Computer Science Department
1999 Examinations
D0a - MSc Basic Paper 1

Answer ALL questions

(The use of electronic calculators is not permitted in this examination)

1. Answer any **two** of parts a) b) or c)

a) i) What are streams and how does C++ use them to read and write to and from files?

[5 marks]

ii) List the member functions or operators that can be applied to a stream. How do these compare to those that can be applied to the standard input/output streams `cin/cout`?

[4 marks]

iii) Describe the mechanism C++ provides for passing arguments from the command line to a program. Illustrate your answer by writing a `main` function that will list the arguments from the command line. For example, if the program is called `echo_line` the result of invoking it would be:

```
prompt>echo_line Hello world
Hello
world
prompt>
```

[Note that `prompt>` refers to the prompt on the terminal and is not an output of the program.]

[7 marks]

b) Study the following C++ function:

```
int split(int *array, int size, int tolerance)
{
  int pos = 0;
  for (int i = 0; i < size; i++) {
    if (array[i] <= tolerance) {
      int temp = array[pos];
      array[pos] = array[i];
      array[i] = temp;
      pos++;
    }
  }
  return pos;
}
```

[Question 1b) is continued on the next page]

[Question 1b) continued]

i) After execution of the following code fragment, what would be the contents of the variables A, B, m, and n?

```
int A[] = {4, 5, 3, 7, 5, 1};
int B[] = {6, 4, 8, 1, 6};
int m = split(A, 6, 3);
int n = split(B, 5, 5);
```

[6 marks]

ii) What does the function `split` do?

[2 marks]

iii) What are the main reasons for using pointers in C++? How do they differ from references?

[5 marks]

iv) Three consecutive lines in the function `split` swap the contents of two array locations. Write the function `swap` that will replace the three lines and show how it would be invoked.

[3 marks]

c) The class `Telephone` may be used to connect to a telephone network. It has a number of public member functions: `dial, redial, store, recall, hangup.` It also has private data members of type `unsigned long: phoneNumber, lastNumber, memStore`; where `phoneNumber` holds the number for the `Telephone` object, `lastNumber` holds the last number dialled, and `memStore` is an array of size `MEMSIZE` to hold stored phone numbers.

Assuming the type `STATE` has been defined and has values `OK`: (all is well), `NOT_OK`: (there is a problem), `BUSY`: (the network is busy), the following operations are possible:

[Question 1c) is continued on the next page]

[Question 1c) continued]

```
STATE state;
unsigned long number;
Telephone phone(2623496);   // creates a telephone object with the data
                            // member phoneNumber set to the unsigned
                            // argument. The other data members are set to
                            // zero
state = phone.dial(5549282);      // connect to the telephone network using
                                  // the argument
state = phone.redial();     // connect to the telephone network with last
                            // number dialled
state = phone.store(3982634, 3); // stores the number 3982634 at index 3 of
                                 // memStore
number = phone.recall(2);        // recall the phone number at index 2 of
                                 // memStore
phone.hangup();                  // disconnect from the telephone network
```

i) Given that `connectToNetwork(number);` connects to a telephone network using the phone `number` argument and `disconnectFromNetwork();` disconnects from the telephone network, complete the definition of the class `Telephone`:

```
class Telephone {
  public:
    // place public member function declarations here
  private:
    // place private data member declarations here
  private:
    STATE connectToNetwork(unsigned long phone_number);
    void disconnectFromNetwork(void);
};
```

[2 marks]

ii) Define the constructor and the member functions `dial`, `redial`, `recall`, `store`, and `hangup`. The functions should have simple error checking and assume a phone number of zero (0) to signify an invalid number. Minor errors in C++ syntax will not be penalised.

[12 marks]

iii) Give the code fragment for a persistent user who dials a number and, if it is busy, will continually redial it up to 10 times before hanging up.

[2 marks]

2. Answer any **two** of parts a) b) and c)

a) i) Explain how the execution speed of a processor may be increased through the use of *data caches*, *instruction caches* and the technique of *pipelining*.

[12 marks]

ii) Discuss the way in which variable length and branch instructions affect pipelining.

