

Tutorial Letter 104/3/2015

Introduction to Programming I

COS1511

Semester 1

School of Computing

Examination Tutorial Letter

Contents

1	TUTORIAL MATTER	3
2	INTRODUCTION.....	4
4	PAST EXAMINATION PAPER	8
5	PAST EXAMINATION PAPER MEMORANDUM.....	18

PERCENTAGES of ASSIGNMENTS: We recommend that you check the percentages that you obtained for your assignments on myUnisa and to let us know immediately if there is an error.

1 TUTORIAL MATTER

You should already have received the material listed below. If any of it is missing, please download it from myUnisa – see tutorial letter COS1511/101/3/2015 and COSALLF/301/4/2015.

STUDY GUIDE

DISK 2015 (with software)

TUTORIAL CD

Tutorial letters:

COSALLF/301/4/2015	General information concerning the School and study at Unisa
COSALLF/302/4//2015	Names and telephone numbers of lecturers
COS1511/101/3/2015	Information about COS1511 Assignments
COS1511/102/3/2015	Solutions to exercises in the Study Guide
COS1511/103/3/2015	This tutorial letter contains the instructions on how to create an assignment as a PDF file
COS1511/104/3/2015	Examination tutorial letter - this letter
COS1511/105/3/2015	This tutorial letter contains self-assessment assignment 3 together with its solution.
COS1511/201/1/2015 COS1511/201/2/2015	Solutions to assignment 1
COS1511/202/1/2015 COS1511/202/2/2015	Solutions to assignment 2 available after the due date for Assignment 2

For e-mail, please use the module address namely COS1511-15-S1@unisa.ac.za for semester 1 and COS1511-15-S2@unisa.ac.za for semester 2

You may phone us at any of the following telephone numbers:

Ms P Le Roux 011 670 9118

Mr T Masombuka 011 670 9123

2 INTRODUCTION

Dear COS1511 student

We trust that you are enjoying your introduction to programming as presented in COS1511.

PLEASE NOTE: To have gained admission to the examination you had to submit assignment 1 by 16 March 2015 if you are registered for the 1st semester, and 31 August 2015 if you are registered for the 2nd semester. **There is NO extension for this assignment.**

Calculation of the semester mark

The marks that you obtain for Assignments 1 and 2 form the semester mark for COS1511. The semester mark forms 20% of the final mark for the module. The weights of the COS1511 assignments are indicated in the table below:

Assignment number	Weight
1	20%
2	80%
3	0%

An example follows: Suppose a student gets 60% for Assignment 1 and 45% for Assignment 2. In order to calculate the semester mark, the mark obtained for the specific assignment is multiplied by the weight. This then forms part of the 20% that the semester mark contributes to the final mark. Therefore:

Assignment	Marks obtained	Weight	Contribution to semester mark	
1	60%	20%	$60/100 \times 20/100 \times 20$	2.40
2	45%	80%	$45/100 \times 80/100 \times 20$	7.20
TOTAL				9.60

In this example the student has a semester mark of 9.60 out of 20. **The semester mark will form part of the final mark for a supplementary examination.**

3 EXAMINATION INFORMATION

Your COS1511 exam is a **written** examination. You can obtain a maximum of 75 marks and you will have two hours to complete it. Although the examination is not done on a computer, it is still practical. In almost all questions you have to write parts of a C++ program. Because **no question is answered on a computer**, you should practice writing a program by hand instead of typing it in on a keyboard. This can be very time consuming if one is not used to it.

Previous exam papers are available on *myUnisa* under Official Study Material. At the back of this tutorial letter is a study exam paper that you can work through to prepare for the examination. Please note that the mark allocation per question and the format of the exam paper may not be the same as in previous years or in the study exam paper. The previous exam papers and the study exam paper serve as revision – no questions will be repeated in the 2015 exam paper.

Note that in the past the exam paper included a section with multiple choice questions. In this exam, the exam paper will **not** include any multiple choice questions. However, we still advise you to work through the multiple choice questions in the previous exam papers and the study exam paper as this provide valuable practise and will help you to make sure you understand the study material.

The exam paper consists of 11 questions, which must be answered in your answer books. The format of the May/June and the Oct/Nov paper are the same. The exam questions are discussed below:

All the questions should be answered in the *answer book*. There are eleven questions, each counting a different total. Some of the questions have subsections. Some of these questions involve writing complete functions. Note that in many questions we supply the names and types of the variables that you should use.

