

CSCP1DB

October/November 2015

C++ AS SECOND PROGRAMMING LANGUAGE

Duration 2 Hours

75 Marks

EXAMINERS .
FIRST
SECOND

MRS P LE ROUX
MR CL PILKINGTON

Closed book examination.

This examination question paper remains the property of the University of South Africa and may not be removed from the examination venue.

This paper consists of 8 pages and 8 questions.
Please ensure that you have all 8 pages with the 8 questions.

INSTRUCTIONS:

- Answer all the questions.
- Do all rough work in the answer book.
- The mark for each question is given in brackets next to the question.
- Please answer the questions in the correct order. If you want to do a question later, leave enough space
- Number your answers and label your rough work clearly
- Marks are awarded for part of an answer, so do whatever you are able to in each question.

GOOD LUCK!

[TURN OVER]

QUESTION 1**[2 marks each = 8 MARKS]**

Give the value of x after execution of each of the following code fragments:

1.1 `int x = 2 * 3 + 4 * 5 / 4 / 2,`

1.2 `#include <iostream>
using namespace std;
int main() {
 int x = 5;
 int Quotient = 5;
 Quotient /= ++x;
 cout << "x = " << x << "Quotient = " << Quotient << endl;
 return 0;
}`

1.3 `int A[3] = {1, 2, 3},
int *p = A;
*(p++) = 0;
int x = A[0];`

1.4 `int A[3];
int *p = A;
int x = 0;
for (int i = 0; i < 3, i++)
{
 A[i] = 1;
 x += p[i];
}`

QUESTION 2**[5 MARKS]**

A bookshop gives discounts to customers as follows:

- Students get 10% discount
- Book dealers get 12% discount
- Pensioners get 15% discount
- All other customers get 10% discount only if their total purchases is more than R200

[TURN OVER]

Write down **ONLY** the necessary C++ statements to calculate and display the final amount that is due, after the discount is applied.

Do NOT write a complete program. Use the following variables.

```
float amount;           // The total purchase amount due before discount
char customerType;      // the type of customer: 'S' (student) or
                        // 'D' (dealer) or 'P' (pensioner) or 'O' (other)
```

Assume that values have been assigned to `amount` and `customerType` already.

You may also need the following variables

```
float discount, finalAmount;
```

QUESTION 3**[10 MARKS]**

Professor Knowal uses a two-dimensional array `marks` to store the test results of his first-year Computer Science students. There are 30 students in the class and the students write 5 tests during the course of the year.

- 3.1 Declare an integer constant `NO_STUD` for the number of students and an integer constant `NO_TESTS` for the number of tests. (2)
- 3.2 Declare a two-dimensional array `marks` with `NO_STUD` rows and `NO_TESTS` columns. (2)
- 3.3 Assume that the following declarations are given:

```
int total, highest;
float average;
```

Further assume that values have been given to all the elements of the array `marks`. Use nested `for` loops and write down the necessary C++ instructions to determine and display the highest mark for each test as well as the average for each test.

Do NOT write a complete program or any functions. Write down **ONLY** the required statements (6)

[TURN OVER]

QUESTION 4**[8 MARKS]**

In both parts of this question you have to write the body of a function. In both cases the function header looks as follows:

```
string changedSentence(string senP)
```

Hint: Below the question we list a number of `string` member functions that you may need.

- 4.1 The function receives a string of characters, indicated by `senP` in the function header. The function has to replace all occurrences of the string `he` with `Theo` and return the changed string to the main function.

Example: If the string

```
When he saw the hen, then and there he heard the noise.
```

is given, the string

```
WTheon Theo saw tTheo Theon, tTheon and tTheore Theo Theoard  
tTheo noise
```

should be returned to the main function. You should write the body of the function
ONLY (4)

- 4.2 The function receives a string of characters, indicated by `senP` in the function header. The function has to replace all occurrences of the stand-alone string `he` with `Theo` and return the changed string to the main function. You may assume that `senP` will not start or end with `he`.

Example: If the string

```
When he saw the hen, then and there he heard the noise.
```

is given, the string

```
When Theo saw the hen, then and there Theo heard the noise
```

should be returned to the main function. You should write the body of the function
ONLY (4)

[TURN OVER]

SEE BELOW FOR SOME STRING MEMBER FUNCTIONS

A number of `string` member functions to help you

```
StringObject.size( )  
StringObject.substr(startPos, length)  
StringObject.find(substring)  
StringObject.find(substring, startPos)  
StringObject.insert(insertPos, substring)  
StringObject.erase(startPos, length)  
StringObject.replace(startPos, length, substring)
```

where

`startPos`, `length` and `insertPos` are of type `int`, and
`substring` is of type `string`

QUESTION 5**[8 MARKS]**

5.1 What does the following statement do?

```
vector<int> list (50);
```

 (2)

