Constructor Overloading

```
public class Doggy {
    private String name;
    private String breed;
    private int age;

    // Constructor for fully known dog details
    public Doggy(String name, String breed, int age) {
        this.name = name;
        this.breed = breed;
        this.age = age;
    }

    // Constructor when only the name is known
    public Doggy(String name) {
        this(name, "", 0); // Calls the first constructor with default values for breed and age
    }

    // Constructor when the name and age are known, but the breed isn't
    public Doggy(String name, int age) {
        this(name, "", age); // Calls the first constructor with a default breed
    }

    // Method to print dog details
    public void printDetails() {
        System.out.println("Name: " + this.name + ", Breed: " + this.breed + ", Age: " + this.age);
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        // Create a Doggy object with full details
        Doggy dog1 = new Doggy("Buddy", "Golden Retriever", 3);
        dog1.printDetails();

        // Create a Doggy object with only the name known
        Doggy dog2 = new Doggy("Shadow");
        dog2.printDetails();

        // Create a Doggy object with the name and age known, but no breed
        Doggy dog3 = new Doggy("Rex", 2);
        dog3.printDetails();
    }
}
```

OUTPUT

```
Name: Buddy, Breed: Golden Retriever, Age: 3
Name: Shadow, Breed: , Age: 0
Name: Rex, Breed: , Age: 2
```