You should read the instructions very carefully. Do ONLY what is required. In many cases, for example, parts of a program are supplied. Do not copy those parts to your answer book - the marker will not even look at it. Also, do not input values or validate input except when explicitly instructed to do so. In many cases you will be informed that you should assume that values have already been assigned to specific variables.

The structure of the questions is as follows:

- Question 1: This question tests your understanding of C++ operators. You will have to provide the values of variables after some code has been executed. Although you are not specifically requested to use variable diagrams, it will be a good idea to use them to help you answer this question.

-
- Questions 2 and 3: These questions test your knowledge of conditional execution, i.e. `if` statements, the `switch` statement, `while` or `do..while` loops, `for` loops and Boolean values. You may have to write code or correct code.
 - Questions 4 to 8: In these questions your knowledge of functions and function parameters is tested. You may have to write complete functions.
 - Question 7: This question requires you to trace the program execution of a program with functions by drawing variable diagrams for the program. Please make sure that you know how to draw variable diagrams, and especially how to use variable diagrams to trace the execution of a function. Use the tutorial to Draw Variable Diagrams and chapters 18 to 23 in the Study Guide to prepare for this question.
 - Question 9: One- and two-dimensional arrays and the `for` loop are tested in this question.
 - Question 10: This question deals with structs.
 - Question 11: Manipulation of strings is tested here. You have to write statements that manipulate strings in this question.

Please make sure that you fully understand and are able to use the different control structures, input and output, data types, functions, etc. No part of the study guide may be omitted in your preparation for the examination, except the very last lesson on objects.

The paper may appear to be very long when you receive it. It is, however, not the case. Remember that program listings are given in many of the questions and, in addition, a problem cannot always be explained in one or two sentences only.

We suggest that you take the number of marks allocated to a question as an indication of how much time you should spend on it. You have 1 min 36 sec per mark, so you should for example not spend more than 19 minutes on a question of 12 marks. Please remember that it is possible to obtain some of the marks if you are on the right track, even if you do not get an answer completely correct, so do whichever parts of the question you can do.

Hints:

- In those cases where you have to write complete functions, you should look at the given calling statement to decide what type of function should be written and how many and what type(s) of parameter(s) should be used.
- We spent hours on the tutorial letters that were dispatched to you. Work through the programs. Read through the explanations given with the answers to the assignment questions and some of the exercises in the study guide.
- We want to stress the point that you should read the examination questions very well. **Do only what is required.** For example, if it is stated that values have been assigned to a variable or an array, then you should *not* input values. You should also not validate anything if you are not explicitly instructed to do so.

- Note that there are also old examination papers included under Official Study Material. After you have studied for the exam, work through the previous examination papers on *myUnisa* as well as the study examination paper and memorandum at the end of this paper.

If you encounter a problem during your preparation for the examination, please contact your e-tutor via your discussion forum so that other students can also benefit.

Finally, remember to bring a pen, pencil and eraser with you. You are welcome to answer the paper in pencil as long as you clearly mark your rough work. We wish you all the best for the exam. We hope you pass with a good mark.

Regards,
The COS1511 team

4 PAST EXAMINATION PAPER

This paper consists of 8 pages and 10 questions.

Please ensure that you have 8 pages and 10 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.

ALL THE BEST!

QUESTION 1**8 marks**

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

1.1 `int x = 5;` (2)

`x -= --x;`

1.2 `int x = 2 * 3 + 4 * 5 / 4 / 2;` (2)

1.3 `int x = 0;` (2)

`for (int i = 0; i < 4; i++)`

`{`

`x += (i % 2 ? 1 : 0);`

`}`

1.4 `int j = 2;` (2)

`int k = 3;`

`int m = 2;`

`int x = (k = m < j || 3 - j > k);`

QUESTION 2**8 mark**

2.1 Consider the following C++ code segment. What value will `c` have after the code has been executed? (2)

`int a = 4, b = 2;`

`int c;`

`if (a * b > b * 4)`

`c = 2 * b;`

`else if (b < a)`

`c = b + a * 2;`

`else`

`c = a;`

2.2 Suppose the input value for `type` is the character `'/'`. What is the value of `value` after the following C++ code has been executed? (2)

```
int value = 5;
char type;
cin >> type;
switch (type)
{
    case '+': value += 5;
    case '/': value = 100/value;
    case '-': value = value/2;
    case '%': value += value % 2 + 4;
    default: value += value / 3 + 10;
}
```

