Tutorial letter 202/1/2017

Introduction to Programming I

COS1511

Semester 1

School of Computing

This tutorial letter contains the solutions and discussion of assignment 2

BAR CODE

Learn without limits.

UNISA university of south africa

ONTENTS

Page

INTRODUCTION

If you followed the guidelines given in the questions, got the same output for the given input data, and used good programming techniques, your programs need not be identical to ours.

In the Study Guide we introduced certain conventions, amongst others

- that names of variables and the names of functions should start with lowercase characters and further words in the names with uppercase characters,
- that the names of void functions should be verbs (or contain a verb) and
- that the names of value-returning functions should be nouns or adjectives.

Also

- use constants where needed
- avoid the use of global variables
- the correct indenting is extremely important
- remember to include comments

The mark allocation is as follows:

Question 1	10
Question 2a	5
Question 2b	5
Question 3	15
Question 4a	15
Question5	15
Question 6a	10
Question 6b	12
Question 6c	13
Question 7	Not marked
Total	100

For e-mail, please use the module address namely by using Course Contact on myUnisa. You may phone me at 011 670 9118.

Best wishes….

Question 1

You are requested in the (a) part of the question to complete the `while` loop in order to make a valid selection between `0` and `4`.

Remember that the `while` loop can execute `0` or more times. Thus you have to ensure that the loop is entered. This is done by ensuring the variable `selection` has a value before the condition in the `while` loop is evaluated. You have to

1. Declare the variable `selection` as follows: `int selection;`
2. There is a statement `cin >> selection` just before the `while` loop ensuring `selection` has a value.
3. Complete the `while` loop. Remember to that you do not want to enter the loop if `selection` has the value of `0, 1, 2, 3` or `4` as it is valid values. You want to enter the loop if it is any other value (less than `0` **OR** greater than `4`).

```
      while (selection < 0 || selection > 4)
```

In the second part of the question you have to add code to calculate the cost per order. After a valid selection has been made, thus the value of selection is either `0, 1, 2, 3` or `4`, the user must enter the number of hotdogs required. You have to

1. Declare the variable. As it is a whole number the type of the variable must be `int` and you can choose any name, e.g. `noDogs` or `numberOfDogs`, etc. Remember to make it descriptive as it contributes to readability of your code and stick to the conventions of naming a variable: `int numberOfDogs;`

2. The next step is to write the `switch` statement. The selector in the `switch` statement must be of an ordinal type. We want to switch depending on the type of hotdog ordered and therefore the statement will be: `switch (selection)`

3. The different cases must now be completed. In each case, only the amount used to calculate the cost is different. There are several ways of doing this. One way is shown below.

```
float price = 0;     //variable  to  hold  the  price  of  each  item
float cost;          //variable to hold the price of each item

switch(selection)
{
        case 1: price = 15.0;
                break;
        case 2: price = 12.50;
                break;
        case 3: price = 17.00;
                break;
        case 4: price = 22.50;
                break;
        case 0: cout << " Please come again" << endl;
                break;
        default:
                cout << "Invalid selection";
                cout << " Try again please" << endl;

}
cost = selection * price;
cout << "The total cost is R " << cost << endl;
             OR
cout << "The total cost is R " << selection * price << endl;
```

4. To format the output which contains a float value , you have to include the statements
```
    cout.setf(ios::fixed);
    cout.precision(2);
```

