**WEEK -4 TASK (OOPS CONCEPTS)**

**INHERITANCE:**

**SINGLE INHERITANCE:**

class Animal {

void sound() {

System.out.println("Animals make sounds");

}

}

class Dog extends Animal {

void sound() {

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

}

}

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

dog.sound();

}

}

**OUTPUT:**

Dog barks

****

**MULTILEVEL:**

class Animal {

void speak() {

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

}

}

class Dog extends Animal {

void bark() {

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

}

}

class Puppy extends Dog {

void play() {

System.out.println("Puppy plays");

}

}

public class Main {

public static void main(String[] args) {

Puppy pup = new Puppy();

pup.speak();

pup.bark();

pup.play();

}

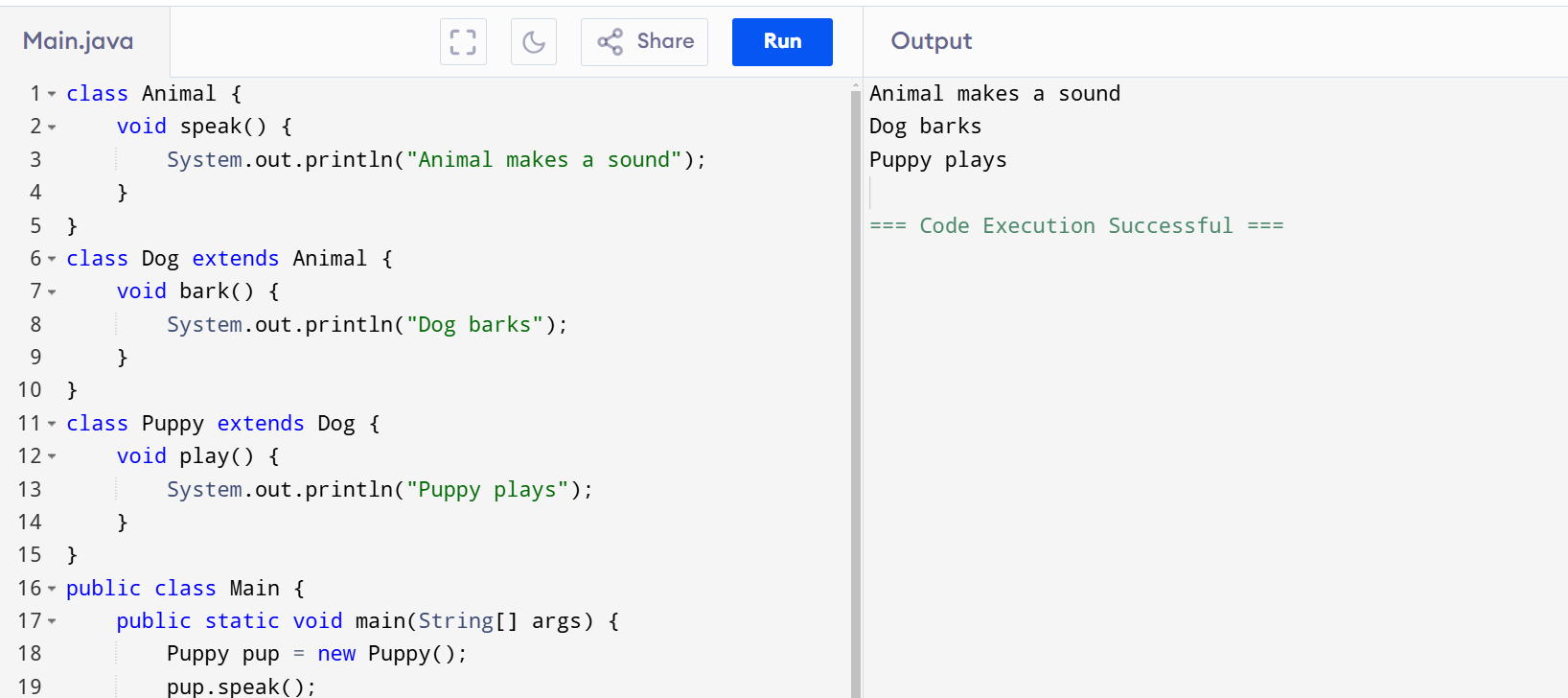
}

**OUTPUT:**

Animal makes a sound

Dog barks

Puppy plays



**HIERARCHICAL:**

class Animal {

void speak() {

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

}

}

class Dog extends Animal {

void bark() {

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

}

}

class Cat extends Animal {

void meow() {

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

}

}

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

Cat cat = new Cat();

dog.speak();

dog.bark();

cat.speak();

cat.meow();

