Programming Methods & Java Sample Tripos Questions

Dr Robert Harle

1. (a) Define what is meant by a class, an abstract class, an interface and a final class in Java. [4]

(b) How many times will the for loop in the following code execute? [3]

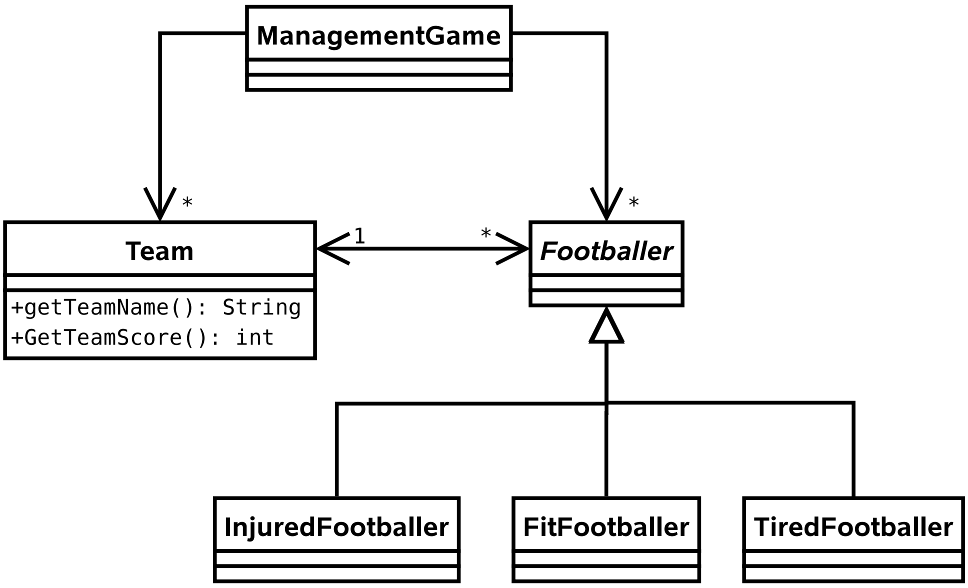
```
for (int i=Integer.MAX_VALUE-17; i<=Integer.MAX_VALUE; i++) {
  ...
}
```

(c) What is a design pattern? [2]

(d) What is the goal of the Decorator pattern? Draw a UML diagram that demonstrates the pattern, and give an example of its use. [6]

(e) A developer suggests that both the Decorator and the Proxy design patterns are about providing indirect access to objects through an intermediary and that they should be considered as one pattern. Comment on this. [5]

2. A computer game allow players to try their hand at managing a virtual football team. The program is structured as shown in the UML class diagram below (not all member functions and variables are shown for clarity):



(a) Give three differences between an abstract class and an interface in Java. Which would you expect Footballer to be and why? [5]

(b) Consider the design of Footballer and its classes. Why is this design inappropriate? [3]

(c) Suggest a design pattern that would be more appropriate to apply to Footballer. Sketch the resultant UML class diagram. [4]

(d) The ManagementGame must store the collection of teams, sorted alphabetically by team name. Suggest how you would do this using the Java class library and without implementing any extra interfaces. [3]

(e) The ManagementGame must *also* store the teams ordered by their score, highest first. Teams with the same score should appear in alphabetical order. Describe how you would achieve this. You may now use an appropriate interface from the library if you wish. [5]

3. (a) What is the decimal number 63 in binary (base-2), hexadecimal (base-16), and base-17? [4]

(b) What will be produced by this code: [3]

```
    int i=-1;
    for (int j=0; j<3; j++) {
i = i >>> 2;
i |= (1<<31);
i = i << 2;
    }
    System.out.println(i);
```

(c) Explain the difference between a shallow copy and a deep copy of an object, giving examples in Java where appropriate. [4]

(d) Comment on the implementation of the following class. Assume that a deep copy is required. Provide corrections for any problems or design faults that you describe. [9]

```
import java.util.LinkedList;

public class MyClass extends OtherClass implements Cloneable {
   public int[] intArray = new int[100];
   public LinkedList<AnotherClass> list = null;

   public Object clone() throws CloneNotSupportedException {
       MyClass m = new MyClass();
       m.intArray = intArray;
       m.list = list;
       return m;
   }

}
```