HW1_David_Mugwaneza

The objective of this assignment is for students to explore and understand the differences between Java interfaces and abstract classes. Students should analyze the characteristics of interface and abstract class and demonstrate their understanding through Java code examples.

Explore java interface and abstract class, then answer the below questions.

1. What is the key difference between an interface and an abstract class? An interface is used to define a contract for classes to implement, with the methods having to all be abstract while an abstract class can provide both abstract and concrete methods and are typically used when classes share some functionality but can also require additional functionality to be defined by subclasses.
2. When would you prefer using an interface over an abstract class, and vice versa?
   - **Use an Interface:** When you want to define a contract that multiple classes can implement.
   - **Use an Abstract Class:** When you need to share some code among subclasses or when subclasses have common behavior.
3. How does the use of multiple inheritance differ between interfaces and abstract classes? Java does not support multiple inheritance with classes so a class can only extend one abstract class. But Java supports multiple inheritance through interfaces so a class can implement multiple interfaces.
4. Write Java code to demonstrate your answer, use comments to explain how each part demonstrates the characteristics of interfaces and abstract classes.
   - **Interfaces Flyable and Swimmable:** Contracts for classes that can fly or swim.

```java
// Interface for ability to fly
public interface Flyable {
    void fly();
}
```

```java
// Interface for ability to swim
public interface Swimmable {
    void swim();
}
```

   - **Abstract Class Animal:** The base class for all animals with a shared sleep() method (concrete method) and an abstract sound() method that must be implemented by subclasses.

```java
// Abstract class represneting general chractersitics of an animal
public abstract class Animal {
    String name;

    public Animal(String name) {
        this.name = name;
    }

    // Abstract method for the sound an animal makes
    public abstract void sound();

    // Concrete method for sleeping
    public void sleep() {
        System.out.println(name + " is sleeping.");
    }
}
```

- **Concrete Class Duck:** Inherits abstract Animal class and implements Flyable and Swimmable interfaces demonstrating multiple inheritance with interfaces.

```java
// Duck class extends Animal and implements Flyable and Swimmable
public class Duck extends Animal implements Flyable, Swimmable {
    public Duck(String name) {
        super(name);
    }

    // Implemeting the sound method from Animal
    @Override
    public void sound() {
        System.out.println(name + " says: Quack!");
    }

    // Implementing the fly method from Flyable
    @Override
    public void fly() {
        System.out.println(name + " is flying.");
    }

    // Implementing the swim method from Swimmable
    @Override
    public void swim() {
        System.out.println(name + " is swimming.");
    }
}
```