5.2 Write a C++ statement that declares `secretList` to be a vector object used to store integers. Do not specify the size of `secretList` (2)

5.3 Write C++ statements to store the following values, in the order given, into `secretList`:

```
56, 28, 96, 75
```

 (2)

5.4 Write a for loop that outputs the contents of `secretList`. Use a function to determine the size of `secretList` (2)

[TURN OVER]

QUESTION 6**[6 MARKS]**

Pollen count readings measure the number of pollen grains in the air and are usually in the range of 10 to 200 grains per cubic metre of air. Write a C++ program to create a file named `pollen.dat` containing 10 pollen counts obtained from the user. If the file fails to open, display an appropriate message, e.g. Failed to open the data file.

QUESTION 7**[9 MARKS]**

A keen runner and programmer keeps meticulous records of his training for Comrades. For each training session he records the date, the distance trained, and the duration of the training session, in a file of objects, `Training.dat`.

Define a class `TrainingSession` that represents one such training session. This class has three member variables:

- `date`, a string that holds the date of the session
- `distance`, a double value that indicates the distance covered and
- `time`, a string that indicates the duration of the training session.

In addition, the class should have the following member functions:

- a default constructor
- an overloaded constructor to set `date`, `distance` and `time` to specified values.
- a destructor that does not perform any action
- accessor functions for the member variables
- an overloaded `operator>` to compare two `TrainingSession` objects. The `operator>` is implemented as a `friend` function with the following prototype:
`bool operator>(const TrainingSession & t1, const TrainingSession & t2)`
This function returns `true` if `distance` for `t1` has a larger value than `distance` for `t2`; and `false` if not
- an overloaded extraction `operator >>` (implemented as a `friend` function) so that it can be used to input values of type `TrainingSession`
- an overloaded insertion `operator <<` (implemented as a `friend` function) that outputs all the member variables of a `TrainingSession` object

Write only the header file `Training.h` that contains the specification of the class `TrainingSession`.

[TURN OVER]

QUESTION 8**[21 MARKS]**

It is expected of you to write an application about shapes. The program accepts three values. If the first two values are greater than 0 and the third is equal to 0, the shape is a rectangle. If the third value is also greater than 0, the shape is a box. Furthermore, the program must calculate the area in the case of a rectangle and the surface area and volume in the case of a box.

8.1 Create a base class implemented in files `Rectangle.h` and `Rectangle.cpp` that contains two private data members `length` and `width` of type `double`. The function members of the base class `Rectangle` should consist of:

- a constructor passing two values of type `double` to set the values of the data members
 - a function named `calcArea` that calculates the area of the shape which is the product of `length` and `width`
 - two accessor functions, `getLength()` and `getWidth()` that returns the values of the `length` and `width` data members
- (6)

8.2 From the `Rectangle` class, derive a class named `Box` that has an additional data member named `depth`. The derived `Box` class should have the following member functions:

- a constructor
- a function named `calcArea` that overrides the `calcArea` function in the class `Rectangle` and returns the surface area of a box. The surface area of a box is calculated as follows:

$$2 * [(length * width) + (length * depth) + (width * depth)]$$

- a `volume()` function that calculate the volume of a box as the product of `length`, `width` and `depth`
 - an accessor function, `getDepth()` that returns the value of the `depth` data member
- (8)

8.3 The program below is a working C++ program making use of the classes above. Fill in the missing statements. There are 7 missing statements. Please write only the missing statement down.

(7)

[TURN OVER]

```

#include <iostream>

_____ // Contains the class declarations
using namespace std;

int main()
{
    double  theLength, // Local variable for length
    double  theWidth,   // Local variable for width
    double  theDepth,   // Local variable for depth

    cout << "This program will calculate the area of a\n",
    cout << "shape Enter 0 if there is no depth \n ",
    cout << "What is the width? ";
    cin  >> theWidth;
    cout << "What is the length? ",
    cin  >> theLength,
        cout << "What is the depth? ",
    cin  >> theDepth,

    _____ // Define an instance of the Rectangle class
    _____ // Define an instance of the Box class

    if ( _____ ) // Determine if the shape
                        // is a rectangle or a box
    {
        cout << "Width  " << theWidth << endl,
        cout << "Length  " << theLength << endl;
        cout << "Area:  " << _____ << endl;
        // call the appropriate function to calculate
        // the area
    }
    else
    {
        cout << "Width:  " << theWidth << endl;
        cout << "Length. " << theLength << endl,
        cout << "Depth:  " << theDepth << endl;

        cout << "Area.  " << _____ << endl;
        // call the appropriate function to calculate
        // the area

        cout << "Volume: " << _____ << endl;
        // call the appropriate function to calculate
        // the volume
    }
    return 0;
}

```