5. The file **`question1.cpp`**

```
    #include <iostream>
    using namespace std;

    int main()
    {
        int selection;     ✓          // variable to hold the choice of hotdog
        cout  << "Please enter the choice of hotdog "
              << "(a number from 1 to 4 or 0 to quit) " << endl;
        cout  << "Hot Dog Menu "     << endl << endl;
        cout  << "1: Plain hotdog "  << endl;
        cout  << "2: Chili hotdog " << endl;
        cout  << "3: Cheese hotdog " << endl;
        cout  << "4: Russian hotdog "<< endl;
        cout  << "0: QUIT " << endl <<endl << endl;

        cin   >> selection ; ½
        while (selection < 0 || selection > 4) ✓
        {
            cout << "Invalid choice - Please re-enter ";
            cin >>  selection; ½
        }
        cout << "You have selected option " << selection << endl;

        int noOfDogs = 0; ✓   // variable to hold the number of hotdogs ordered
        float cost = 0;       // variable to hold the cost of the order

         if (selection !=0 )
         {
            cout << "How many hotdogs would you like?" << endl;
            cin  >> noOfDogs; ✓
         }

         cout.setf(ios::fixed);
         cout.precision(2);

         switch(selection)
        {
            case 1: cost = noOfDogs * 15.0;      ½
                    cout << "The total cost is R " << cost << endl;
                    break;
            case 2: cost = noOfDogs * 12.50;     ½
                    cout << "The total cost is R " << cost << endl;
                    break;
            case 3: cost = noOfDogs * 17.00;     ½
                    cout << "The total cost is R " << cost << endl;
                    break;
            case 4: cost = noOfDogs * 22.50;     ½
                    cout << "The total cost is R " << cost << endl;
                    break;
            case 0: cout << "Please come again" << endl; ½
                    break;
            default:                             ½
                    cout << "Invalid selection";
                    cout << "Try again please" << endl;
            }
            return 0;
    }
```

Output 2 marks ✓✓

Question 2

Question 2a

You are required to run the code and change the `for` loop to a `while` loop. When running the code, the first observation and error is that `n` is not declared. Thus the declaration below must be added:

```
int n = 10;
```

The output: `XXX`

Discussion: The code will be more readable if { } are added as follows:

`for (int i = 1; i <= n; i++)`
`     if (i < 5 && i != 2)`
`          cout << 'X';`
	`for (int i = 1; i <= n; i++)`
	`{`
	`     if (i < 5 && i != 2)`
	`          cout << 'X';`
	`}`

A variable diagram will assist in understanding the output:

i	**(i < 5 && i != 2)**	**Output**
1	2 < 5 && 2 != 2 true && true => true	X
2	2 < 5 && 2 != 2 true && false => false	
3	3 < 5 && 3 != 2 true && true => true	X
4	4 < 5 && 4 != 2 true && false => false	X
5	5 < 5 && 5 != 2 false && true => false	
6	6 < 5 && 6 != 2 false && true => false	
7	7 < 5 && 7 != 2 false && true => false	
8	8 < 5 && 8 != 2 false && true => false	
9	9 < 5 && 9 != 2 false && true => false	
10	10 < 5 && 10 != 2 false && true => false	

To change the `for` loop to a `while` loop you must remember that the variable in the condition of the `while` loop must change otherwise the loop will never exit. Thus, `i` must be declared and must be changed inside the while loop. Here it is necessary to add the { }.

```
int i = 1;      ✓✓
while (i <=10)  ✓✓
{
     if (i < 5 && i != 2)
          cout << 'X';
          i++;✓
}
```

Question 2b

You are required to examine the semantics of the code.

When running the code, the sum produced is 19. A variable diagram will assist in understanding the output:

next	sum	(next <= 5)	next++	sum + next
2	1	2 <= 5 => true	3	4
3	1	3 <= 5 => true	4	8
4	1	4 <= 5 => true	5	13
5	1	5 <= 5 => true	6	19
6	1	6 <= 5 => false		

The explanation of the logical error must refer to two issues. The first is the initial values of the variable `sum` ✓and second is where the value of the variable in the condition changes. ✓

Thus changing the code as below, will yield the correct answer.

```
#include <iostream>

using namespace std;
int main()
{
    int next = 2, sum = 0; ✓
    while (next <= 5)
    {

        sum = sum + next;
        next++;✓✓
    }
    cout << "The sum of 2 through 5 is " << sum << endl;
    return 0;
}
```

Question 3

Question 3a

The following steps must be followed to add statements to the existing code:

