ST ANDREW'S JUNIOR COLLEGE

Preliminary Examination

H2 COMPUTING   **9754/01**

PAPER 1   **25 AUG 2008**

TIME: 1300 – 1600 hrs   **3 hours**

READ THESE INSTRUCTIONS FIRST

If you have been given an Answer Booklet, follow the instructions on the front cover of the Booklet.

Write your class and name on all the work you hand in.

Write in dark blue or black pen on both sides of the paper.

You may use a soft pencil for any diagrams, graphs, music or rough working.

Do not use staples, paper clips, highlighters, glue or correction fluid.

Answer **all** questions.

At the end of the examination, fasten all your work securely together.

The number of marks is given in brackets [   ] at the end of each question or part question.

Total marks for this paper is **120** marks.

This question paper consists of 4 printed pages.

Answer **all** questions.

1 The Central Processing Unit (CPU) is connected to main memory by three separate **buses**.

(a) What is a **bus**? [2]

(b) Name the **three** buses connecting the CPU to main memory and, for each, briefly describe its purpose. [6]

(c) Name the **two** registers in the CPU which are used in communication between the CPU and main memory via the buses. Explain their roles in the fetch-execute cycle. [4]

2 A department store operates its own computerised credit system by issuing privileged customers with credit cards against which purchases can be charged, up to the customer's credit limit.

Each credit card has a five-digit account number, for example,

3475D

where D is a modulus-11 check digit for the account number.

(a) What is the purpose of including a check digit at the end of each credit card number?[1]

(b) Describe an algorithm for the calculation of the check digit. [5]

(c) Using your algorithm, calculate the value of D. [1]

(d) Describe the other operations which you would expect to be performed by the store's computerised system when a credit card purchase is made. [3]

3 (a) Explain the difference between a class and an object in the context of object oriented programming. [2]

(b) Draw an appropriate inheritance diagram to show the relationship between the classes: Abstract Data Type (ADT) List, ADT Stack, ADT Queue. [2]

(c) Using the inheritance diagram in part (b), explain, with clear examples, how the concept of inheritance was appropriately applied. [4]

(d) For the relationship illustrated in part (b), suggest one method/operation that may be polymorphic and hence explain what is meant by polymorphism. [3]

(e) A specification for an ADT is typically implementation independent.

(i) Write the specification for ADT Stack. [4]

(ii) Explain, using the specification in part (e)(i) as an example, what is meant by *implementation independent*. [3]

4 A garden centre stocks a wide range of garden plants, including shrubs and trees. The manager of the garden centre is planning to introduce a computer system, which customers can interrogate to obtain information about the plants which are in stock. For example, customers will be able to find out whether a shrub produces flowers or not, its maximum growing height and spread, the type of soil it prefers and any recommended growing conditions such as sun or shade.

(a) Bearing in mind that the system is to be made easy to use by the customers, describe suitable hardware for this system and a suitable software interface for customers, giving reasons for your answers. [6]

(b) The plant stock held by the garden centre changes according to the time of year. Suggest how the system should be designed so that the information held can be

(i) kept as up-to-date as possible, with minimum staff effort, and [2]
(ii) changed by the manager or one of his assistants, but not by a customer. [2]

(c) Identify the benefits of this system to

(i) the garden centre manager and staff [2]
(ii) the customers. [2]

(d) Is this system an expert system? If you think it is, describe the particular features it has which make it an expert system. If you think it is not, describe any features it requires before it can be an expert system. [4]

5 An ADT **Binary Tree** can be implemented using pointers or arrays.

(a) Given the following sequence of item entries into the binary tree: *Jane, Tom, Bill, Jack, Alice, Don*. Draw the diagram to illustrate the binary tree for (you should include illustration of the **free list** for *both* implementations)

(i) an array-based implementation,

(ii) a pointer-based implementation. [6]

(b) For the array-based implementation, outline the algorithm that will delete an item from the ADT Binary Tree. [4]

(c) For the pointer-based implementation, write the algorithm that will delete the whole binary tree, i.e. remove every node and return the memory to the free list. [4]

(d) A **Binary Search Tree** (BST) is a special type of binary tree that is ordered. Write the algorithm that will insert an item into a BST. [5]

(e) Explain a possible efficiency issue that the BST may face and suggest a way to resolve it. [3]

6 A library holds several thousand volumes of printed music. Each volume has a unique reference number and may contain between one and a hundred pieces of music. Many of the volumes contain music by several different composers, and throughout the library there will be hundreds of works by each of the best-known composers

Details of all this music will be stored in a database system, which will be used to produce alphabetical lists or to search for particular works by title or composer. There will be one record for each piece of music in each volume, comprising *its title, opus number*, and the *instruments* for which it is written. Also included will be the name and number of the volume in which it will be found, the composer, and the date and place of the composer's *birth* and *death*, a total of ten fields in all.

(a) If the data were all contained in a single file, the names of some of the volumes, and the place and date of birth and death of the most popular composers, would be typed many times. If these details were occasionally mis-spelt, the corresponding records would not be found during search operations, and, if any needed changing, the alteration would have to be made in each record separately.

   (i) Explain how a *relational* database can help to overcome these problems. [3]

   (ii) Suggest how this data could be organised in such a database, indicating how many files there would be and what fields each would contain. [6]

   (iii) Draw a relationship diagram to show the relationships between files. [6]

(b) Describe how this relational database system would make a list of all works by one composer in alphabetical order of title, each work being printed along with all its relevant information. [4]

7 (a) Explain the process of top-down analysis and explain why it helps in the solution of complex problems. [5]

(b) A manufacturer holds, on a computer system, details of over 5,000 products. Each component has a unique identifier.

   (i) Using an example, explain how the data can be organised as a random file. [4]

   (ii) Describe how the random file is updated when a particular product is no longer produced by the manufacturer. [5]

(c) The manufacturer accepts a product if it passes the following three tests.

   ▪ All dimensions are correct
   ▪ Strength tests are passed
   ▪ Paint tests are passed

   If the first test is passed but exactly one of the other two fails, the component is sent for repair. Otherwise the component is rejected.

   (i) Create a **decision tree** showing all the possible outcomes and results. [4]

   (ii) Simplify your solution by removing redundancies. [3]

~~ END OF PAPER ~~