One and a half hours

Appendices A and B are located at the back of the exam

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Software Evolution

Date: Monday 1st June 2015
Time: 09:45 - 11:15

Please answer any TWO Questions from the FOUR Questions provided

This is a CLOSED book examination

The use of electronic calculators is NOT permitted
Question 1.

a) You have recently become the IT manager for a firm that supplies cakes to a chain of coffee shops. The firm’s IT system was bought in from a third party development firm that is no longer in business. The original contract required that the design documents and source code were supplied to your new employers. It is known that previous IT managers have overseen both corrective and perfective changes in the system. Your first development task is to make a perfective change to the system so that it allows an analysis that shows how the demand for different types of cakes varies with the season.

i) Excluding the source code, identify what you believe will be the three most useful sources of information that you could use to understand the system. For each of your sources, describe why it will be useful to you. [3 marks]

ii) Outline the approach that you will use to apply your three sources of information so that you understand the system code sufficiently to make the required change. [3 marks]

b) Outline what representations like Data Dependency Graphs (DDG), Control Flow Graphs (CFG), Control Dependency Graphs (CDG) and Post Dominates Tree (PDT) can be used to show, and the use to which they are typically put in software evolution. [3 marks]
c) The code fragment shown below comes from software that staff in the School of Computer Science use to transfer marks between different systems. Draw a control flow graph for the body of the method (i.e., lines 5-30).

```java
1 static final int marksColumn = 3;
2 Map<Integer, Float> marks = new HashMap<Integer, Float>();
3 boolean marksUpdated;
4
5 private void updateSpreadsheet(final File file) throws ...
6 {
7     DocumentBuilderFactory dbf
8         = DocumentBuilderFactory.newInstance();
9     DocumentBuilder db = dbf.newDocumentBuilder();
10     spreadsheet = db.parse(file);
11     final NodeList rows
12         = spreadsheet.getElementsByTagName("Row");
13     for (int i = 0; i < rows.getLength(); i++) {
14         final Node row = rows.item(i);
15         final String idString = getStudentId(row);
16         if (idString != null) {
17             Integer id = Integer.parseInt(idString);
18             if (marks.containsKey(id)) {
19                 final Node markCell = getColumn(row, marksColumn);
20                 if (markCell != null) {
21                     final Node mark =
22                         markCell.getFirstChild().getFirstChild();
23                     mark.setTextValue(marks.get(id).toString());
24                     marksUpdated = true;
25                 }
26             }
27         }
28     }
29 }
30 }
```

d) Which of the idioms mentioned in the course unit is present in this fragment? Give the language-independent pseudocode description of each idiom you discover, and state clearly (e.g. using the line numbers provided) which parts of the code correspond to which components of the idiom.
Question 2.

a) Lehman’s Second Law of Software Evolution can be paraphrased as the longer software exists, the worst that its quality becomes. Outline why this is true. [3 marks]

b) Code refactoring is one-way to improve the quality of software. However, historical studies have shown that it was seldom performed. In business terms, describe why this lack of refactoring occurred. [2 marks]

c) Current software development practices have resulted in an increase in refactoring being performed. Outline the features of these practices that have produced this increase in refactoring. [3 marks]

```java
public class Arithmetic {
    public int perform(int left, String op, int right) {
        int result = 0;
        switch (op) {
            case "+":
                result = left + right;
                break;
            case "-":
                result = left - right;
                break;
            case "*":
                result = left * right;
                break;
            case "%":
                // remainder
                result = left % right;
                break;
            case "<<":
                result = left << right;
                break;
            case ">>":
                result = left >> right;
                break;
            case ">>>":
                result = left >>> right;
                break;
            case "&":
                result = left & right;
                break;
            case "^":
                result = left ^ right;
                break;
            case "|":
                result = left | right;
                break;
            default:
                // Default case
                break;
        }
        return result;
    }
}
```

Question 2 continues on next page
Question 2 (continued)

d) Identify three bad code smells in the following piece of code. You should clearly indicate, e.g. by using line numbers, where the code smells you have identified are located. [6 marks]

e) One of the skills that a maintenance engineer must have is coping with unfamiliar languages. The code below uses in the Ecore modelling language.

