

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, Algorithms design and analysis

Date and Time: Wednesday 11 May 2016: 14.30 – 17.30

Duration: 3 hours

There are SIX questions on this paper. Candidates should answer **FOUR** questions (**TWO** from **PART A** and **TWO** from **PART B**). The mark for each part of a question are indicated at the end of the part in [.] brackets.

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

There are 100 marks available on this paper.

A hand held calculator may be used when answering questions on this 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) Explain briefly what is dynamic binding in object-oriented programming. You may use an example to illustrate your answer

[5]

- b) ToDoNotes.com is an Internet start-up company that aims at “digitalising” the traditional paper-based list on which users are writing down a list of tasks that they will need to carry out in the near future. The aim is to provide a web interface where the user can register by providing his personal details (name, age, address and e-mail) and select his username and password. The following rules apply:

- After logging on to the site, the user is able to create to-do notes containing the text of the task (s)he wants to perform and a deadline.
- The system will attach a status to each note (new once created, open until five days before the deadline, deadline approaching up until three hours before the deadline, critical for the last three hours and overdue once the deadline has passed).
- The notes can be enhanced by the addition of “attachments”, i.e. other documents with content helpful to the user when carrying out the tasks specified in the note; examples of such documents can be Word documents, images and so on.
- On top of the free user, the user can register for a “premium” account: the benefit of such an account is that a user can be linked to “shared” notes from other users. This way users can create a project, each can create a note with their own tasks and share them together (a project can be created even if notes are not shared if the user wishes to group a number of their own notes together).
- In this case one can only edit her notes, but the status and text of the other note will be visible as well.
- Finally, in order to make the development of notes faster and easier for end-users ToDoNotes.com is offering a collection of already created text to be used as part of a note; these ready-made sentences are divided into different categories (e.g. home, work, computing and so on) depending on the text.

Develop a class diagram for the above scenario using the notions starting with a capital letter as class names, 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) Briefly explain the meaning of the agile process. How does it deal with requirements changes? [5]
- b) Prepare an activity diagram to illustrate the process of adding a note for the scenario described in **Question 1**. The following rules apply:
- The process starts when a customer has logged in and selected the option to add a new note.
 - The system will prompt the user to select the type of note he wants to add (personal or shared) and, if the user chooses to include it in a project, it will check if a project by that name exists or not.
 - If the project does not exist, then it will be created and a blank note will be added simultaneously; if it already exists, then a blank note will be added to the specified project while at the same time the contents of the project will be read in.
 - The user will, then, proceed to fill in the specifics for the note.
 - He might choose to include attachments to the note (e.g. Word documents, image files and so on).
 - Once the process of adding documents to the note is finished, the user should click on the Add Note button.
 - The system will then proceed to save the note.

Show on the activity diagram the flow of control using specialised UML components, like activities, decisions, merges, forks and joins.

[20]

Question 3

a) What is the purpose of a state diagram? How does it compare to a sequence diagram?

[5]

b) Create a state machine diagram describing the behaviour of an instance of the Note class for the scenario described in **Question 1**. The following rules apply:

- The note will be in a new state once created and will then progress to incomplete as the user is adding the information.
- Once the user finishes providing the information and submits it, the note becomes added; if the note is not part of a project then it is a standalone note, otherwise it is an embedded one.
- If the note has a start time attached to it once this time is reached it becomes in progress and when the user marks it as complete it will be completed.
- The conditions specified in **Question 1** would also need to hold, i.e. the note should be open until five days before the deadline, deadline approaching up until three hours before the deadline, critical for the last three hours and overdue once the deadline has passed.
- After three months that the note has been marked as completed, the note should become archived.

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

[20]

Part B

Question 4

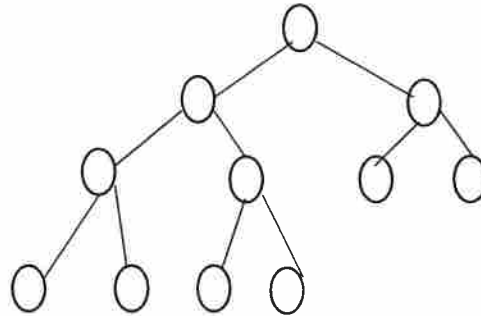
- a) Explain briefly what a queue is. You may use an example to illustrate your answer [5]
- b) Describe how a queue can be implemented if we use the data structure of an array for this purpose. Provide the implementation of the **enqueue** method for this type of implementation. [7]
- c) What is hashing and how does it work? You may use an example to illustrate your answer. [6]
- d) How does closed hashing resolve collisions? Illustrate your answer by using the hash function $h(k) = (k^2 - 1) \bmod 25$ and the rehash function $h(k) = (k + 1) \bmod 12$ for the following list of numbers: 4, 23, 78, 94, 521, 124, 144, 12, 75. [7]

Question 5

- a) What are the benefits of using recursive algorithms? You might use examples to support your arguments [6]
- b) Briefly explain how the merge sort algorithm works. Demonstrate the algorithm by sorting the following lists (please make sure that you show every step of the process): (12, 3, 42, 25) and (2, 45, 24, 7). [10]
- c) What are the main steps involved when using dynamic programming? [4]
- d) How is dynamic programming different from recursion? You may use an example to illustrate your answer. [5]

Question 6

- a) What are the three different ways in which a tree can be traversed? Provide the pseudocode for them. [10]
- b) What is the difference between a full binary tree and a complete binary tree? Is the following tree full and/or complete – you will need to fully justify your answer



[7]

- c) Briefly explain how binary search works and provide the pseudocode for its implementation. How would you use it to search for 15 in the following list of numbers: 1, 2, 4, 7, 9, 12, 15, 16, 21, 23, 24, 37, 41, 58?

[8]

END OF PAPER