THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

**UNIVERSITY OF LONDON** **CO2226 ZB**

**BSc Examination**

**COMPUTING AND INFORMATION SYSTEMS, CREATIVE COMPUTING and COMBINED DEGREE SCHEME**

Software engineering, algorithm design and analysis

Friday 11 May 2018: 14.30 – 17.30

Time allowed: 3 hours

This paper is in two parts: Part A and Part B. There are a total of **THREE** questions in each part. You should answer **TWO** questions from Part A and **TWO** questions from Part B.

Full marks will be awarded for complete answers to a total of **FOUR** questions, **TWO** from Part A and **TWO** from Part B. The mark for each part of a question is indicated at the end of the part in [.] brackets.

Only your first **TWO** answers from Part A and first **TWO** answers from Part B, in the order they appear in your answer book, will be marked.

There are 100 marks available on this paper.

A handheld calculator may be used when answering questions on the paper but it must not be pre-programmed or able to display graphics, text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

## Part A

### Question 1

a) You are responsible for the verification and validation of a new design that was proposed to your company and your manager has given you the task to verify use cases against sequence diagrams. Briefly explain how you would go about this and what you would be looking out for and comparing.

[5]

b) PartyFun.com is a new online company organising parties and celebration events. They cover a range of activities, from the catering to entertainment, space configuration, *etc.*

- Some of these activities (*e.g.* catering) will have a minimum price, a standard set-up and a price that will vary according to the number of people while others (*e.g.* audio configuration for an event) will have a fixed price for the event.
- The customer will have to register with the website providing their name, phone number, postal and email addresses and obtain a username and a password. They will also have to indicate what kind of activities they are interested in.
- Once the registration process is over, the customer will have the option to open up a new event configuration, enter the details of the event and activities they will require as well as, where needed, information about how many guests will be taking part.
- As it is a new start-up company, to attract new customers, PartyFun.com has introduced a loyalty scheme which places customers into discount bands on the basis of previous number of events/expenditure (*e.g.* if you have done five events with the company or spent over £10,000, whichever comes first, you get 5% off, and for ten events or £10.000 spend, 10% discount, *etc.*, up to a maximum of 20% discount). The manager will have the option to overwrite the rules at their discretion and offer a higher discount and, if this is the case, this will be recorded on the system. A special report will be produced by the administrator every month listing this information.
- Once the information has been provided by the customer, the system will allocate an event code and present the customer with a quote, taking into account any discount rules. The customer will have a choice of either "locking" the price for the next two days until they make up their mind, or pay a deposit of 10% upfront. The event is not confirmed unless they make the payment and after two days, if payment has not been made, it will be deleted from the system, and a follow-up email sent.
- The company does not process the payment, but passes the information to a payment gateway that does it for them. They do, however, keep information about the payment such as means of payment, end result, authorisation code if they need to trace back any details, *etc.*
- The company has both individual and corporate clients, with the latter allocated a dedicated event manager who oversees the details of any event they organise. In the case of individual customers, an events manager is

allocated from a pool of relevant employees. Thus, an individual client may be allocated different event managers for different events they organise.

- The customer has the right to cancel up to two days before the event. If the cancellation is made a week before the event, the deposit is refunded, after which they lose 20% of it for every day that passes, and if they cancel two days before they lose their deposit). Also, for some activities (*e.g.* catering) they may still have to pay the full amount or part of it; for others they might get a full refund if no preparation or work has been done (*e.g.* audio set-up).
- The company makes use of a third-party mass mailing system to boost their business. When they deem appropriate they send out emails advising customers that there is a discount available or that a new service has become available.
- At the end of every month the system should generate a number of reports for management; these reports would include the number of events that the company ran that month, as well as what activities were requested. The administrators can also run these reports any time on demand.

Develop a **class** diagram for the above scenario using the appropriate naming conventions (*e.g.* class names starting with a capital letter), and suggest class attributes with name and type, as well as methods with name and return type. Illustrate associations, aggregations, and generalisation relationships between the objects.