i) Identify three characteristics of this language or questions you would need to answer to understand it more. For each of these, set them in terms of general programming knowledge that you have. [3 marks]

ii) Identify three characteristics of the domain being modelled, or questions that the model raises about the domain. For each of these, justify why you think this is a valid characteristic or question. [3 marks]

```java
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#';
2 import gmfgen : '…/';
3
4 package trace : trace = 'http://www.eclipse.org/gmf/2006/Trace' {
5     class TraceModel {
6         property nodeTraces : GenNodeTrace[*] { ordered composes };
7         property childNodeTraces : GenChildNodeTrace[*]
8             { ordered composes };
9         property linkTraces : GenLinkTrace[*] { ordered composes };
10        property toolGroupTraces : ToolGroupTrace[*]
11            { ordered composes };
12     }
13     abstract class AbstractTrace {
14        attribute visualID : ecore::EInt[?] { id };
15        attribute processed : Boolean[?] = 'false' { transient };
16     }
17     abstract class MatchingTrace extends AbstractTrace {
18        attribute queryText : String[?] { derived };
19     }
20     class GenNodeTrace extends MatchingTrace {
21        property nodeLabelTraces : GenNodeLabelTrace[*]
22            { ordered composes };
23        property compartmentTraces : GenCompartmentTrace[*]
24            { ordered composes };
25     }
26     class GenChildNodeTrace extends GenNodeTrace;
27     class GenNodeLabelTrace extends MatchingTrace {
28     }
29     class GenLinkTrace extends MatchingTrace {
30        property linkLabelTraces : GenLinkLabelTrace[*]
31            { ordered composes };
32     }
33     class GenCompartmentTrace extends MatchingTrace {
34     }
35     class GenLinkLabelTrace extends MatchingTrace {
36     }
37     class ToolGroupTrace extends MatchingTrace {
38     }
39 }
```
Question 3.

The hospitals in a region are merging their patient care facilities with those provided by community health centres and doctor’s surgeries to provide integrated care for patients with changing needs. One key aspect of creating this integrated care environment will be combining the IT services of these organisations.

The architecture of the hospital’s, reasonably modern, IT system is shown in Appendix A. The architecture of health centres’ and doctors surgeries’ IT systems is more ad hoc and is generalised in Appendix B.

You are the IT manager for the hospitals group. Your deputy is being asked to oversee the current systems, and you have been assigned the role of merging your IT services with those of the health centres and doctor’s surgeries. You have been given a report from IT consultants on how to support integrated patient care that suggests three options: 1) mutual limited query access between the two original IT systems, 2) migrating health centre and doctor’s surgeries data into the hospital system and then using this system, and 3) developing a new combined IT system.

a) Your first task is to assess each of the consultant’s options, (for example in terms of cost, complexity, and speed of deployment), and to recommend to the board with justifications which of them should be followed. [6 marks]

b) If option 2, migration to hospital system, was selected for supporting the integrated care environment, describe the process via which you would achieve this with no data being lost and all existing data being available to staff. [5 marks]

c) If option 3, new system, was chosen, your second task will be to define the architecture of the combined IT system. Outline an architecture that could result from this task; assumptions that you make should be stated and the board will want justification for your recommended architecture. [4 marks]

d) Assuming that your architecture from part c) is accepted and that a forward migration strategy will be used to migrate the current systems to this new architecture. Outline how you plan to do this. [5 marks]
Question 4.

a) You are head of the IS section of an engineering company that manufactures precision-made components for the aerospace industry. All of your components must undergo various stress tests to ensure that they will not break when in use. Currently you are testing physical devices to their limits, which because it involves destruction, is expensive. Someone has suggested that some of these tests could be performed on virtual representations of the components which would be both cheaper and quicker.

You have heard that another company has an existing implementation of the analysis program that you could license; experience within the manufacturing industry suggests that this implementation is reliable. However, the results that it outputs are unstructured, which means that post-processing software would need to be created to make the results useful to you. It has also occurred to you that if you had your own implementation of the program, the analysis results could be passed to the subsequent processing code via internal data structures.

Evaluate whether buying or creating the analysis algorithm is better in terms of the relative effort from your team, the quality of the solution that they would provide and their long-term maintenance costs. With reasons, identify which of these options you would recommend. [10 marks]

b) You have heard that an open source version of the core analysis algorithm mentioned in part a) is being developed. Also, rather than being a solid block of code, it will have an application programming interface (API) that allows integrators control over how they combine the algorithm into a complete system.

Evaluate the use of this open-source solution using the same criteria (effort from your team, the quality of the solution and long-term maintenance costs) as you used in part a). Recommend, with reasons, whether or not your recommendation of part a) should be changed to the adoption of this open-source software solution. [6 marks]

c) Software evolution is not just about evolving programs, it is also about evolving the techniques and languages that are used to develop programs. For both Dependency Injection (DI) and Aspect Oriented Programming (AOP) describe what advantages the adoption of these techniques can bring to the development of software. [4 marks]
Appendix A – Hospital’s IT System Architecture
Appendix B – Health Centre’s and Doctor’s IT System Architecture