}

}

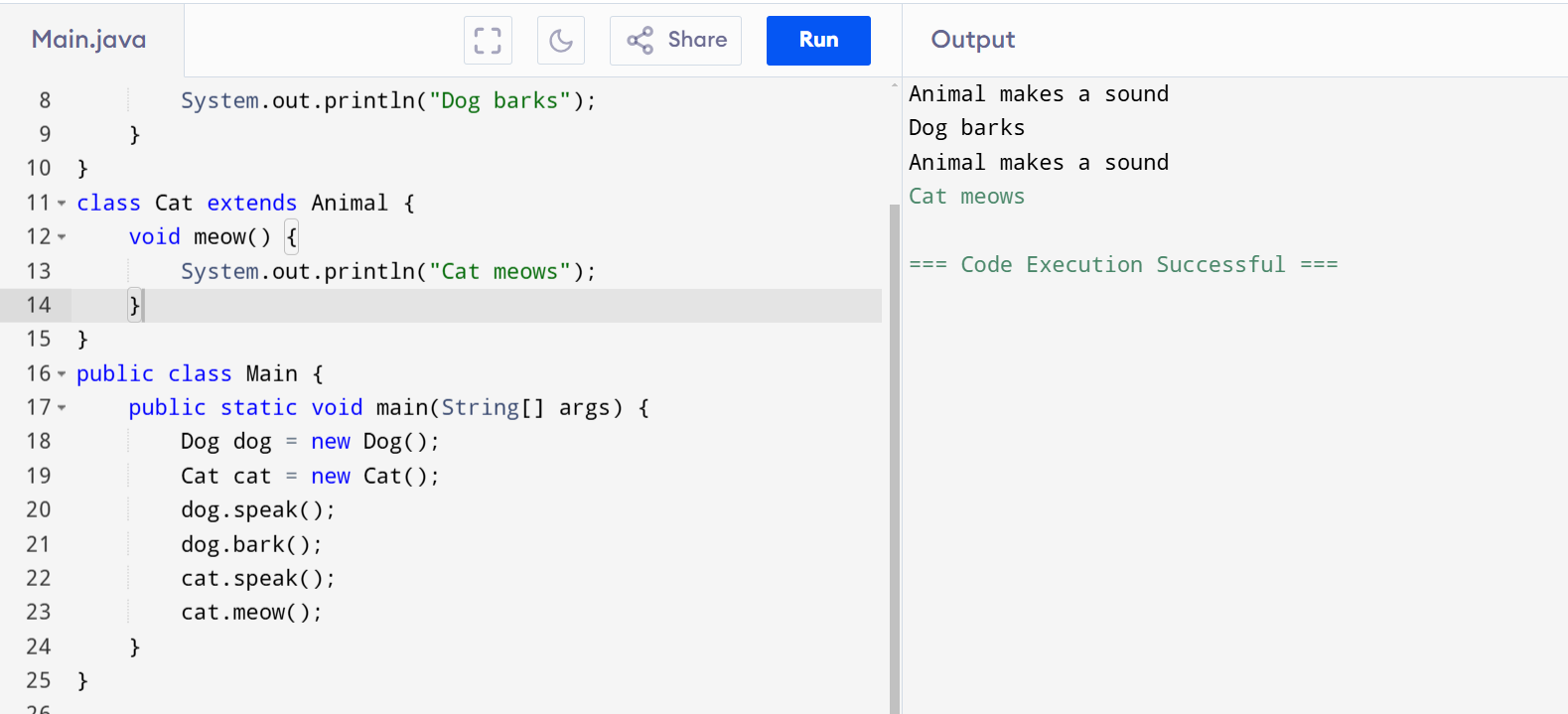
**OUTPUT:**

Animal makes a sound

Dog barks

Animal makes a sound

Cat meows



**HYBRID:**

class Animal {

void speak() {

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

}

}

class Dog extends Animal {

void bark() {

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

}

}

class Cat extends Animal {

void meow() {

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

}

}

class Puppy extends Dog {

void play() {

System.out.println("Puppy plays");

}

}

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

Cat cat = new Cat();

