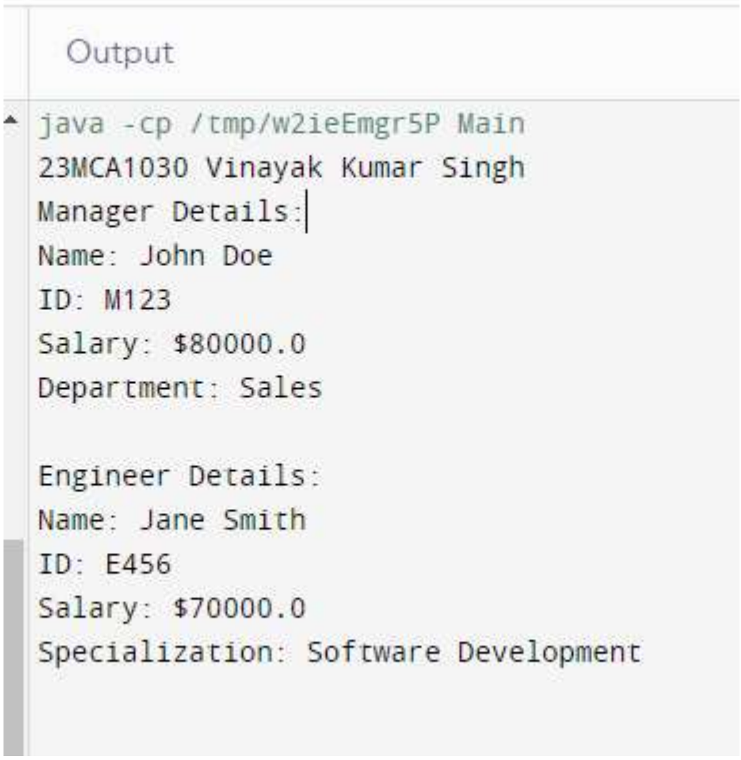
Engineer Details:
Name: Jane Smith
ID: E456
Salary: $70000
Specialization: Software Development

Code:

```
class Employee {
    String name;
    String id;
    double salary;
    public Employee(String name, String id, double salary) {
        this.name = name;
        this.id = id;
        this.salary = salary;
    }
}
class Manager extends Employee {
    String department;
    public Manager(String name, String id, double salary, String department) {
        super(name, id, salary);
        this.department = department;
    }
    public void displayDetails() {
        System.out.println("Manager Details:");
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Salary: $" + salary);
        System.out.println("Department: " + department);
    }
}
class Engineer extends Employee {
    String specialization;
    public Engineer(String name, String id, double salary, String specialization) {
        super(name, id, salary);
        this.specialization = specialization;
    }
    public void displayDetails() {
        System.out.println("Engineer Details:");
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Salary: $" + salary);
        System.out.println("Specialization: " + specialization);
    }
}
```

```
class Main {
    public static void main(String[] args) {
        System.out.println("23MCA1030 Vinayak Kumar Singh");
        Manager manager = new Manager("John Doe", "M123", 80000, "Sales");
        manager.displayDetails();
        System.out.println();
        Engineer engineer = new Engineer("Jane Smith", "E456", 70000, "Software
Development");
        engineer.displayDetails();
    }
}
```

Output:



```
java -cp /tmp/w2ieEmgr5P Main
23MCA1030 Vinayak Kumar Singh
Manager Details:
Name: John Doe
ID: M123
Salary: $80000.0
Department: Sales

Engineer Details:
Name: Jane Smith
ID: E456
Salary: $70000.0
Specialization: Software Development
```

4. Create a base class Animal with attributes like name and methods like makeSound(). Then create subclasses like Dog, Cat, and Bird inheriting from Animal and implement the makeSound() method for each subclass.

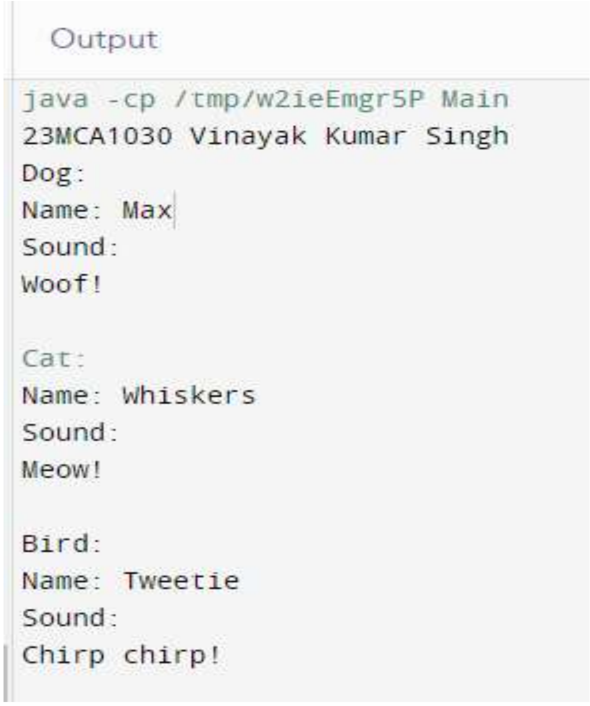
Expected Output

```
Dog:
Name: Max
Sound: Woof!

Cat:
Name: Whiskers
Sound: Meow!

Bird:
Name: Tweetie
Sound: Chirp chirp!
```

Code:

```
class Animal {
    String name;
    public Animal(String name) {
        this.name = name;
    }
    public void makeSound() {
        System.out.println("Unknown sound");
    }
}
class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }
    @Override
    public void makeSound() {
        System.out.println("Woof!");
    }
}
class Cat extends Animal {
    public Cat(String name) {
        super(name);
    }
    @Override
    public void makeSound() {
        System.out.println("Meow!");
    }
}
class Bird extends Animal {
    public Bird(String name) {
        super(name);
    }
    @Override
    public void makeSound() {
        System.out.println("Chirp chirp!");
    }
}
class Main {
    public static void main(String[] args) {
        System.out.println("23MCA1030 Vinayak Kumar Singh");
        Dog dog = new Dog("Max");
        System.out.println("Dog:");
        System.out.println("Name: " + dog.name);
        System.out.println("Sound: ");
        dog.makeSound();
        System.out.println();

        Cat cat = new Cat("Whiskers");
        System.out.println("Cat:");
        System.out.println("Name: " + cat.name);
        System.out.println("Sound: ");
        cat.makeSound();
        System.out.println();

        Bird bird = new Bird("Tweetie");
        System.out.println("Bird:");
        System.out.println("Name: " + bird.name);
        System.out.println("Sound: ");
        bird.makeSound();
        System.out.println();
    }
}
```

Output:



```
java -cp /tmp/w2ieEmgr5P Main
23MCA1030 Vinayak Kumar Singh
Dog:
Name: Max
Sound:
Woof!

Cat:
Name: Whiskers
Sound:
Meow!

Bird:
Name: Tweetie
Sound:
Chirp chirp!
```