OCR
RECOGNISING ACHIEVEMENT

**ADVANCED GCE UNIT** **2509**

# COMPUTING

Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms

**FRIDAY 15 JUNE 2007** Afternoon

Time: 1 hour 30 minutes

No additional materials are required.

Candidate Name

Centre Number

Candidate Number

**INSTRUCTIONS TO CANDIDATES**

- Write your name, Centre Number and Candidate number in the boxes above.
- Answer **all** the questions.
- Use blue or black ink. Pencil may be used for graphs and diagrams only.
- Read each question carefully and make sure you know what you have to do before starting your answer.
- If you run out of space for an answer, continue on the spare pages at the back of the booklet.
- Do **not** write in the bar code.
- Do **not** write outside the box bordering each page.
- WRITE YOUR ANSWER TO EACH QUESTION IN THE SPACE PROVIDED. ANSWERS WRITTEN ELSEWHERE WILL NOT BE MARKED.

**INFORMATION FOR CANDIDATES**

- The number of marks for each question is given in brackets [ ] at the end of each question or part question.
- The total number of marks for this paper is 90 (86 + 4 for the quality of written communication).
- You will be awarded marks for the quality of written communication where an answer requires a piece of extended writing.
- No marks will be awarded for using brand names of software packages or hardware.

| For Examiner's Use | | |
|---|---|---|
| Question no. | Max. mark | Mark |
| 1 | 9 | |
| 2 | 6 | |
| 3 | 9 | |
| 4 | 19 | |
| 5 | 14 | |
| 6 | 20 | |
| 7 | 9 | |
| WC | 4 | |
| Total | 90 | |

This document consists of **14** printed pages and **2** lined pages.



**[Turn over**

Answer **all** questions.

**1** **(a)** In a typical desktop PC operating system,

**(i)** state the purpose of the boot file.

.....

..... [1]

**(ii)** explain the term virtual memory.

.....

.....

.....

..... [2]

**(b)** In a network operating system,

**(i)** explain the term transparency.

.....

.....

.....

..... [2]

**(ii)** describe how printing is handled.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**2** **(a)** Describe what happens during lexical analysis when using a compiler.

.....

.....

.....

.....

.....

..... [3]

**(b)** Typing errors such as 'PINT' instead of the reserved word 'PRINT' must be handled by the compiler.

Explain how this is done.

.....

.....

.....

.....

.....

..... [3]

**3** **(a)** During the fetch-execute cycle,

**(i)** state what is held in the program counter.

.....

..... [1]

**(ii)** state **two** circumstances that will cause the value in the program counter to change. For each, state how the value changes.

1 .....

.....

.....

.....

2 .....

.....

.....

..... [4]

**(b)** An array processor architecture is one in which an array of processors, usually controlled by another processor, can perform identical operations on a number of different data items at the same time.

**(i)** State an advantage of an array processor architecture compared with von Neumann architecture, giving a reason for your answer.

.....

.....

.....

..... [2]

**(ii)** Comment on the use of an array processor architecture for a system in which a number of different calculations need to be performed on the same data item.

.....

.....

.....

..... [2]

**4** **(a)** Draw a diagram to show the binary sort tree obtained by adding the words

hockey, cricket, netball, football, swimming, rugby

in the order given so that they can be sorted into alphabetical order.

[3]

**(b)** The diagram shows a data structure, called MyData, of fixed size. Data is always added to one end of the structure and removed from the other end.

| position | |
|---|---|
| 8 | |
| 7 | |
| 6 | *data* |
| 5 | *data* |
| 4 | *data* |
| 3 | *data* |
| 2 | *data* |
| 1 | |

free = 7

start = 2

Two pointers are used: **start** points to the first data item and **free** points to the free space immediately after the last data item.

**(i)** State the correct name for this type of data structure.

..... [1]

A simple algorithm to add another data item could be
MyData(free) := new_data
free := free + 1

**(ii)** Give a simple algorithm to read and remove one data item from this structure (ignore any errors that could occur).

.....

.....

.....

..... [2]

**(iii)** Explain the term overflow for this data structure.

.....

.....

.....

..... [2]

**(iv)** Explain the difference between static and dynamic data structures.

.....

.....

.....

..... [2]

**(c) (i)** State a characteristic of files that can be merged.

.....

..... [1]

**(ii)** Two files contain the following names.

| | |
|---|---|
| File A | Bill, Katy, Luisa, Majid |
| File B | Ann, Kunal, Omar, Rashmine, Sam, Tom |

Write the result of merging these files.

.....

.....

.....

..... [2]

**(iii)** Write an algorithm to merge two files, stating any assumptions you make.
(No marks will be awarded if you only describe how you merged the files in part **(ii)**.)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

**5** **(a)** Various types of addressing may be used in a low level language.

**(i)** A description of direct addressing is that an instruction gives the address where data can be found.

State **one** disadvantage of direct addressing.