Puppy pup = new Puppy();

dog.speak();

dog.bark();

cat.speak();

cat.meow();

pup.speak();

pup.bark();

}

}

**OUTPUT:**

Animal makes a sound

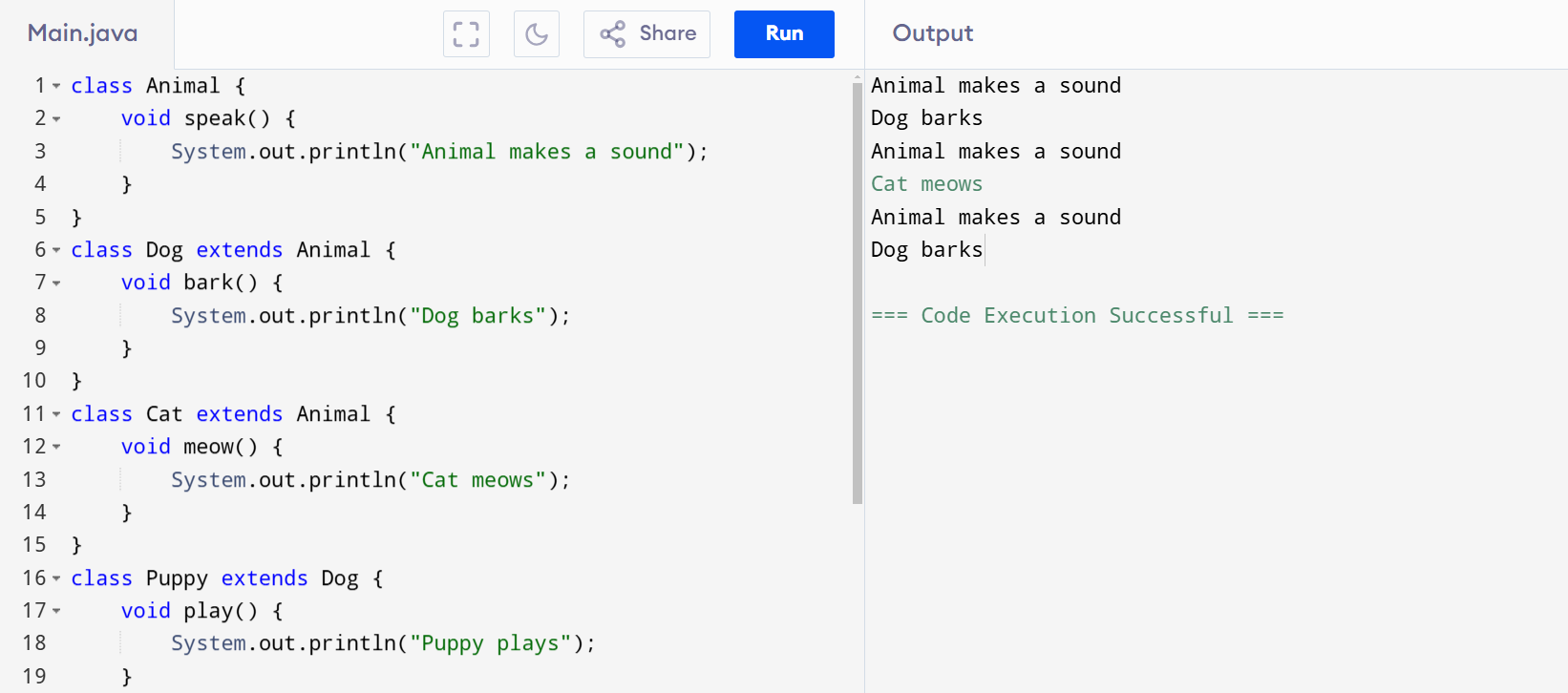
Dog barks

Animal makes a sound

Cat meows

Animal makes a sound

Dog barks



**ABSTRACTION:**

**Program**

abstract class Shape {

abstract double area();

void display() {

System.out.println("This is a shape.");

}

}

class Circle extends Shape {

double radius;

Circle(double radius) {

this.radius = radius;

}

double area() {

return Math.PI \* radius \* radius;

}

}

class Rectangle extends Shape {

double length, width;

Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

double area() {

return length \* width;

}

}

public class Main {

public static void main(String[] args) {

Shape circle = new Circle(5.0);

Shape rectangle = new Rectangle(4.0, 6.0);