2.3 Consider the following C++ code segment below. (2)

```
int result(int valueP)
{
    int count = 0;
    int a = 2;
    while (count < valueP)
    {
        a += count + a / 2;
        count += 2;
    }
    return a;
}
```

What will the output of the program be if the following instruction appears in the main function?

```
cout << result(6);
```

2.4 The video shop has a special discount under the following conditions: (2)

- It is Tuesday or Thursday , AND
- You hire 3 old DVDs.

A string variable `day` represents the day of the week. The int variable `old` represent the number of old DVDs you want to hire. The bool variable `discount` should receive the value `true` if the above conditions are met. Write the statement that will assign the correct value to `discount`?

QUESTION 3**9 marks**

- 3.1 Consider the following nested if code fragment. Note that `icecream` is a string variable and `choice` is an int variable. Convert the code by making use of the switch statement. (4)

```
if (choice == 1)
    icecream = "vanilla";
else if (choice == 3)
    icecream = "chocolate";
else if (choice == 2)
    icecream = "blueberry";
else icecream = "rum & raisin";
```

- 3.2 Convert the following for loop into a while loop. (3)

```
for (int i = 1; i <= n; i++)
    cout << i * i;
```

- 3.3 The following code is supposed to write out the positive even numbers less than 12. That is, it will output the numbers 2, 4, 6, 8, and 10. However, there is a problem. Explain the problem and correct the code. (2)

```
int x = 1;
while (x != 12)
{
    cout << x << endl;
    x = x + 2;
}
```

QUESTION 4**4 marks**

Write function headers for the functions described below.

- 4.1 The `divideByTwo` function receives an integer passed to it and returns the result which may contain a decimal place. (1)
- 4.2 The `getGrossPay` function returns a value of type `float` and has two formal parameters: an `int` variable named `hours` and a `float` variable named `rate`. (1)

-
- 4.3 The function `addAndMultiply` has four formal parameters: two `int` variables named `a` and `b` representing the two values that must be added and multiplied respectively. These values are returned to the calling program by two reference parameter named `added` and `multiply`. (2)

QUESTION 5	3 marks
-------------------	----------------

The program below allows a user to enter the amount of a salesperson's sales. The program should calculate a 5% bonus and then display the bonus. For example, if the sales of a person are R4000; the calculated bonus will be R200. The input is given as an integer. There are three functions, one to read the sales amount from the keyboard, one to do the bonus calculation and one to display the output. Complete the program by filling in the three correct calling statements of the functions in the `main()` function.

```
#include <iostream>
using namespace std;
void getSales(float &salesAmt)
{
    cout << "Enter sales: ";
    cin >> salesAmt;
}
float getBonus (int sold, float bonusRate)
{
    float bonus = 0.0;
    bonus = sold * bonusRate;
    return bonus;
}
void displayBonus(float bonus)
{
    cout << "Bonus : R " << bonus << endl;
}
int main()
{
    float sales = 0;
    float bonus = 0.0;

    //call function to get the sales
    5.1 Add your statement here

    //call function to calculate the bonus
    5.2 Add your statement here

    //call function to display the bonus
    5.3 Add your statement here

    return 0;
}
```

QUESTION 6**16 marks**

Study the program below and answer the questions that follow.

```

1  #include <iostream>
2  using namespace std;
3  int x;
4  void summer(int & a, int b)
5  {
6      int intNum1;
7      intNum1 = a + 12;
8      a = 2 * b + 5;
9      b = intNum1 + 4;
10 }
11 void winter(int u, int & v)
12 {
13     int intNum2;
14     intNum2 = x;
15     v = intNum2 * 4;
16     x = u - v;
17 }
18 int main()
19 {
20     int intNum1 = 2;
21     int intNum2 = 5;
22     x = 6;
23     summer(intNum1, intNum2);
24     cout << intNum1 << " " << intNum2 << " " << x << endl;
25     winter(intNum1, intNum2);
26     cout << intNum1 << " " << intNum2 << " " << x << endl;
27     return 0;
28 }

```

- 6.1 Complete the part of the variable diagram below by adding the missing line number of the next five lines after line 23 and the values of the variables in your answer books. Also, clearly indicate if a value is accessible or not as indicated on the first line of the variable diagram on the next page. (13)

Remember the following conventions used for variable diagrams:

- A question mark ? shows an uninitialised value for a variable.
- The notation 25 → 5 means that execution jumps from line 25 to line 5.
- Use square brackets [] around the name of a variable to show that it is inaccessible while the current function is being executed.

