



University  
of Glasgow

Thursday 5 May 2016  
2.00 pm – 4.00 pm  
(Duration: 2 hours)

**DEGREES of MSc in Information Technology,  
MSc in Software Development**

## **Advanced Programming (IT)**

**Answer all 4 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) Describe a possible application for the **Observer** design pattern. [4 marks]
- (b) Describe the objects required for the **Observer** pattern. Ensure it is clear what type of object each is (class, abstract class, interface, etc). [6 marks]

Consider the following classes that implement the **Composite** design pattern to store University courses and sets of courses (programmes):

```
// Component
public interface Component {
    public Integer getCredits();
    public String description();
}
// Course Leaf
public class Course implements Component {
    private String name;
    private Integer credits;
    public Course(String n,Integer c) {
        name = n; credits = c;
    }
    public Integer getCredits() {
        return credits;
    }
    public String description() {
        return name + "(" + credits + ")";
    }
}
// Composite Leaf
public class Programme implements Component {
    private String name;
    private ArrayList<Component> courses = new ArrayList<Component>();
    public Programme(String n) {
        name = n;
    }
    public void addComponent(Component c) {
        courses.add(c);
    }
    public Integer getCredits() {
        Integer total = 0;
        for(Component c: courses) {
            total += c.getCredits();
        }
        return total;
    }
    public String description() {
        String des = "Programme contents: ";
        for(Component c: courses) {
            des += c.description() + ", ";
        }
        return des;
    }
}
```

- (c) Write a main method that uses this set of classes to create an `mscit` programme object that includes three courses: Machine Learning (20 credits), Advanced Programming (IT) (20 credits) and Research Readings in Computing Science, which is itself made up of Information Retrieval (5 credits) and Human-Computer Interaction (5 credits). At the end of your main method your code should display the entire programme and calculate the total number of credits. Minor syntactical errors will not be penalised.

[5 marks]

- (d) Describe how polymorphism is used within the `Composite` pattern.

[5 marks]

2. (a) In multi-threaded programs, state what is meant by a *race condition*.

[2 marks]

- (b) Sketch a multi-threaded system in which a *race condition* could occur and describe the circumstances in which the *race condition* would occur. Detailed code is not required, but all objects and the interactions between them should be defined. Minor syntactical errors will not be penalised.

[8 marks]

- (c) The `synchronized` keyword in Java can be used to overcome race conditions. Show how it could be used in the system you have described in part (b).

[4 marks]

- (d) Describe (no code required) an alternative approach to overcoming the race condition in a multi-threaded application.

[4 marks]

- (e) State two applications in which multi-threading is particularly important.

[2 marks]

3. Assume that the University has stored details of all students in a Java class called `studentDatabase()`. The class has two methods: `getName(String matric)` and `getProgramme(String matric)` which return the students name and degree programme when passed their matric number. Your task is to design a client / server system that allows clients to query the `studentDatabase()` object (stored on the server). Clients should specify if they want the student's name, or programme.
- (a) Describe a suitable protocol for this system. The protocol should be sufficiently detailed that someone else could implement the system.  
[4 marks]
  - (b) Write the code for the server as a class with a `main` method. Assume that the server is only ever connected to one client at a time.  
[6 marks]
  - (c) Write the code for a client as a class with a `main` method. Minor syntactical errors will not be penalised. The client should ask for the name of the student with matric number 999999.  
[6 marks]
  - (d) Briefly describe how you would extend your code to allow for multiple clients to connect at once.  
[4 marks]
4. Please select **one** answer (A, B, C or D) for each of these ten multiple-choice questions. Each correct answer is worth 2 marks. Incorrect answers will result in a penalty of two thirds of a mark to discourage guessing. A negative mark overall for this question will be rounded up to zero.
- (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) 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]

- (c) Consider the following line of code that might appear in a class definition, where the object `MyObject` is mutable:

```
public final MyObject a = new MyObject();
```

The implication of the `final` decorator is:

- A. a cannot be changed to reference a different object.
- B. The attributes of the `MyObject` object referenced by a cannot be changed.
- C. a is a constant.
- D. a is static.

[2 marks]

Parts (d) and (e) are based on the following code snippet:

```
public class ExamSnippet {  
    public static void IntChange(Integer x) {  
        x += 3;  
    }  
    public static void main(String[] args) {  
        Integer a = 2;  
        IntChange(a);  
        System.out.println(a); // Line A  
    }  
}
```

- (d) Which of the following statements correctly describes the output from Line A?
- A. The output is 2 because `Integer` objects are immutable. When it looks like the object is changed in `IntChange`, a new object is actually created.
  - B. The output is 2 because `Integer` is a Java primitive and therefore passed by value. Changes in the scope of `IntChange` are not reflected in `main`.
  - C. The output is 5 because the object is changed in `IntChange`.
  - D. The output is 2 because the variable is given different names (x and a) in the two scopes.

[2 marks]

- (e) The `IntChange` method is declared `static`. What does this mean?

- A. The method cannot be changed.
- B. The method cannot be overloaded within `ExamSnippet`.
- C. The method can be called without creating an instance of `ExamSnippet`.
- D. The method always gives the same output.

[2 marks]

Parts (f) and (g) are based on the following code:

```
public class ExamThread {
    public static class PrintThread extends Thread {
        private int mynumber;
        public PrintThread(int n) {
            mynumber = n;
        }
        public void run() {
            System.out.println("I am thread " + mynumber);
        }
    }
    public static void main(String[] args) {
        PrintThread[] threadSet1 = new PrintThread[100];
        PrintThread[] threadSet2 = new PrintThread[100];
        for(int i=0;i<100;i++) {
            threadSet1[i] = new PrintThread(1);
            threadSet2[i] = new PrintThread(2);
        }
        for(int i=0;i<100;i++) {
            threadSet1[i].start();
            threadSet2[i].start();
        }
        System.out.println("Finished!");
    }
}
```

(f) Which of the following statements most accurately describes the output of the program:

- A. The program will output "I am thread 1" 100 times followed by "I am thread 2" 100 times.
- B. The program will alternate between "I am thread 1" and "I am thread 2".
- C. The program will not output anything.
- D. It is impossible to predict the order in which the 100 "I am thread 1" messages and the 100 "I am thread 2" messages will be output.

[2 marks]

(g) At the end of main, the program prints "Finished!". Which of the following is true:

- A. The program will output "Finished!" once the main has finished starting all of the threads.
- B. The program will output "Finished!" after all of the "I am thread 1" and "I am thread 2" messages.
- C. The program will output "Finished!" after all of the PrintThread objects have completed.
- D. The program won't work because main has to join threads in order for them to run.

[2 marks]

- (h) The *composite* design pattern is:
- A. A design pattern suitable for adding functionality to existing classes
  - B. A design pattern suitable for systems where the same operations might be performed on objects or groups of objects
  - C. A design pattern suitable for wrapping objects within others to add functionality
  - D. A design pattern suitable for systems where the same operations might be performed on different types of objects
- [2 marks]
- (i) Consider a class `Parent` that is subclassed by a class `Child`. The expression: `Parent c = new Child();` is an example of:
- A. Overloading
  - B. Abstraction
  - C. Polymorphism
  - D. Overriding
- [2 marks]
- (j) Which of the following statements about Java program execution is true:
- A. Each thread has its own heap and all threads share a single stack
  - B. There is only ever one stack and heap
  - C. Each thread has its own stack and all threads share a single heap
  - D. Each thread has its own heap and its own stack
- [2 marks]