1. (2 pts) Suppose A extends B and B extends C. In the following code, will there be a compile-time error or a run-time error? Explain.

```java
B b = new B();
Object o = (Object)b;
A a = (A)o;
C c = (C)o;
```

There will not be a compile-time error because each cast is to a superclass or subclass type. However, there will be a run-time error because the object that is created is of type B. So when we try to cast the compile-time type of the reference to A, there will be a ClassCastException.

2. (2 pts) What is printed when Question2 is executed? Explain the output.

```java
public class Question2 {
    class B {
        protected int num = 2;
        public static void print() { System.out.println("B"); }
        public boolean isPrimitiveRoot() { return(false); }
    }
    class A extends B {
        protected int num = 1;
        public static void print() { System.out.println("A"); }
        public boolean isPrimitiveRoot() { return(true); }
    }
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        a.print();
        System.out.println(a.isPrimitiveRoot());
        System.out.println(a.num);
        b.print();
        System.out.println(b.isPrimitiveRoot());
        System.out.println(b.num);
    }
}
```

Recall that there is dynamic dispatch or polymorphism with overridden instance methods, but not with instance variables or hidden class methods.

A true
B false

3. (2 pts) A class definition that is declared abstract cannot be used to create an instance. A class definition must be concrete before we can use it to create an instance.

4. (2 pts) If a class definition implements an interface, then a contract is established between the class definition and the interface. What is the class definition required to do to honor the contract, and what ability does the class definition receive by honoring the contract? (You may use the back to answer.)

The class definition must give implementations to all of the abstract methods it inherits from the interface.
The “is a” relationship will exist between the class definition and the interface.
5. (2 pts) Suppose we have the following declarations. Are there any syntax errors when we compile Test.java? If not, what is printed?

```java
public interface Feasible {
    public void feaseIt(Object o);
    public boolean canBeFeased();
}

public class Test implements Feasible {
    public void feaseIt() {
        System.out.println("We feased it");
    }
    public boolean canBeFeased() {
        return(true);
    }
    public static void main(String[] args) {
        Test test = new Test();
        test.feaseIt();
    }
}
```

There will be a compile-time error when Test.java is compiled. Test.java inherits the method public void feaseIt(Object o) from the interface, but doesn’t give it an implementation.