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

circle.display();

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

rectangle.display();

}

}

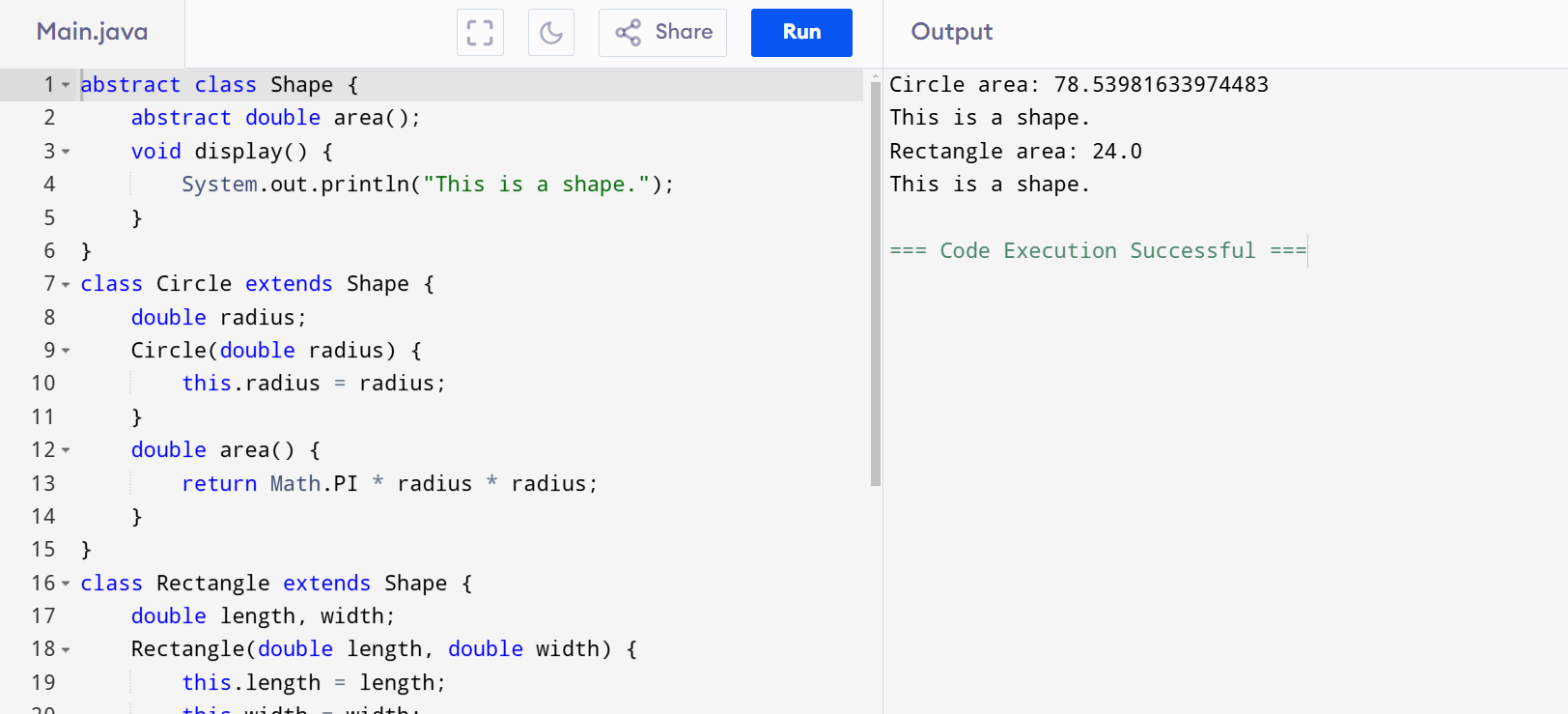
**Output**

Circle area: 78.53981633974483

This is a shape.

Rectangle area: 24.0

This is a shape.

