Tutorial Letter 103/1/2017
Introduction to Programming I

COS1511

Semester 1

School of Computing

Examination Tutorial Letter

Contents

1 INTRODUCTION

Dear COS1511 student

We trust that you are enjoying your introduction to programming as presented in COS1511. 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. See tutorial letter 101 on how to calculate your semester mark.

2 EXAMINATION INFORMATION

Your COS1511 exam is a ***written*** examination. You can obtain a maximum of 100 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.

The exam paper consists of 2 sections. Section A is multiple choice questions to be answered in your answer book. Section B consist of 6 questions, which must also be answered in your answer books. The exam questions are discussed below:

All the questions should be answered in the *answer book*. All the questions counts 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.

You can use what follows as a rough guide of the structure of the questions in Section B:

- **Question 1**: This question tests 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.

- **Question 2**: In this question your ability to write coded is tested. You have to complete code.

- **Question 3**: In this question your knowledge of variables, functions and function parameters is tested.

- **Question 4**: In this question your knowledge of arrays is tested.

- **Question 5**: Here your programming skills are tested and you have to define a structure and write code to use it.

- **Question 6**: In this question you have to write statements that manipulate strings.

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

3 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;
x -= --x;
```
(2)

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

1.3
```
int x = 0;
for (int i = 0; i < 4; i++)
{
    x += (i % 2 ? 1 : 0);
}
```
(2)

1.4
```
int j = 2;
int k = 3;
int m = 2;
int x = (k = m < j || 3 - j > k);
```
(2)

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  {
21     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.

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

6.2

	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
line						

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
`int size()`	Returns the size (i.e. length) of a string object.
`string substr(int,int)`	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.

`int find(string,int)`	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.
`void insert(int,string)`	Inserts a string (specified as the second parameter) into a string object at a particular position (specified as the first parameter)
`void erase(int,int)`	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
`void replace(int,int,string)`	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.

4 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 (3)

```
switch (choice) ½
        {
            case 1 ½  :  icecream = "vanilla";     ½
                      break;     ½
            case 2      : icecream = "blueberry";
                      break;
            case 3      : icecream = "chocolate";
                      break;
            default: ½
                     icecream = "rum & raisin";½
    }
```

3.2 (3)

```
int i = 1;     √
while (i <= n) √
{
cout << i * i << endl;
i++;     √
}
```

1.3 Endless loop √ (2)

```
int x = 2;  √  or  while (x <= 12)
```

QUESTION 4 **4 marks**

4.1 `double ½ divideByTwo(int Amt ½)`

4.2 `double ½ getGrossPay (int hours, double rate ½)`

```
4.3     void ½ addAndMultiply( int a, int b,        ½
                               int & added,         ½
                               int & multiplied     ½)
```

QUESTION 5 **3 marks**

5.1 `getSales(sales);` √

5.2 `bonus = getBonus(sales, 0.05);` √

5.3 `displayBonus(bonus);` √

QUESTION 6 **10 marks**

6.1 (1 mark for each line = 7)

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

	intNum1	[intNum2]	x	a	b
line 6 23->6	?		6	2	5
Line 7	14		6	2	5
Line 8	14		6	15	5
line 9	14		6	15	18

	intNum1	intNum2	x	[a]	[b]
line 24	15	2	6		

6.2 `15 24 -9` √ √ √ (3)

QUESTION 7 **5 marks**

```
void calcSumAndDiff   (int n1    ½,
                       int n2    ½,
                       int & sum ½,
                       int & diff½)
{
    sum = n1 + n2;    ½
    diff = n1 - n2;   ½
}

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;  ½
s1.brand = "addidas"; ½
s1.quantity = 100;         ½
s1.price = 360.00;         ½
```
(2)

9.2 (2)

```
100 addidas @ R 360   √
100 addidas @ R 120   √
```

9.3 (4)

```
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;              √√
}
```

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");` √