1. (2 pts) What happens when you try to compile and execute the following program? Explain your answer.

```java
public abstract class Question1 implements Comparable {
    private int num;
    public Question1(int num) {
        this.num = num;
    }
    public int compareTo(Object o) {
        return(-1);
    }
    public static void main(String[] args) {
        Question1 question1 = new Question1(1);
        Question1 question2 = new Question1(2);
        System.out.println(question1.compareTo(question2));
    }
}
```

There will be a compile-time error. Since the class is abstract, we cannot create an instance of it.

2. (2 pts) Explain the interface contract.

If a class definition implements an interface, it is required to implement all abstract methods listed in the interface. In return, the class definition has the “is a” relationship with the interface.

3. (2 pts) Suppose class A extends B and implements C. If neither A nor B have constructors and B does not implement C, which of the following are correct assignments? Explain your reasoning.

- A a = new B(); Incorrect. B is the superclass so an instance of B cannot be referred to as an A.
- C c = new B(); Incorrect. B has no relationship with C so an instance of B cannot be referred to as a C.
- C c = new A(); Correct. Since A implements C, it has the “is a” relationship with C so an A can be referred to as a C.
- B b = new A(); Correct. Since A is a subclass of B, it has the “is a” relationship with B so an A can be referred to as a B.

4. (2 pts) Autoboxing is a feature which allows a primitive type to be automatically encapsulated into its corresponding wrapper, and autounboxing is a feature which allows an instance of a wrapper class to be automatically converted to the corresponding primitive type.

5. (2 pts) Components in the java.awt package are considered heavyweight components since they rely heavily on the platform on which Java is running. Most of the components in the javax.swing package are considered lightweight since they do not rely heavily on the underlying platform.