1. Introducción Teórica

Polimorfismo: Es la capacidad que tiene un lenguaje de programación para que un mismo objeto pueda tener distintas formas.
Herencia: Es un mecanismo que permite a una clase (clase derivada) heredar las características (atributos y métodos) de otra clase (clase base). En otras palabras, permite crear clases a partir de otras ya existentes.
Sobrecarga de métodos: Es la posibilidad de tener dos o más métodos con el mismo nombre pero con diferentes parámetros en una misma clase.
Polimorfismo paramétrico: Permite definir clases y métodos que pueden trabajar con diferentes tipos de datos sin necesidad de modificar el código.
Polimorfismo de inclusión: Permite que un objeto de una clase derivada pueda ser utilizado como si fuera un objeto de su clase base.

2. Ejemplos de Código

Herencia:

```
class Animal { //clase base Animal
        void emiteSonido() { // método para emitir un sonido para cada animal
            System.out.println("Este animal hace un sonido");
        }
    }
    //clase Perro que hereda de Animal y sobrescribe el método emitirSonido
    class Perro extends Animal {
        @Override
        void emiteSonido() {
            System.out.println("El perro ladra");
        }
    }
    //clase Gato que hereda de Animal y sobrescribe el método emitirSonido
    class Gato extends Animal {
        @Override
        void emiteSonido() {
            System.out.println("El gato maulla");
        }
    }
```

Polimorfismo de inclusión:

```
public class PolimorfismoInclusion {
        public static void main(String[] args) {
            Animal miAnimal = new Perro(); //crea un objeto Perro
            miAnimal.emiteSonido();  //ejecuta el método emitir sonido
            miAnimal = new Gato(); //cambia la referencia a Gato
            miAnimal.emiteSonido();  //ejecuta el método emitir sonido
        }
    }
```

Sobrecarga:

```
//Dos métodos sumar con diferentes tipos de parámetros
class Sobrecarga {
int sumar(int a, int b) {
  return a + b;
}
double sumar(double a, double b) {
  return a + b;
}
}
```

Polimorfismo paramétrico:

```
class Contenedor<T> {
        private T contenido;
    // constructor que acepta un tipo genérico T como parámetro
        public Contenedor(T contenido) {
            this.contenido = contenido;
        }
        public T obtenerContenido() { // método para devolver el contenido del
contenedor
            return contenido;
        }
    }
```

Polimorfismo:

```
class Figura { //clase base figura
        void dibujar() {
            System.out.println("Dibujando una figura");
        }
    }
    //clase Circulo que hereda de Figura y sobrescribe el método dibujar
    class Circulo extends Figura {
        @Override
        void dibujar() {
            System.out.println("Dibujando un círculo");
        }
    }
    public class Polimorfismo {
        public static void main(String[] args) {
            Figura figura = new Circulo(); //crea un objeto Circulo
            figura.dibujar();  //se ejecuta el método dibujar
        }
    }
```

3. Análisis Comparativo

- **Explicar las diferencias entre polimorfismo y sobrecarga de métodos**

  El polimorfismo permite usar un mismo nombre para diferentes acciones, mientras que la sobrecarga de métodos permite realizar operaciones diferentes con el mismo nombre, pero con diferentes parámetros.

- **Diferenciar entre sobrecarga (overloading) y redefinición (overriding) de métodos**

  La sobrecarga permite definir varios métodos con el mismo nombre en una misma clase pero con diferentes parámetros, mientras que la redefinición permite crear diferentes comportamientos para un mismo método en diferentes clases con la misma firma.

- Preguntas

- **¿Qué es el término firma?**

  La firma de un método está formada por el nombre del método y la manera específica en que recibe información (parámetros) y devuelve resultados. La firma ayuda al programa a entender que versión del método tiene que usar según cómo se le llame.

- **¿Diferencias entre los términos Overloading y Overriding?**

  Overloading se refiere a tener varios métodos con el mismo nombre en la misma clase pero con diferentes firmas, mientras que Overriding se refiere a implementar un método en una clase hija con la misma firma que el método en clase padre.

- **¿Se pueden sobrecargar métodos estáticos?**

  Sí, se pueden tener métodos estáticos con el mismo nombre pero con diferentes parámetros. Sin embargo, no pueden ser redefinidos.

- **¿Es posible sobrecargar la clase main() en Java?**

  Sí, se puede sobrecargar el método main mientras que los parámetros sean diferentes.