Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Modern Software Engineering Practice

Monday 14th January 2008
Time: 14:00 – 16:00

Please answer any TWO Questions from the FOUR questions provided
This is a CLOSED book examination

The use of electronic calculators is NOT permitted
1. a) Explain, briefly, **four** reasons why Formal Methods have **not** been widely adopted by mainstream Information Technology (IT) software developers.

   (8 marks)

b) Draw a **state transition diagram** that abstracts over the facilities normally provided by conventional file-based program development systems which embody the “edit-compile-execute” cycle. Your diagram should model the **states** and **transitions** between those states for programs which parse correctly, are made executable and run, and also those for which the parse process fails. Ensure that the diagram models the application of the editor to a program’s source description such that a program is returned to an appropriate state regardless of its current state.

   (8 marks)

c) Explain, briefly, two advantages of the emphasis on **iterative development** in the Rapid Unified Process (RUP).

   (4 marks)

2. a) What main purpose does a Work Package (WP) in PRINCE2 serve, how does a manager exploit such WP’s, what form(s) can such WP’s take, and what does a WP contain?

   (8 marks)

b) Consider the Use Case diagram for the normal operation of a CD player shown below. In this diagram, the CD player retracts the tray holding the CD, reads the CD, and starts playing.

   Construct a modified Use Case diagram in which the presence of the CD is verified, i.e. the behavior of playing the CD extends the behavior of verifying that the CD is present such that it is a **special case** of the original, and, if the CD is not present, the user is prompted for a CD.

   (8 marks)

c) Explain, briefly, two **weaknesses** of PRINCE2 as a technique for structured project management.

   (4 marks)
3. a) Explain, briefly, what the terms “concrete syntax” and “abstract syntax” mean in the context of the board used to play a game of noughts and crosses and the data structure used to represent the state of the game in a program that allows users to play the game of noughts and crosses. Why is it important to distinguish between the concrete and abstract syntax representations and to make them a separation of concerns? Why is the board that is typically used to play the game unable to record the order in which the moves were made? (8 marks)

b) Develop a BNF grammar to describe the sequences of keystrokes made by an infallible (i.e. never makes mistakes) user of the simplified mobile telephone whose interface is shown below. To make a call users must obtain the interaction dialogue on the screen by pressing the Start button, they may then enter the digits representing the number to be called where numbers with only 3 digits are “speed dial” numbers (all other numbers are of longer length), followed by the Dial button, or, alternatively, they may press the Contacts button, scroll up and down a list of telephone numbers with contact names using the Up and Down buttons, and select the number to dial by pressing the Dial button. Calls are ended by pressing the “End Call” button which clears the display. Ensure that the resulting grammar is systematically refined such that it is possible to identify each step in the refinement process. (8 marks)

c) Explain, briefly, two disadvantages of XML as a grammar notation-technique. (4 marks)
4. a) State, briefly, two limitations of a module construct in a modular imperative language, and show, with the aid of an example stack ADT defined in such a module, together with a program that uses the stack module to declare and manipulate a variable of type stack, how these limitations manifest themselves.

(8 marks)

b) Given the definition of the type time in terms of a conventional cartesian product, and the two variables arrival_time and departure_time of type time shown below:

```
TYPE time = RECORD
    hour: 1..23;
    minute: 0..59;
END;
```

VAR arrival_time, departure_time: time;

Construct an equivalent Java implementation of a class Time, with appropriate attributes and a constructor, and also definitions of two suitably initialised objects arrivalTime and departureTime of class Time.

(8 marks)

c) Explain, briefly, two disadvantages of object orientation.

(4 marks)