CS_211. Programming with Objects and Threads

(Attempt 2 questions out of 3)

Question 1

(a) Explain, using code fragments where necessary, the two different ways of creating and running a thread in Java. Your answer should include an explanation of why you would choose one method over the other.

[8 marks]

(b) Give a detailed explanation of the responsibilities of a layout manager for an application with a GUI interface.

[5 marks]

(c) Give the general format of a try-catch-finally clause. Explain what code you would put in each block and explain when, and in what order, the code is executed.

[9 marks]

(d) What are the only types in Java which are **not** classes?

[3 marks]

Question 2

(a) Explain, using example code, what is meant by **thread synchronisation**. Your answer should explain why this is necessary and how, in general, it is implemented in Java.

[8 marks]

(b) The “Gang of Four” describe the intent of the **Facade** pattern as being to “Provide a unified interface to a set of interfaces in a subsystem”. Explain the purpose and features of the facade pattern. Your answer should include a diagram showing how the main contributors to the facade pattern interact.

[8 marks]

(c) C++ allows multiple inheritance. Explain the problems which arise from this with regard to **diamond** inheritance and show how C++ defines which, of two identically defined inherited methods, is being invoked.

[9 marks]

Question 3

(a) A resource shared by two or more threads suggests the use of which design pattern? Explain why this is so and, using example code, how this design pattern can be implemented.

[10 marks]

(b) Describe, using the format introduced by the "Gang of Four" the main features of one of the following design patterns.

1. Strategy
2. Bridge
3. Decorator

Your answer should include the following: **Intent**, **Problem**, **Solution**, **Participants** and **Implementation**.

[12 marks]

(c) Explain the semantic error in the following code:

```
try{
   //Protected code.
}
catch(Exception e){
   //Exception handling code.
}
catch(IOException e){
   //Exception handling code.
}
```

[3 marks]