

Problem 1: Differences and Similarities of Interfaces, Abstract Classes, and Classes

Interfaces

- Define a contract with abstract methods (default `public` and `abstract`).
- Support **multiple inheritance**.
- Fields are `public static final` (constants).
- No constructors; cannot hold state.

Example:

```
public interface Vehicle {  
    void start();  
    void stop();  
}
```

Abstract Classes

- Partially implemented base class with abstract and concrete methods.
- Supports **single inheritance**.
- Can have instance variables and constructors.

Example:

```
public abstract class Animal {  
    public abstract void sound();  
    public void sleep() {  
        System.out.println("Sleeping...");  
    }  
}
```

Classes

- Fully implemented blueprint for objects.
- Supports **single inheritance**.
- Can be instantiated.

Example:

```

public class Dog extends Animal {
    @Override
    public void sound() {
        System.out.println("Bark!");
    }
}

```

Key Differences

- Interfaces support multiple inheritance; abstract classes and classes do not.
- Abstract classes and classes can hold state; interfaces cannot.

Key Similarities

- Both interfaces and abstract classes enforce structure for subclasses.
- Both can define reusable methods.

Problem 2: Concept of Inheritance

- Inheritance allows a class to acquire the properties and methods of another class.
- Promotes code reuse and hierarchical classification.
- The `extends` keyword is used for classes, and `implements` for interfaces.

Example:

```

public class Parent {
    public void greet() {
        System.out.println("Hello from Parent!");
    }
}

public class Child extends Parent {
    public void greet() {
        System.out.println("Hello from Child!");
    }
}

```

Here, Child inherits methods from Parent but can override them.

Key Points

- Inheritance supports polymorphism, where a subclass can override methods of a superclass.
- A class cannot inherit from multiple classes in Java (single inheritance restriction).

Problem 3: Questions on Inheritance

Interface Inheritance

- Can an interface inherit from another interface? **Yes.**
- Can an interface inherit from multiple interfaces? **Yes.**
- Can an interface inherit from a class? **No.**

Example:

```
public interface A { void methodA(); }
public interface B extends A { void methodB(); }
public interface C extends A, B { void methodC(); }
```

Class Inheritance

- Can a class inherit from an interface? **No, it implements an interface.**
- Can a class inherit from multiple interfaces? **Yes.**
- Can a class inherit from a class? **Yes.**
- Can a class inherit from multiple classes? **No.**

Example:

```
public interface A { void methodA(); }
public interface B { void methodB(); }
public class C implements A, B {
    public void methodA() { }
    public void methodB() { }
}
```