

Question 1. [9 MARKS]

The following code compiles without errors:

```
public class A {
    int num = 13;

    public A() {
        System.out.println("one");
    }

    public void report() {
        System.out.println("two");
    }
}

public class B extends A {
    int num = 2;

    public B() {
        System.out.println("here");
    }
}

public class C extends B {
    public C() {
        System.out.println("some");
    }

    public void report() {
        System.out.println("all");
    }
}
```

Part (a) [5 MARKS]

Suppose we have a main method in another class that says: `A var1 = new C();`

This compiles and runs without error.

- What output will be created by constructing that instance of `C`? (Line breaks are omitted below to save space.)
 - some here one
 - one here some
 - some
- If the expression `var1.num` is then used, which variable is accessed?
 - The instance variable `num` in class `A`.
 - The instance variable `num` in class `B`.
 - The instance variable `num` in class `C`.
 - None of the above; this expression is illegal.
- If the expression `((C) var1).num` is then used, which variable is accessed?
 - The instance variable `num` in class `A`.
 - The instance variable `num` in class `B`.
 - The instance variable `num` in class `C`.
 - None of the above; this expression is illegal.
- If the the method call `var1.report()` is then used, which method is called?
 - The method `report` in class `A`.
 - The method `report` in class `B`.
 - The method `report` in class `C`.
 - None of the above; this method call is illegal.
- If the the method call `((B) var1).report()` is then used, which method is called?
 - The method `report` in class `A`.
 - The method `report` in class `B`.
 - The method `report` in class `C`.
 - None of the above; this method call is illegal.

Part (b) [4 MARKS]

Define a public interface called `Reportable` that imposes just one obligation on those classes that implement it: they must have a method called `report` with the same signature as the one in class `A`.

(*I.e.*, Their `report` method must have the same accessibility, return type, name and parameters.)

```
public interface Reportable {  
  
    public void report();  
  
}
```

Below, modify class `A` so that instances of it could be used anywhere a `Reportable` object is required.

The only change required is to say that `A` "implements `Reportable`":

```
public class A implements Reportable {  
  
    int num = 13;  
  
    public A() {  
        System.out.println("one");  
    }  
  
    public void report() {  
        System.out.println("two");  
    }  
}
```

Question 2. [5 MARKS]**Part (a)** [1 MARK]

Suppose we are writing a program that will involve Sneetches and Smurfs. If I tell you that every Sneetch is a Smurf, which design makes the most sense?

- Class Sneetch is a parent of class Smurf.
- Class Smurf is a parent of class Sneetch.
- Classes Sneetch and Smurf are both children of a common parent class.
- A common child class has both class Sneetch and class Smurf as parents.

Part (b) [1 MARK]

For any class that doesn't have a no-argument constructor, Java will define one.

- True False

Part (c) [1 MARK]

If a class has any methods that are not abstract, the class must not be declared to be abstract.

- True False

Part (d) [1 MARK]

An abstract class can be instantiated as long as it has at least one method that is not abstract.

- True False

Part (e) [1 MARK]

Suppose class C2 is a child of class C1. The private members of a class C1 can be accessed from within class C2.

- True False