University of Glasgow

Monday 23 April 2018
9.30 am – 11.30 am
(Duration: 2 hours)

DEGREES of MSc Information Technology, MSc in Software Development

Advanced Programming (IT)

(Answer All Questions)

This examination paper is worth a total of 80 marks

The use of a calculator is not permitted in this examination

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. **(a)** Explain why it is important that computationally intensive operations are not placed on Java's event dispatch thread.

[3 marks]

(b) Describe how computationally intensive tasks should be implemented and initiated within Java's Swing framework.

[4 marks]

(c) Often it is useful for the user to be able to observe the progress of an intensive task. How should this be accomplished within Java Swing?

[5 marks]

(d) In the more general context of threading, describe what is meant by a **blocking method**, giving an example of one, and state what has to be done when using a blocking method.

[3 marks]

(e) In multi-threaded applications, it is easy to create **race conditions**. Describe a race condition that you have observed, explaining how the race condition happens, the implications to the output of the program and how it would be fixed. You can include code snippets / diagrams if you like, but this is not necessary.

[5 marks]

2. **(a)** Java is an object-oriented language. One of the key benefits of object-oriented software is the ability to use polymorphism. By sketching out a simple class hierarchy (in text or UML as you prefer), explain what is meant by polymorphism and describe why this is such a useful property.

[6 marks]

(b) **Interfaces** are common in Java applications. Describe what an interface is and give an example of an interface that you have used.

[3 marks]

(c) Describe how **abstract classes** differ from interfaces, and describe an example where you would use an abstract class over an interface with justification for your choice.

[4 marks]

(d) The following code is a copy of Java's Comparable interface:

```
public interface Comparable<T> {

   public int compareTo(T o);

}
```

What is 'T' and why is it there?

[2 marks]

(e) Write the source code for a class that will represent a University course, with an attribute to represent the number of students on the course. Your class should implement the comparable interface such that if objects are sorted using Arrays.sort they are ordered in **descending** order of class size. A course should also have a name and a toString method to display its name and number of students.

[5 marks]

3. **(a)** The **composite** design pattern consists of three classes:

i) component

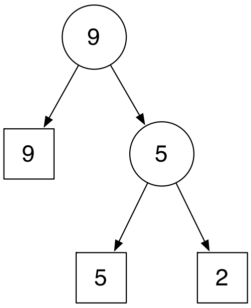
ii) leaf

iii) composite

Describe the role of each including what type of component it is (class, abstract class, interface etc) and the relationships between them.

[6 marks]

(b) The diagram below represents a hierarchy of objects, each represented by an integer. Leaf objects have an integer attribute. The integer corresponding to a composite object is the maximum value of its components.



Using the composite design pattern, write the code for the three objects (minor syntactical errors will not be penalised):

i) component

[2 marks]

ii) leaf

[4 marks]

iii) composite

[5 marks]

(c) Write a main method that uses your classes to create the data shown in the figure. The last line in your main should display the value of the top node.

[3 marks]

4. Please write the answers to the following 10 multiple-choice questions clearly in your answer booklet. Each correct answer is worth 2 marks. Incorrect answers will result in a penalty of two thirds of a mark to discourage guessing.

(a) Which of the following is true in Java:

A. A class can extend many classes and implement many interfaces.
B. A class can extend only one other class and implement only one interface.
C. A class can extend only one other class but implement many interfaces.
D. A class can extend many classes and implement only one interface.

[2 marks]

(b) Which of the following statements about assertions is **true**:

A. Assertions can be enabled at run-time.
B. Assertions can be enabled when compiling.
C. Assertions are the same as exceptions.

D. You should never use assertions.

[2 marks]

(c) Which of the following correctly describes the output of this code snippet:

```
public static void main(String[] args) {
    int a = 6;
    int b = a;
    b = b * 2;
    System.out.println("a: " + a + ", b: " + b);
}
```

A. a: 6, b: 6
B. a: 12, b: 12
C. a: 6, b: 12
D. a: 12, b: 6

[2 marks]

(d) Which of the following correctly describes the output of this snippet:

```
public class Elephant {
        public static int age = 3;
        public static void main(String[] args) {
           Elephant e = new Elephant();
           Elephant b = new Elephant();
           b.age = 2;
           System.out.println(""+ e.age + " " + b.age);
        }
}
```

A. 2 2
B. 3 3
C. 3 2
D. Nothing: the attribute cannot be accessed without a getter method.

[2 marks]

(e) Which of the following statements is **false:**

A. A final method cannot be overloaded.
B. A static method can be public
C. Private attributes can be accessed by sub-classes
D. Objects stored in `final` references cannot be modified.

[2 marks]

(f) A class extending an abstract class must:

A. Implement all abstract methods declared within the abstract class.
B. Implement all abstract methods declared within the abstract class or be declared abstract itself.
C. Implement none of the abstract methods within the abstract class.
D. Not change any attributes declared within the abstract class.

[2 marks]

(g) What is the correct (concurrent) way to use the following object:

```
public class EggTimer implements Runnable {
    private int duration = 5000;
    public void run() {
        try {
         Thread.sleep(5000);
        }catch(InterruptedException e) {
         // do nothing
        }
    }
}
```

A. `new EggTimer().start();`
B. `new EggTimer().run();`
C. `new Thread(new EggTimer()).start();`
D. `new Thread(new EggTimer()).run();`

[2 marks]

(h) An object is destroyed by Java's garbage collection when:

A. it is unreachable.
B. the method in which it was created ends.
C. the program terminates running.
D. It is no longer required.

[2 marks]

(i) `aThread.join()` has the following effect:

A. Pause aThread until the calling thread has finished.
B. Pause the calling thread until aThread has finished.
C. Give aThread access to the calling scope.
D. Stop aThread.

[2 marks]

(j) When programming in Swing, which of the following statements about concurrency is **false**:

A. Native `Thread` and `Runnable` objects should not be used.

B. Concurrency can be achieved by creating objects that extend `SwingWorker`.

C. The event dispatch thread is not the place for heavy computation.

D. Objects extending `SwingWorker` should implement a `run()` method.

[2 marks]