Ollscoil na hÉireann
The National University of Ireland

Coláiste na hOllscoile, Corcaigh
University College, Cork

Summer Examinations 2011

Fourth Science: Computer Science

CS4000 Software Engineering

Dr Carron Shankland
Professor J. Bowen
Dr. J. Herbert

Answer four questions (all questions carry equal marks)

Three hours

1.

(a) Many modern software development projects follow an agile process, such as extreme programming. Discuss, with software application examples, why one might **not** use an agile process for a software project. (6 marks)

(b) For some large projects (such as the software for Boeing's 777 aircraft) the cost of testing of various kinds can amount to half the total software development cost. Do you think this is a reasonable proportion or do you think it can be improved on, and, if so, suggest how this cost might be reduced while not affecting product quality. (6 marks)

(c) What are the main benefits of software configuration management in a software development project? (6 marks)

A smartphone app (application) is under development to support a real-time location-based taxi-booking service. This will allow mobile users to easily check availability and book a taxi. You are the software project manager for this project. (In the following questions please ensure your answers are relevant to **this particular project**.)

(d) What general software process would you use to develop the software for this particular system? Describe the process and give the three most important reasons for its use on **this** project. (10 marks)

(e) Describe any limitations of the chosen software process. (4 marks)

(f) What, giving reasons, are the two most important non-functional tests for the software system? (4 marks)

(g) What is meant by software validation, and state, with reasons, the single most important method of validation that should be done on this project? (4 marks)

2.

(a) Use cases have a very central role in certain software development processes. Present arguments for and against such a central role for use cases. (6 marks)

A software system is under development to support online ticketing for sports events. The system will support various functions including event browsing, ticket booking and purchase. A software process using Larman's lightweight UML process will be used in the development of this system.

(b) For the above system: give the use case text for an important use case; construct a simple conceptual model; draw an appropriate **system** sequence diagram; for one identified operation (method) write down the Larman-style contract; draw a possible collaboration diagram implementing this operation. (15 marks)

(c) Which (if any) of the above five activities corresponds to object-oriented analysis and which (if any) corresponds to object-oriented design? (3 marks)

(d) What role (function) does the **system** sequence diagram play in the process?(5 marks)

(e) Discuss how the collaboration diagram relates to previous stages of Larman's UML process. (5 marks)

(f) Describe how Larman's Controller software design pattern might be used in this system, and describe the benefits of this pattern for this project. (6 marks)

3.

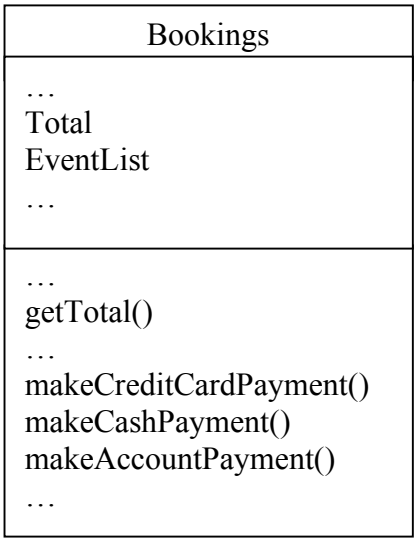
(a) Discuss why component-based software development has not become as successful as component-based hardware development. (7 marks)

(b) Coupling and cohesion may be described by Larman as *evaluation patterns* in software design. Explain what this means and discuss why coupling and cohesion are important in software design. (6 marks)

(c) Explain the subtle differences between fault (or bug), error and failure. (3 marks)

(d) A program takes two integers and computes an integer result. You do not have the source of the program but you are told that the program takes numbers representing a year and a month and returns the number of days in that month. What method would you use to devise tests for the program, and give examples of the tests? (6 marks)

The following diagram illustrates a design of a class to represent a Bookings object in an event booking system. Attributes of the class include the list of items being purchased and the running total cost of items being purchased. Methods of the class include various get methods and methods for the various ways of paying for the events.



(e) Evaluate the above design, stating any limitations it has. (4 marks)

(f) Improve the above design so that it can be more easily extended for other kinds of payment. Name the software design pattern used, state its benefits, and illustrate your answer with an altered class diagram. (8 marks)

(g) Describe the Factory software design pattern, and discuss how it could be used to improve the above design. (6 marks)

4.

(a) Discuss the challenge of scalability, with appropriate examples, as faced by large modern distributed applications, such as those of companies such as Facebook, ebay and Amazon. (8 marks)

(b) Certain properties of a distributed system are captured by Brewer's CAP conjecture (later proved as a theorem). State Brewer's CAP theorem, and explain how the CAP theorem influences companies such as Amazon in the design of their information systems. (6 marks)

An on-line film (movie) store is being developed. The system will allow customers to browse, search and buy films, which can then be downloaded to their home device. The site will act as a portal allowing customers buy films from various third party film distributors. The site will keep customer profiles, provide recommendations, and also support social interaction through special interest film discussion groups. (Please ensure that your answers for following questions are relevant to this particular project.)

(c) What is the most suitable distributed architecture model for **this** system? Give detailed reasons for your answer, stating three benefits and any limitations of the architecture. (8 marks)

(d) State the three most important non-functional requirements for **this** system. Describe in detail how you would implement the system to satisfy these. (8 marks)

(e) Describe any trade-offs that arise when trying to satisfy these non-functional requirements. (4 marks)

(f) Describe an alternative distributed architecture model that might be suitable for **this** system, and describe two advantages and two disadvantages with respect to the previous architecture. (6 marks)

5.

C ::= ***V* := *E***
| *C1* ; *C2*
| IF *B* THEN *C1* ELSE *C2*
| BEGIN *V1; … Vn; C* END
| WHILE *B* DO *C*
E ::= N | V | E1 + E2 | E1 - E2 | E1 x E2 | …
B ::= T | F | E1=E2 | E1 <= E2 | …

The above gives the syntax of a simple programming language (E and B being the syntax of the numeric and boolean expressions respectively). The semantics of the language may be defined using Floyd-Hoare logic.

(a) Give the meaning of the Hoare triple {P}C{Q} as used in program verification. (5 marks)

(b) State the Floyd-Hoare axiom or rule for one construct of the above language (such as the assignment axiom for ***V* := *E*** or sequencing rule for ***C1* ; *C2*).** (4 marks)

(c) Discuss how one might formally verify a full program using Floyd-Hoare logic. (6 marks)

(d) Meyer's Design by Contract paradigm is based on the Floyd-Hoare formal approach but has some differences. What are these differences? (4 marks)

(e) A class represents a simple lift controller. This class has boolean attributes representing the state of the lift: door_is_open; moving_up; moving_down; between_floors; and stopped. Write down a class invariant in standard mathematical logic for this class stating the most important invariant properties. The class includes two operations (methods): one to make the lift go up; one to open the lift doors. Specify these two operations using a contract expressed in standard mathematical logic. (6 marks)

(f) Construct a statechart to model accurately a software controller for a self-serve petrol pump. The object should respond to events such as: credit card swiped; nozzle out of pump housing; select fuel type; nozzle on; nozzle off; nozzle in housing. Include in the statechart at least four features not found in conventional state machines, and indicate clearly these four statechart features in the diagram. (12 marks)

(g) Discuss what issues arise if the software crashes and is restarted while you're filling with petrol. (3 marks)