Line 23	intNum1 2	intNum2 5	x 6	[a]	[b]	[intNum1]
line	intNum1	intNum2	x	a	b	intNum1
line	intNum1	intNum2	x	a	b	intNum1
line	intNum1	intNum2	x	a	b	intNum1
line	intNum1	intNum2	x	a	b	intNum1
line	intNum1	intNum2	x	a	b	intNum1

6.2 What will be outputted by line 26? (3)

QUESTION 7	5 marks
-------------------	----------------

Write a C++ void function that receives four `int` parameters: the first two by value and the last two by reference. Name the formal parameters `n1`, `n2`, `sum` and `diff`. The function should calculate the sum of the two parameters passed by value and then store the result in the first variable passed by reference. It should calculate the difference between the two parameters passed by value and then store the result in the second parameter passed by reference. When calculating the difference, subtract the larger value from the smaller value. Name the function `calcSumAndDiff`. Also, write the statement that invokes the `calcSumAndDiff` function, passing it variables called `num1`, `num2`, `numSum` and `numDiff`.

QUESTION 8	9 marks
-------------------	----------------

8.1 Consider the following declarations:

```
const int ARRAYSIZE = 7;
float length[ARRAYSIZE] = {7.8, 6.4, 4.9, 11.2};
```

What is the value of `length[1]` and `length[4]`? (2)

- 8.2 Write a C++ program to input eight integer numbers into an array. As each number is entered, add the numbers into a total. After all the numbers are entered, calculate the average and display the numbers and the average. Name the array `grades` and use the variables `total` and `average` to determine the total marks and average.

(3)

- 8.3 Assuming the following declarations:

```
const int NUMROWS = 3;
const int NUMCOLS = 4;
int val[NUMROWS][NUMCOLS]={8,16,9,52,3,15,27,6,14,25,2,10};
```

- 8.3.1 Write a statement to change the value of the element with the value 27 to 55.

(1)

- 8.3.2 Write a nested for loop to add 5 to each value in the table.

(3)

QUESTION 9**8 marks**

Consider the following struct definition:

```
struct TShirts
{
    string brand;
    int quantity;
    float price;
};
```

- 9.1 Define two members of type `TShirts` named `s1` and `s2`. Initialize `s1` with `addidas`, 100 and `R360.00`.

(2)

- 9.2 Given this structure type definition and declarations above, what output will be produced by the code below?

(2)

```
cout << s1.quantity << " " << s1.brand << " @ R " << s1.price
    << endl;
s2 = s1;
s2.price = s2.price/3;
cout << s2.quantity << " " << s2.brand << " @ R " << s2.price
    << endl;
```

- 9.3 Write a function `readShirtRecord` which accepts a reference parameter `new_tShirt` of type `TShirts`. `readShirtRecord` fills `new_tShirt` with values entered from the keyboard.

(4)

QUESTION 10**5 marks**

Write C++ statements to do the following string manipulations.

(See list of `string` member functions on the next page.)

(1 mark each)

- 10.1 Assign the number of characters contained in the `message` variable to an `int` variable named `numChars`.
- 10.2 Use the `erase` function to remove the first two characters from the `message` variable.
- 10.3 Rewrite the code for 10.2 using the `replace` function.
- 10.4 Assign the first four characters of a string variable named `address` to a string variable named `streetNum`.
- 10.5 Change the contents of the `word` variable from “mend” to “amend”.

Member functions of the string class

The following member functions are provided with the `string` class to manipulate the values of `string` objects:

Function signature	Description
<code>int size()</code>	Returns the size (i.e. length) of a string object.
<code>string substr(int, int)</code>	Returns a substring of a string object. The first parameter specifies the starting position (i.e. the position from which the substring should be copied) and the second parameter specifies how long the substring should be (i.e. how many characters should be copied). The second parameter may be omitted, in which case the sub-string consisting of all the characters from the starting position (specified by the first and only parameter) to the end of the string are returned.

