

UNIVERSITY COLLEGE LONDON

University of London

EXAMINATION FOR INTERNAL STUDENTS

For the following qualifications :-

B.Sc.

Information Studs. G4: Programming 1

COURSE CODE : **INSTG004**

UNIT VALUE : **0.50**

DATE : **15-MAY-02**

TIME : **14.30**

TIME ALLOWED : **2 hours**

02-C0777-3-70

© 2002 *University of London*

TURN OVER

INSTG004 (“Programming 1”), 2002

Answer **ANY THREE** questions.

1. a) Briefly explain the terms *compiler*, *source code* and *object code*. [3 marks]
- b) Give two examples of *data types* in C++. Give an example of how to convert a piece of data from one type to another. [3 marks]
- c) Briefly explain the difference between the use of a single “=” sign in C++ and the use of a double “==” sign. [2 marks]
- d) What is meant by a *Boolean expression* in C++? What truth values (TRUE or FALSE) do the following two expressions evaluate to?

```
(2 <= 7 || 4 < 9) && 5 > 6  
2 <= 7 || (4 < 9 && 5 > 6)
```

 [5 marks]

- e) A certain theatre currently sells tickets for a full price of £10, but sells tickets for half price to people who are less than 18 years old or who have a season card. Use an “if ... else if ... else” statement to complete the following program, which outputs the ticket price for a single user. The program should output either “incorrect input”, or “Your ticket costs £10.00”, or “Your ticket costs £5.00”. You may assume that the user enters a positive whole number after the first prompt and a single lower case character after the second prompt, and that all user ages up to and including 120 years are reasonable. Your solution should make it as easy as possible to alter the full price of a ticket at a later date.

```
#include <iostream.h>  
  
const float FULL_PRICE = 10.0;  
const int MIN_AGE = 18;  
const int MAX_POSSIBLE_AGE = 120  
  
int main()  
{  
    cout.setf(ios::fixed);  
    cout.precision(2);  
    int user_age;  
    char season_card;  
    cout << "Enter your age: ";  
    cin >> user_age;  
    cout << "Do you have a season card (y/n): ";  
    cin >> season_card;  
  
    add an “if ... else if ... else ...” statement here  
  
    return 0;  
}
```

[7 marks]

[Total 20 marks]

PAPER CONTINUED ON NEXT PAGE

2. a) Briefly explain what is meant by a *function* and a *parameter* in a C++ program. [4 marks]

b) A miracle cure for baldness, called Capellex, has been developed. For maximum effect patients must take an exact dosage dependent on their head circumference and age. The correct dosage is $(40 - X)$ milligrams, where X is the result of dividing the patient's head circumference (in centimetres) by the patient's age (in years), and then multiplying by 5. The following is the “main” part of a program which advises patients how much Capellex to take.

```
#include <iostream.h>

.....
.....

int main()
{
    cout.setf(ios::fixed);
    cout.precision(1);

    float head_circumference;
    int age;
    input_details(head_circumference,age);
    output_medicine_needed(head_circumference,age);
    return 0;
}

.....
.....
```

Complete the program by adding function declarations and definitions for the functions “input_details(...)” and “output_medicine_needed(...)” so that the program will produce the following example input/output:

```
What is your head circumference (in centimetres)? 56.0
How old are you (in years)? 28

Milligrams of Capellex needed: 30.0
```

(The numbers “**56.0**” and “**28**” in bold are user inputs.)

[10 marks]

c) For a cross-channel trip, a certain ferry company charges a basic rate of £60 for a single car with one or two passengers, plus £20 extra per extra passenger. So, for example, a cross-channel trip with 3 passengers costs £80, and a cross-channel trip with 5 passengers costs £120. Write a definition for a C++ function “crossing_cost(...)” which takes a number of passengers as a parameter and returns the price in pounds. For example, the call “crossing_cost(3)” should return 80 and the call “crossing_cost(5)” should return 120. You may assume that input error checking occurs elsewhere in the program so that “crossing_cost(...)” is always passed a sensible value for its parameter.