****

**Polymorphism**

**METHOD OVERLOADING**

class Calculator {

int add(int a, int b) {

return a + b;

}

int add(int a, int b, int c) {

return a + b + c;

}

double add(double a, double b) {

return a + b;

}

double add(int a, double b) {

return a + b;

}

}

public class Main {

public static void main(String[] args) {

Calculator calc = new Calculator();

System.out.println("Sum of 10 and 20: " + calc.add(10, 20));

System.out.println("Sum of 10, 20, and 30: " + calc.add(10, 20, 30));

System.out.println("Sum of 10.5 and 20.5: " + calc.add(10.5, 20.5));

System.out.println("Sum of 10 and 20.5: " + calc.add(10, 20.5));

}

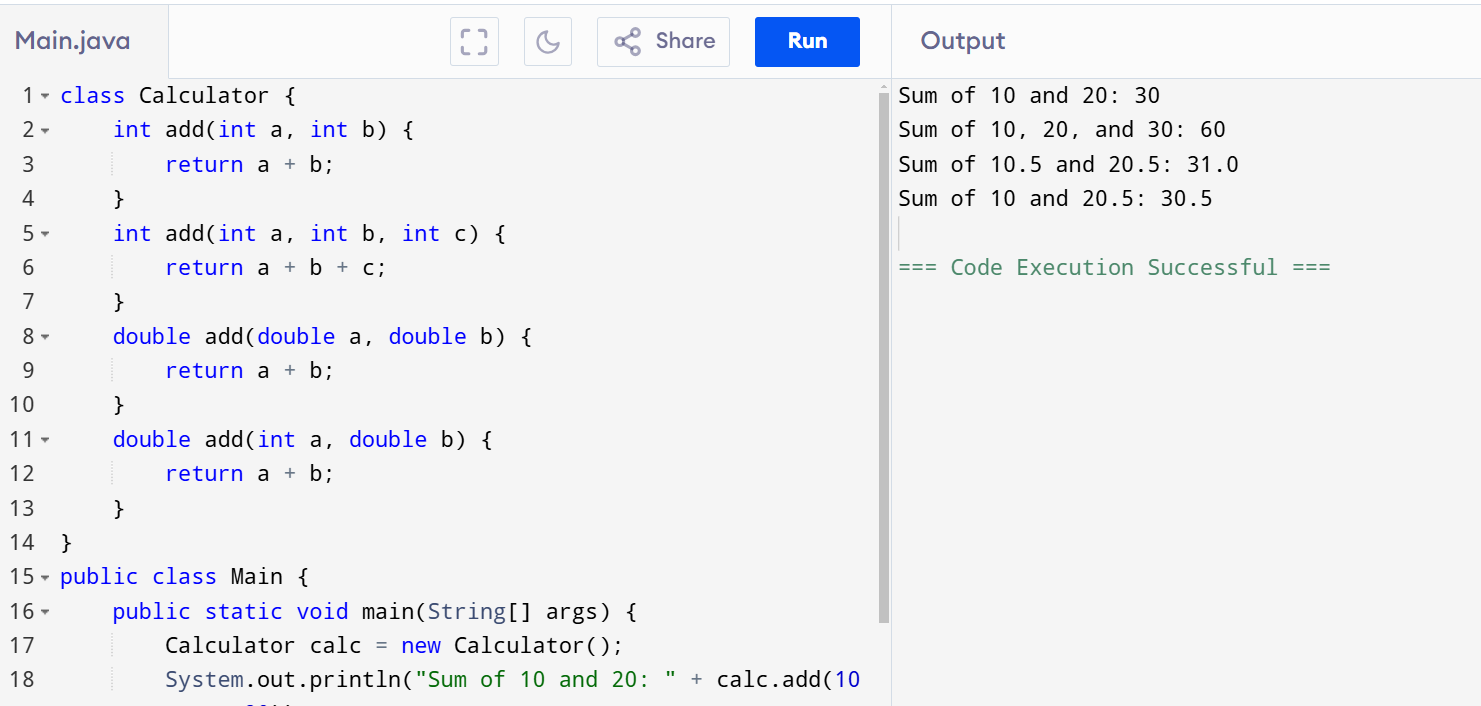
}

**OUTPUT**

Sum of 10 and 20: 30

Sum of 10, 20, and 30: 60

Sum of 10.5 and 20.5: 31.0

Sum of 10 and 20.5: 30.5

**METHOD OVERRIDING**

class Animal {

void sound() {

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

}

}

class Dog extends Animal {

@Override

void sound() {

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

}

}

class Cat extends Animal {

void sound() {

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

}

}

public class Main {

public static void main(String[] args) {

Animal animal = new Animal();

Animal dog = new Dog();

Animal cat = new Cat();

animal.sound();

dog.sound();

cat.sound();

}

}

**OUTPUT**

Animal makes a sound

Dog barks

Cat meows

