

UNIVERSITY OF EDINBURGH  
COLLEGE OF SCIENCE AND ENGINEERING  
SCHOOL OF INFORMATICS

**INFR08019 INFORMATICS 2C - INTRODUCTION TO  
SOFTWARE ENGINEERING**

**Saturday 19<sup>th</sup> December 2015**

**09:30 to 10:30**

**INSTRUCTIONS TO CANDIDATES**

**Answer any TWO questions.**

**All questions carry equal weight.**

**CALCULATORS MAY NOT BE USED IN THIS EXAMINATION**

Convener: D. K. Arvind  
External Examiner: C. Johnson

**THIS EXAMINATION WILL BE MARKED ANONYMOUSLY**

1. (a) One requirement elicitation technique involves exploring scenarios for the use of the system to be designed, perhaps captured in the form of user stories or use cases. Name and briefly describe two other elicitation techniques. [4 marks]
- (b) Consider designing software for a smart fridge that keeps track of the items it contains and the use-by date of each. A display mounted on the outside of the fridge door continually displays a list of all items in the fridge, highlighting those items that are near or past their use-by dates. A camera system observes items added to or removed from the fridge. As items are added or removed, the camera system can easily both recognise the items and read use-by dates that we assume are clearly marked on all items.  
Construct a use-case diagram for the system. [7 marks]
- (c) Identify key classes required for a design of this system, and draw a UML class diagram for the system, including multiplicities on the associations between classes. For each class, include any attributes and operations that are needed for supporting the use-cases identified previously. There is no need to show the types of the attributes and operations. However, do use names for the attributes and operations that describe what they are for or what they do. If any need a few words of extra explanation, add a UML comment box with these words. [8 marks]
- (d) Name the first three phases of the Unified Process, and give an example of a deliverable that each produces. [6 marks]

2. (a) i. Test first and test-driven software development processes regard as essential the automation of test running and result checking. Explain why and discuss briefly whether automation is as useful if testing is done after code development. [4 marks]
- ii. JUnit is a popular framework for automating the testing of Java programs. Consider a Java class for a stop-watch
- ```
public class Stopwatch {
    public int seconds;
    public int minutes;
    public Stopwatch(int theMinutes, int theSeconds) {...}
    public void increment() { ... }
    ...
}
```
- where the `increment()` method advances the time by one second, incrementing the `minutes` field and resetting the `seconds` field to zero as needed. Write a JUnit 4 test class with a method for checking that `increment()` functions correctly when invoked on a stop-watch object for 2 minutes, 59 seconds. [4 marks]
- iii. Write JML Java comments expressing a post-condition for the `increment()` method and a class invariant. The post-condition should specify the relationship between the new and old values of the `seconds` field. The class invariant should give suitable bounds on the values of the `seconds` and `minutes` fields. [4 marks]
- iv. JML assertions can be checked either statically without any program runs or dynamically at run-time. What is the main advantage of checking them statically? What is the main problem with static checking? [3 marks]
- (b) One common security requirement for software is that it should maintain confidentiality of sensitive information. Name two other common security requirements and, for each, give an example of a situation in which the requirement would be violated. [6 marks]
- (c) i. The Gnu General Public Licences are called *copyleft* licences. What does *copyleft* mean? [2 marks]
- ii. You have developed an app that links with a library released under one of the Gnu GPL licences. You don't want to release your source code, but does the licence require you to? Discuss. [2 marks]

3. (a) Draw a UML sequence diagram for the invocation of the following `getFile` method on some `FileUtils` object. Be sure to show calls of every method and constructor and returns for every method.

```
public class FileUtils {
    public File getDefaultPath {...}

    public File getFile(String fileName) {
        File file = new File(fileName);
        bool pathMissing = ! file.isAbsolute();
        if (pathMissing) {
            File defaultPath = getDefaultPath();
            file = new File(defaultPath, fileName);
        }
        return file;
    }
    ...
}
```

[9 marks]

- (b) i. Why is refactoring an important practice in agile processes? [2 marks]
- ii. Give two reasons for why the *Extract Method* refactoring might be useful. [2 marks]
- iii. Show the two new methods that result from applying the *Extract Method* refactoring to the body of the *for* loop in the following code.

```
public class SignalProc{

    public int[] clipWaveform(Signal[] wf, Signal lo, Signal hi) {
        for (int i = 0; i != wf.length; i++) {
            if (wf[i].compareTo(lo) == -1) {
                wf[i].set(lo);
            } else if (wf[i].compareTo(hi) == 1) {
                wf[i].set(hi);
            }
        }
    }
    ...
}
```

[5 marks]

QUESTION CONTINUES ON NEXT PAGE

*QUESTION CONTINUED FROM PREVIOUS PAGE*

(c) One common model that version control systems adopt for managing multiple programmers working on the same code base goes by the name of *copy-modify-merge*.

i. Name a version control system that uses this model. [1 mark]

ii. Two programmers Alice and Ben in their checked out versions of a code base make modifications to overlapping portions of one of the files. Describe a sequence of steps they must go through in order to end up with a merged version of the file in the repository, mentioning any *update* and *check-in* VC system operations they execute. Assume for simplicity they are not using a distributed VC system. [6 marks]