[6 marks]

[Total 20 marks]

PAPER CONTINUED ON NEXT PAGE

3. a) Briefly explain what an *array* is in C++. [3 marks]
- b) Briefly explain what a *string* is in C++. [3 marks]
- c) A computer system requires each user to have a password, which must be more than 5 and less than 11 characters long. The password can contain any characters except spaces, tabs and new lines. Without using any predefined string functions (such as `strlen`) add a definition for the function “`is_valid(...)`” to the following program:

```
#include <iostream.h>

const int MAX_STRING_LENGTH = 100;

enum Logical {False, True};

Logical is_valid(char password[]);

int main()
{
    char a_password[MAX_STRING_LENGTH];
    cout << "Type in a password: ";
    cin >> a_password;
    if (is_valid(a_password))
        cout << "This is a valid password.";
    else
        cout << "This is not a valid password.";
    return 0;
}

.....
```

so that it reports whether the input string is a valid password or not, as in the following example input/output.

```
Type in a password: Secret
This is a valid password.
```

(The letters in bold are user inputs.) You may assume that the user does not attempt to input a password containing spaces, tabs or new lines.

[7 marks]

- d) The computer system described in part (c) is to be altered so that in addition to being more than 5 and less than 11 characters long, passwords must start with a lower or upper case letter of the alphabet. Again without using any predefined string functions, alter the definition of “`is_valid(...)`” from part (c) accordingly. Again, you may assume that the user does not attempt to input a password containing spaces, tabs or new lines.

[7 marks]

[Total 20 marks]

PAPER CONTINUED ON NEXT PAGE

4. A certain university exam has three possible grades; “fail”, “ordinary pass” and “distinction”. For an ordinary pass, students have to score at least 50. To gain a distinction, they have to score at least 70. A corresponding program asks the user to input the last name of each student followed by their exam score. The program then prints out a list of students with an ordinary pass, followed by a list of students who have gained a distinction. The input information is stored in an array of structures called “student_list”, which is declared in the “main()” program. The first part of the program is as follows:

```
#include <iostream.h>

const int MAXIMUM_NAME_LENGTH = 50;
const int NUMBER_OF_STUDENTS = 10;
const int PASS_GRADE = 50;
const int DISTINCTION_GRADE = 70;

typedef char name_string[MAXIMUM_NAME_LENGTH];

struct student
{
    name_string name;
    int grade;
};

void input_details(student& this_student);
void display_results(student student_list[]);

/*****/

int main()
{
    student student_list[NUMBER_OF_STUDENTS];
    int count;

    for (count = 0 ; count < NUMBER_OF_STUDENTS; count++)
    {
        cout << "STUDENT NUMBER " << (count +1) << ":\n";
        input_details(student_list[count]);
        cout << endl;
    }

    display_results(student_list);

    return 0;
}

/*****/

.....
.....
.....
```

QUESTION CONTINUED ON NEXT PAGE

- a) Write a definition for the function “input_details(...)”. (The corresponding function declaration is already written just above the “main()” part of the program.) You may assume that all the students have different last names and that there will be no input errors. The function should cause the program to have input/output such as:

```
STUDENT NUMBER 1:  
Please enter this student's last name: Jones  
Please enter this student's grade: 56  
  
STUDENT NUMBER 2:  
Please enter this student's last name: Miller  
Please enter this student's grade: 19  
  
.....  
.....  
.....  
  
STUDENT NUMBER 9:  
Please enter this student's last name: Patel  
Please enter this student's grade: 72  
  
STUDENT NUMBER 10:  
Please enter this student's last name: Smith  
Please enter this student's grade: 67
```

(The names and numbers in bold are user inputs.)

[8 marks]

- b) Write a definition for the function “display_results(...)” (The corresponding function declaration is already written just above the “main()” part of the program.) The function should result in output such as:

The following people have gained an ordinary pass:

```
Jones  
Smith
```

The following people have gained a distinction:

```
Patel
```

[12 marks]

[Total 20 marks]

PAPER CONTINUED ON NEXT PAGE

5. The following is part of a program which counts the number of words in the file "essay.txt":

```
#include <iostream.h>
#include <fstream.h>

int main()
{
    char next_character;
    int count = 0;
    ifstream in_stream;

    .....
    .....
    .....

    cout << "There are " << count
         << " words in the file.\n";

    return 0;
}
```

- a) Briefly explain the function of the statements

```
#include <fstream.h>
and
    ifstream in_stream;
```

in the second and seventh (non-blank) lines of the program.

[4 marks]

- b) Complete the program so that it correctly counts the number of words in the file "essay.txt". You may assume that words are separated by one or more blank characters, that there are no blank characters at the very beginning or at the very end of the file, that there is no punctuation in "essay.txt" other than full-stops and commas immediately after some of the words, and that no word is more than 100 characters long.

[8 marks]

- c) Adapt your answer to part (b) so that the program correctly counts the number of sentences in the file "essay.txt". You may assume that full stop characters are used within the file only to end sentences.

[8 marks]

[Total 20 marks]

END OF PAPER