University of Glasgow

Monday 27 April 2015
9.30 am – 11.30 am
(2 hours)

DEGREES OF MSc in Information Technology, MSc in Software Development, and Designated Degree in Computing Science

ADVANCED PROGRAMMING (M)

Answer all questions.

This examination paper is worth a total of 75 marks.

INSTRUCTIONS TO INVIGILATORS: Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

1. Examine the following code snippet that shows two ways of creating an `ArrayList` object:

```
ArrayList<String> a = new ArrayList<String>();
List<String> b = new ArrayList<String>();
```

(a) Describe when you might use each one over the other.

[4]

(b) What is the role of `<String>`?

[2]

Consider the following code:

```
public class ExamQuestion {
        public static void modify(String in) {
                in += "exam";
        }
        public static void main(String[] args) {
                String s = "lovelydayforan";
                modify(s);
                System.out.println(s); // Point A
        }
}
```

(c) State the console output produced by the line labeled ‘Point A’ if this code was compiled and run. Explain why this output is produced with reference to how Java handles objects and the Immutable property of String objects.

[4]

(d) Explain why the `modify` method has to be declared static, sketching the changes in `main` that would be required to get the same behaviour if it was not declared static.

[4]

(e) Classes, methods and attributes can all be declared with the `final` modifier. What is the implication in each case?

[3]

2. **(a)** Outside of Swing, what are the two ways Java gives you to create objects that can be run in a thread?

[2]

(b) State why multi-threading is useful in i) GUI design, and ii) high performance computing.

[2]

The following code declares a class called `OneDimMover` that holds a position that can be changed by other objects (through `get` and `set` methods). In the main loop, I create 100 threads, each of which repeatedly changes the position in a particular way a certain number of times. The total net change on the position should be zero.

```
public class OneDimMover {
        private int pos = 0;
        public void setPos(int pos) { this.pos = pos;}
        public int getPos() { return pos;}

        public static class Changer implements Runnable {
                private int change,nChanges;
                private OneDimMover m;
                public Changer(int change,int nChanges,OneDimMover m) {
                        this.change = change;
                        this.nChanges = nChanges;
                        this.m = m;
                }
                public void run() {
                        for(int i=0;i<nChanges;i++) {
                                int currentPos = m.getPos();
                                currentPos += change;
                                m.setPos(currentPos);
                        }
                }
        }

        public static void main(String[] args) {
                OneDimMover odm = new OneDimMover();
                Thread[] t = new Thread[100];
                for(int i=0;i<50;i++) {
                        t[i] = new Thread(new Changer(1,200,odm));
                        t[i+50] = new Thread(new Changer(-2,100,odm));
                }
                for(int i=0;i<100;i++) { t[i].start();}
                try {
                        for(int i=0;i<100;i++){ t[i].join();} // A
                }catch(Exception e) {}
                System.out.println(odm.getPos());
        }
}
```

(c) What is the role of the `Thread.join` method (labeled ‘A’)?

[2]

(d) Why do we need the `try` block around the `Thread.join` method call?

[2]

(e) When I run the code, I don't get an output of 0. State what this behaviour is an example of and explain (with a diagram if necessary) why it happens in this example.

[6]

(f) Provide the code necessary to remove this problem. Only include code blocks that are changed – I will assume that anything not included in your answer is the same as in the code above.

[6]

3. The Turing Test is a procedure invented by Alan Turing for testing whether or not an artificial intelligence system has human level intelligence. During the test, a human operator asks the system questions (in the form of written questions) and the system replies with answers. You have been asked to build a distributed 'Turing Test' system. Artificial Intelligence researchers will encapsulate their algorithms into classes that implement an Abstract Class called `Intelligent` (defined below). You will have to build a Server that takes an `Intelligent` object and allows a client system, once connected, to send questions (in the form of Strings ending with a new line character) that, when received by the server are passed to the `Intelligent` object that returns an answer that should be sent back to the client as a string terminating with a new line character. This process continues until the client systems sends the string TERMINATE followed by a new line character.

```
public abstract class Intelligent {
        public abstract String askQuestion(String question);
}
```

(a) Design a suitable protocol for the system.

[4]

(b) Write the code for a server (your class can just have all its code inside a `main` method if you like). You do not need to include error handling and you do not need to use threads. Once the client has terminated, the server stops.

[4]

(c) Write the code for a client (your class can just have all its code inside a `main` method if you like). Your client should prompt the user for the question and provide the response on standard output.

[4]

(d) Why would it be sensible to make your served multi-threaded?

[2]

(e) Sketch how you would change your server code to support multi-threading. You don't need to write the code, but you can use code fragments in your answer if you like.

[6]

4. A musical instrument shop has a computer system to manage the instrument types that they stock. For historical reasons, the classes representing these instruments are not organised into a class hierarchy and typically contain different methods and attributes. Skeleton classes for trumpet and violin are provided below. The shop would like to add additional functionality to all instruments with *minimal* modifications to the instrument classes. In particular, they would like to add `displayInformation()` and `computeHirePrice()` methods for all instruments.

```
public class Trumpet {
        private int price;
        private int stockLevel;
        public int getPrice() { return price; }
}

public class Violin {
        private int price;
        private String manufacturer;
        private String recommendedStrings;
        private int stockLevel;
        public int getPrice() { return price; }
}
```

(a) Describe the main components of the *visitor* design pattern and what benefits would be conveyed through the use of the visitor *pattern* in this application.

[6]

(b) Design the system using the visitor design pattern. Provide all class and method definitions, but you don't need to populate the methods with code. Make sure you include how the current classes need to be modified.

[6]

(c) Which other design pattern would be sensible to use for this application?

[2]

(d) Discuss the advantages and disadvantages of the visitor pattern over your answer to part (c) within this particular context.

[4]