[20]

**Question 2**

a) How would you demonstrate concurrency in an activity diagram? Illustrate your answer with an example.

[5]

b) Prepare a use case diagram to accommodate the specification of the PartyFun.com system described in **Question 1**. Your answer should include all use cases, actors and any associations between different use cases, different actors or use cases and actors. Please state any assumptions you make about the system.

[20]

**Question 3**

a) Briefly explain the concepts of abstraction and interface in Object-Oriented Design; use an example for each to illustrate your answers.

[5]

b) Create a state machine diagram describing the behaviour of an order object described in **Question 1**. The following rules apply:

- The order object becomes **New** once the customer selects the New Event option on the website;
- It moves to **Updating** while the customer is adding the new events that they want included;
- Once the customer has finished adding the events they are interested in, the order becomes **Specified**;
- In case the customer decides to lock the price for the next two days, the order will become **Pending**, otherwise, if the payment deposit is made, it will become **In Process**;
- A week before the event is to be held, the order will become **Active** as PartyFun.com will need to start preparing for the various events;
- It will remain **In Preparation** until the final day of the event while updates on the progress of the various events are registered;
- After the event finishes it will go into **Consolidation**, waiting for the final payment to be processed;
- If the payment is processed, the event will move into **Archived** for a further week, while it is being added to the customer's history of events and feedback is sought from the client;
- If feedback is provided, the object will spend a further day in **Feedback Processing**;
- After that day, it will be moved into **Closed**;
- If the customer cancels, assuming it is done in time for a refund, it will go into **Refund Processing**, and once the refund is processed, into **Cancelled**. Otherwise, it goes straight into **Cancelled** mode.

Design the state machine diagram using state transitions and labels with three parts.

[20]

## Part B

### Question 4

This question focuses on sorting and heaps.

a) Consider the partially sorted list below.

a b d c e g f

Apply **both** insertion sort and quicksort to the list above to sort into **alphabetical order**. Take care to show the steps taken for each.

Report which of the two methods is faster in this case.

[8]

b) Give the **worst-case** performance for **both** of the two sorting methods in (a). In the case of mostly sorted lists, which method would you expect to give better **average-case** performance? Give your reason(s).

[6]

c) What is a binary heap? State the **TWO** main properties of a heap. Give **ONE** example of where a binary heap would be a useful data structure.

[6]

d) A binary heap can be implemented as an array. Given a heap stored on the array below, draw the heap structure as a diagram. Explain briefly how the array and tree are mapped to each other.

a c b e g f d h

[5]

**Question 5**

This question focuses on linked lists and graphs.

a) What is a linked list? Use an example to illustrate your answer.

[4]

b) Give **ONE** advantage and **ONE** disadvantage of using a linked list as opposed to using an array.

[4]

c) What is the definition of a graph? Provide an example to illustrate your answer.

[4]

d) You have the following adjacency matrix for a graph G with vertices (A, B, C, D):

$$\mathbf{G} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

i) Draw the graph and digraph corresponding to this matrix.
ii) Write the adjacency list for both the graph and digraph.

Give a brief explanation of the process you used.

[10]

e) For the **graph** above, starting from vertex D, write the vertex sequence in the order that each vertex is visited when applying the depth-first traversal algorithm. Show the steps at each stage.

[3]

**Question 6**

This question focuses on hashing.

a) Explain what is meant by Hashing and give **TWO** reasons why it can be useful.

[6]

b) Given a hash function *h(k) = k mod 8* and an empty hash table, show the hash table after inserting the data (2 8 23 27 42 30 121). Show the steps in your calculation.

[6]

c) Using your answer to (b) as an example, explain what Collision is and why this is an issue in hashing.

[4]

d) Describe briefly what is meant by each collision method below, and then apply each to your answer in (b) to resolve the collision.

i) Closed address hashing.
ii) Double hashing.

[6]

e) Using the data item 69 as a search query what would be retrieved from the resolved hash table produced in (d) by closed address hashing?

[3]

**END OF PAPER**