Disadvantage .....

..... [1]

**(ii)** Describe indexed addressing.

.....

.....

.....

..... [2]

**(iii)** State **one** advantage of indexed addressing.

.....

..... [1]

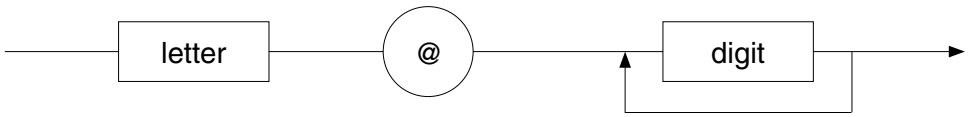
**(b)** In a particular programming language the following features are defined.

digit is 1, 2, 3 or 4 only.
letter is A, B, C or D only.

The symbols @ and # are also defined.

The syntax diagram shown defines a label in the language.

Label:




State why each of the following examples is **not** a label according to this definition.

**(i)** AB@2

.....

..... [1]

**(ii)** A@B

.....

..... [1]

**(iii)** A@45

.....

..... [1]

**(c)** Using the definitions given in **(b)**, draw syntax diagrams to show

**(i)** thing: thing has one # symbol followed by at least one letter

[2]

**(ii)** value: value has one digit followed by any number of @ symbols

[2]

**(iii)** wotsit: wotsit has one @ symbol, then at least one digit, then one letter and at least one # symbol

[3]

**6** A relational database is used to store information at a dogs' home. Staff at the home look after stray and abandoned animals until new owners can be found.

One entity in the database is Dog.

Dog (DogNumber, Name, Age, Breed)

An example is

167, Ben, 4, Border collie.

**(a)** State why DogNumber is underlined.

.....

..... [1]

**(b)** Complete the table and calculate the storage required for the entity Dog if up to 600 sets of data need to be stored.

| Attribute | Data type | Size (bytes) |
|---|---|---|
| DogNumber | | |
| Name | | |
| Age | | |
| Breed | | |

.....

.....

.....

.....

.....

..... [6]

**(c)** Each staff member looks after a number of dogs and each dog may be looked after by a number of staff.

**(i)** State the type of relationship between the entities StaffMember and Dog.

.....

..... [1]

**(ii)** Show the relationship on the entity-relationship (E-R) diagram.


[1]

**(d)** Information about people who want to own a dog is also stored. These people may visit the home more than once before choosing their dog. Data is stored about each meeting between a person and a dog in the entity Meeting.

Part of the Meeting entity is

Meeting (PersonId, DogNumber, Date, Time)

Using the Meeting and Dog tables (relations), explain the use of a foreign key.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(e)** People who want a dog may discuss this with one or more members of staff.
The relationship is shown as


**(i)** Explain why normalisation is needed.

.....

.....

.....

..... [2]

**(ii)** Complete the diagram to show the result of normalisation.


[3]

**(f)** State **two** tasks carried out by a data description language (DDL).

1 .....

.....

2 .....

..... [2]

**7** In a school a particular programming language is used to record facts and rules.
These include

| | |
|---|---|
| teaches (jones, art). | {Jones teaches art} |
| teaches (ahmed, maths). | {Ahmed teaches maths} |
| teaches (cooper, maths). | {Cooper teaches maths} |
| | |
| learns (mary, art). | {Mary learns art} |
| learns (dipesh, art). | {Dipesh learns art} |
| learns (arjun, maths). | {Arjun learns maths} |

worksWith (Teacher, Pupil) if teaches (Teacher, Subject) and learns (Pupil, Subject)

**(a)** State the result of the query

teaches (Teacher, maths)?

.....

.....

.....

..... [2]

In this language, the query worksWith (jones, Pupil)? is solved as follows:

| | | |
|---|---|---|
| *step 1* | attempt to solve | teaches (jones, Subject) |
| step 2 | finds | Subject = art |
| | {set Subject = art} | |
| *step 3* | attempt to solve | learns (Pupil, art) |
| *step 4* | finds | Pupil = mary |
| *step 5* | a solution is | (Subject = art, Pupil = mary) |
| *step 6* | attempt to solve | learns (Pupil, art) |
| *step 7* | finds | Pupil = dipesh |
| *step 8* | a solution is | (Subject = art, Pupil = dipesh) |

**(b)** Using this example to help you, explain the terms

**(i)** fact

.....

.....

.....

..... [2]

**(ii)** goal

.....

.....

.....

..... [2]

**(iii)** backtracking

.....

.....

.....

..... [3]

If you use these lined pages to complete the answer to any question, the question number **must** be clearly shown.