344/01

COMPUTING CP4

SOFTWARE AND SYSTEM DEVELOPMENT

P.M. FRIDAY, 16 June 2006

(2 hours)

ADDITIONAL MATERIALS

In addition to this examination paper, you will need a 12 page answer book.

INSTRUCTIONS TO CANDIDATES

Answer **all** questions.

The intended marks for questions or parts of questions are given in brackets [ ]. You are advised to divide your time accordingly. The total number of marks available is 70.

You are reminded of the necessity for good written communication and orderly presentation in your answers.

1. *(a)* Describe a **computer** application where a *queue* is the most appropriate data structure to use, and explain why it is the most appropriate data structure. [2]

(b) Describe a **computer** application where a *stack* is the most appropriate data structure to use, and explain why it is the most appropriate data structure. [2]

2. It is very important that a computer system has a well designed Human-Computer Interface (HCI).

(i) Describe **two** advantages and **one** disadvantage of *handwriting recognition* as an **input** method. [3]

(ii) Describe **two** advantages and **one** disadvantage of *speech synthesis* as an **output** method. [3]

(iii) When might a *text-based interface* be preferable to a *graphical user interface* (GUI)? [1]

(iv) What is meant by the term *natural language* interface? [1]

3. *(a)* (i) Convert the 12 bit binary number 101111010111 to hexadecimal. [1]

(ii) Why is hexadecimal often used as an alternative to binary? [1]

(b) In a certain computer program, real numbers are *rounded* before being added together.

(i) Explain the term *rounding*, and give an example of its use. [2]

(ii) Why might the addition of rounded numbers cause excessive inaccuracy? [1]

(c) (i) Give an example of a number which **cannot** be stored as an *integer*. [1]

(ii) Describe what is meant by *floating point* form. [1]

(iii) Describe **one** advantage of storing a number as an *integer* when it is possible to do so, compared with storing the number in *floating point* form. [1]

4. *(a)* Draw a labelled diagram to show what is meant by a *linked list*. [2]

(b) In a certain implementation, a linked list of integers is actually stored in table form as shown below:

Address	Integer	Pointer
101	6786	
102	4231	
103	6971	
104	3217	
105	2410	
106	5233	
...		
...		

The integers are to be processed in ascending numerical order.

(i) The variable *first* points to the smallest integer in the list. What is the value of *first*? [1]

(ii) Copy the table and complete the pointer column. [3]

5. *(a)* A certain software tool includes a *program trace facility* and allows for *break points* to be set up.

(i) What name is given to this type of software tool? [1]

(ii) Outline the role of a *program trace facility*. [1]

(iii) Outline the role of a *break point*. [1]

(iv) Name one other facility which this type of software tool is likely to include. [1]

(b) Describe what is meant by a *subprogram library* and give an example of its use. [2]

(c) Give an example of an application which might use a *special purpose language*. [1]

(d) Describe **two** features of CASE tools which could be used during program development. [2]

(e) Explain what is meant by the term *relocatable code*. [1]

Turn over.

6. **J** is the eight digit binary number   10110110
   K is the eight digit binary number   01011010

There are three logical operations that can be carried out on a pair of binary numbers. These are **AND**, **OR** and **XOR**.

  (i)   A logical operation is carried out on **J** and **K** giving the result 11101100
      Which logical operation has been carried out in this case?   [1]

  (ii)   A different logical operation is carried out on **J** and **K** giving the result 11111110
      Which logical operation has been carried out in this case?   [1]

  (iii) Write down the effect of carrying out the logical operation **NOT J**.   [1]

  (iv) Write down the effect of carrying out an arithmetic shift left by one place on **K**.   [1]

7. *(a)* Explain what is meant by the term *algorithm*.   [2]

  (b) One method of describing an algorithm is a *flow chart*.
      State **two** other methods.   [2]

  (c) A *recursive algorithm* is one which calls itself.
      What other feature does a recursive algorithm need to have?   [1]

  (d) (i)   A program can pass variables to a procedure. What name is given to the variables passed in this way?   [1]

       (ii) During this operation, variables are sometimes passed by *value*. Variables can be passed by another method.
          Name the other method and explain how it operates.   [2]

8. One way of defining an *integer* is:

    a plus sign or minus sign or neither, followed by one or more digits

For instance, the following are covered by this definition:

    4   –3   +3   56   –247   0

  (i)   Give an appropriate Backus Naur Form (BNF) expression for an integer, as defined above, defining any terms you introduce.   [4]

  (ii)   Draw a syntax diagram for the definition of an integer given above.   [2]

9. An algorithm is required to work out the **minimum number of coins** required to pay any amount of cash up to 99 pence (99p), and output a list of the required coins.

Note: coins of the following values are available: 50p 20p 10p 5p 2p 1p.

For instance, for the amount 97p, the algorithm is required to output the following:

      50p coin 20p coin 20p coin 5p coin 2p coin

The first few lines of the required algorithm are shown below.

```
algorithm SelectCoin

input (AmountInPence)

if AmountInPence ≥ 50 then
begin
        output " 50p coin "
        set AmountInPence = AmountInPence - 50
end

while AmountInPence ≥ 20
            output " 20p coin "
            set AmountInPence = AmountInPence - 20
endwhile
```

(i)   Complete this algorithm. (It is not necessary to copy the above, but you may do so if you wish.) [4]

(ii)   This algorithm will **not** give the minimum number of coins when the amount input is greater than 99p. Describe **in detail** what will happen if an amount greater than 99p is input. [2]

10. In the following question, additional credit (up to 3 marks) will be gained if your answer demonstrates skill in written communication.

Computer programs normally need to be translated into a form acceptable to the computer before they can be executed.

Briefly describe the purpose of a *compiler*, an *interpreter* and an *assembler*, ensuring that you distinguish between the three.

Describe **in detail** the *principal stages* of compilation. [7+3]