<code>int find(string, int)</code>	Returns the position of a string (specified as the first parameter) within a string object. The second parameter is optional, and can be used to specify where the search has to be commenced. If omitted, the search commences at the beginning of the string object. If the string being sought is not found, -1 is returned.
<code>void insert(int, string)</code>	Inserts a string (specified as the second parameter) into a string object at a particular position (specified as the first parameter)
<code>void erase(int, int)</code>	Erases a substring from a string object. The substring that is to be erased is determined by the two parameters: from the position specified by the first parameter, as many characters as specified by the second parameter
<code>void replace(int, int, string)</code>	Replaces specified characters of a string object with another string. The characters to be replaced are determined by the first two parameters: from the position specified by the first parameter, as many characters as specified by the second parameter. The string to be inserted in their place is specified by the third parameter.

5 PAST EXAMINATION PAPER MEMORANDUM

QUESTION 1

8 marks

- | | | | |
|-----|---|----|-----|
| 1.1 | 0 | ✓✓ | (2) |
| 1.2 | 8 | ✓✓ | (2) |
| 1.3 | 2 | ✓✓ | (2) |
| 1.4 | 0 | ✓✓ | (2) |

QUESTION 2

8 marks

- | | | | |
|-----|---|----|--------|
| 2.1 | 10 | ✓✓ | (2) |
| 2.2 | 28 | ✓✓ | (2) |
| 2.3 | 13 | ✓✓ | (2) |
| 2.4 | bool discount = ((old == 3) && (day == "Tu" day == "Th")); | | ✓✓ (2) |

QUESTION 3

9 marks

3.1 switch (choice) 1/2 (3)

```
{
    case 1 1/2 : icecream = "vanilla"; 1/2
                break; 1/2
    case 2 : icecream = "blueberry";
                break;
    case 3 : icecream = "chocolate";
                break;
    default: 1/2
                icecream = "rum & raisin"; 1/2
}
```

3.2 int i = 1; ✓ (3)

```
while (i <= n) ✓
{
    cout << i * i << endl;
    i++; ✓
}
```


QUESTION 7**5 marks**

```
void calcSumAndDiff (int n1 1/2,
                    int n2 1/2,
                    int & sum 1/2,
                    int & diff1/2)
{
    sum = n1 + n2; 1/2
    diff = n1 - n2; 1/2
}

int numSum;
int numDiff;
int num1 = 12;
int num2 = 7;

calcSumAndDiff(num1, num2, numSum, numDiff); ✓✓
```

QUESTION 8**9 marks**

8.1 length[1] = 6.4 ✓

length[4] = 0 ✓ (2)

8.2 (3)

```
int grades[8];
int total = 0;
double average;
for (int i = 0; i < 8; i++) ✓
{
    cin >> grades[i];
    total = total + grades[i]; ✓
}
for (int i = 0; i < 8; i++)
{
    cout << grades[i] << " " ;
}
cout << endl << "The total is " << total << endl;
cout << endl << "The average is " << total/8 << endl; ✓
```

8.3.1 val[1][1] = 55; ✓ (1)

8.3.2

```
for (int i = 0; i < 3; i++) ✓ (3)
{
    for (int j = 0; j < 4; j++) ✓
```

```

        val[i][j] + 5;
    }

```

✓

QUESTION 9**8 marks**

9.1

```

tShirts s1, s2; 1/2
s1.brand = "addidas"; 1/2
s1.quantity = 100; 1/2
s1.price = 360.00; 1/2

```

(2)

9.2

```

100 addidas @ R 360 ✓
100 addidas @ R 120 ✓

```

(2)

9.3

```

void readShirtRecord(tShirts & new_tShirt) ✓✓
{
    cout << "Enter the brand of the tShirt (string): ";
    cin >> new_tShirt.brand;
    cout << "Enter the quantity of tShirts (integer): ";
    cin >> new_tShirt.quantity;
    cout << "Enter the price of the tShirt (double): ";
    cin >> new_tShirt.price; ✓✓
}

```

(4)

QUESTION 10**5 marks**

```

10.1 numChars = message.size(); ✓
10.2 message.erase(0,2); ✓
10.3 message.replace(0,2,""); ✓
10.4 streetNum = address.substr(0,4); ✓
10.5 word.insert(0,"a"); ✓

```