[4 marks]

b) i) The program below is written for a Motorola 68000 series processor with a 32-bit address space. It deals with character strings, where a string is defined to be "a contiguous sequence of bytes terminated by a zero byte". The program below is intended to copy a string starting in memory location 0x40000000 to memory location 0x50000000.

The program includes several errors. Identify these errors and say how they can be corrected. [N.B. do not concern yourself with the efficiency or elegance of the code – concentrate on errors which prevent the program from doing its job]

```
loop:   mov.l    0x40000000,    a0        |1
        mov.l    0x50000000,    a1        |2
        mov.b    (a0),          d0        |3
        bne      end                      |4
        add.b    #1,            a0        |5
        mov.b    d0,            (a1)      |6
        add.b    #1,            a1        |7
        bra      loop                     |8
end:    halt     0                        |9
```

[6 marks]

ii) A stack is implemented on a Motorola 68000 series processor using register `a6` as a stack pointer. The stack holds 32-bit values and grows downwards. Write a program fragment to replace the top two items in the stack by their sum. Your program should make correct use of autoincrement and autodecrement instructions.

[5 marks]

[Question 2b) is continued on the next page]

[Question 2b) continued]

iii) The C++ fragment below shows some simple declarations and a procedure call. Assuming that `myfunc()` declares two local variables and that the procedure call mechanism makes use of a stack, illustrate the state of the stack whilst the procedure is being executed. Indicate what caused each item to appear on the stack.

```
int a = 4;
int b;
int myfunc(int x);
. . .
b = myfunc(a);
```

[5 marks]

c) A hard disc has an average rotational latency of 10ms and an average seek time of 15ms. The disc has 1024 cylinders, 4 surfaces and 48 sectors. Each disc block is 512 bytes.

i) At what speed (in r.p.m) does the disc rotate?

[2 marks]

ii) What is the total capacity of the disc in Mbytes?

[2 marks]

iii) A program reads 12 Kbytes of data from the disc. Estimate how long this will take if the data is in consecutive blocks on one cylinder. (State any assumptions you make).

[3 marks]

iv) Repeat iii) assuming that the data is randomly distributed across the disc. (State any assumptions you make).

[3 marks]

v) A file contains 1000 bytes. A program processes the file one byte at a time. Estimate how long this would take with and without a disc cache. You may assume that all the file is on one cylinder and that CPU processing time is negligible compared with disc access time.

[6 marks]

3. Answer any **two** of parts a) b) and c)

a) The C++ procedure `char inchar()`is used, in the program fragment below, to retrieve one character from the keyboard. When `inchar()` is called, a *trap* instruction is invoked and the calling process is *blocked*.

```
char x;
. . .
x = inchar();
```

i) Explain the terms *trap* and *blocked*.

[3 marks]

ii) Describe the sequence of events that are likely to occur in the calling process and operating system between the call to `inchar()` and its return following keyboard input.

[9 marks]

iii) Explain what is meant by *compute-bound* and *I/O-bound* processes. Which would normally be given the highest priority by the scheduler in a general-purpose operating system? Justify your answer.

[4 marks]

b) A computer has a 32-bit address space. A *memory management* scheme is employed which allocates 22 bits to the page number and 10 bits to the offset.

i) What is the page size?

[1 mark]

ii) What is the page number and offset for the address 0x84008400. (Give your answers in hexadecimal).

[2 marks]

iii) A *memory management unit* (MMU) typically employs an internal cache. What kind of information is stored in this cache?

[2 marks]

[Question 3b) continues on the next page]

[Question 3b continued]

iv) Outline the sequence of events that occurs in each of the scenarios below. In each case, state which tasks are carried out by hardware and which by the operating system.

- Processor does a memory read for a virtual address and the MMU cache search succeeds.
- Processor does a memory read for a virtual address, the MMU cache search fails and the required page is currently in memory.
- Processor does a memory read for a virtual address, the MMU cache search fails and the required page is currently on disc.

[11 marks]

c) i) The access controls applied to a file system can be visualised as a table which specifies who has access rights to which files. In practice, access rights are expressed either as *capabilities* or *access control lists*. Explain what these terms mean.

[7 marks]

ii) Briefly describe the access control scheme used in the Unix operating system.

[5 marks]

iii) Where does Unix store file access control information? What are the implications of this choice in situations where a file has multiple names?

[4 marks]

[END OF PAPER]