1. We have to declare a variable to hold the number of days of the long weekend – `numberOfDays`.
2. Add a `cout` statement to inform the user to enter the number of days of the weekend, as well as a `cin` statement to read in the value.
3. Change the control loop to the variable declared.
   `(day = 1; day <= numberOfDays; day++)`
   Or
   `(day = 0; day < numberOfDays; day++)`
4. Remember to change the denominator to calculate the average.

```
#include <iostream>
using namespace std;
int main()
{
    int numStudents;
    float numHours, total, average;
    int student,day = 0;
    int numberOfDays;              Step1
    cout  << "This program will find the average number of hours a day"
          << " that a student spent programming over a long weekend\n\n";
    cout  << "How many students are there?" << endl;
    cin   >> numStudents;

    cout  << "Enter the number of days in the long weekend:" << endl;
    cin   >> numberOfDays;                                              Step2

    for( student = 1; student <= numStudents; student++)
    {
        total = 0;
                                                                  Step 3
        for(day = 1; day <= numberOfDays; day++)
        {
            cout << "Please enter the number of hours worked by student"
            << student <<" on day " << day << "." << endl;
            cin >> numHours;
            total = total + numHours;
        }                                                        Step 4

        average = total / numberOfDays;
        cout   << endl;
        cout   << "The average number of hours per day spent programming "
               << "by student " << student << " is " << average
               << endl << endl << endl;
    }
    return 0;
}
```

Question 3b

The following steps must be followed to add statements to the existing code:

1. We have introduced four variables: `totalB` and `averageB` to hold the total number of hours spent on Biology, and `totalP` and `averageP` for the total number of hours spent on Programming.
2. The two totals are calculated. Notice the way the total is calculated, e.g. `totalB = totalB + numHours;` Two values (`totalB` and `numHour`) are added together and the final answer is assigned to `totalB`.
3. We know that `numHours` has been entered by the user, but what about `totalB`? We need to make sure it is initialised before it is used.
4. Calculation of the averages. There is no need to initialise the average variable here. Can you see why?
5. Comparison of the two variables and the correct message is displayed.

The file **`question3.cpp`**

```
#include <iostream>
using  namespace std;

int main()
{
      int numStudents,  numOfDays;
      float numHours,  totalB,½ totalP, ½ averageB, ½ averageP½;
      int student,day =  0;

      cout << "This program will  find  the  average number of  hours  a day"
           << "that  a student spent programming over a long weekend\n\n";
      cout << "How  many   students are there ?" <<  endl <<  endl;
```

```
        cin  >> numStudents; ½
        cout << "Enter  the  number of days in  the long weekend\n\n";
        cin  >> numOfDays; ½

        for(student =  1; student <=  numStudents; student++)
        {
             totalB = 0;     // Biology         ½
             totalP = 0;     // Programming     ½

             for(day =  1;  day <=  numOfDays; day++)✓
             {
                  cout <<  "Please enter the number of  hours  worked by   student"
                       <<  student << " on day (biology) " <<  day <<  " . "
                       <<  endl;
                  cin >>  numHours; ½

                  totalB =  totalB +  numHours; ✓

                  cout <<  "Please enter the number of  hours  worked by   student"
                       <<    student << " on day (Programming) " <<  day <<  " . "
                       <<    endl;
                  cin   >>    numHours; ½

                  totalP      =  totalP +  numHours; ✓
             }

             averageB =  totalB / numOfDays; ½
             averageP =  totalP / numOfDays; ½

             cout <<  endl;

             if(averageB  >  averageP)   ✓
                  cout << "Student " << student
                       << "spent most of the time on Biology" << endl; ½
             else
                  cout << "Student " << student
                       << " spent most of  the  time on Programming" << endl; ½
        }

        return 0;
}
```

Output 4 marks

Question 4

Question 4a

Notice that we had to include `<iomaip>` in order to use `setw()` function. We also used "`math.h`" to compute the squares and cubes of `i`, otherwise we could have used `i*i` and `i*i*i` instead of `pow(i,2)` and `pow(i,3)` respectively. As you can see, it is easier to read the `pow()` function than counting the number of `i`'s to understand to what power it is.

```
void printTabs()
{
     cout <<"NUMBER  SQUARE CUBE"  <<  endl;
     for(int  i = 1; i <= 10; i++)
          cout << setw(4) << i << " " << setw(5) << pow(i,2) << " "
               << setw(5) << pow(i,3)<<endl;
 }
```

Question 4b

Notice the statement (`incr > 1)? i+=incr : i++;` . This is a shorthand for writing the `if` statement. If the condition `(incr > 1)` is true the statement `i+=incr` is executed, if it is false `i++` is executed. We could have written this as:

```
if(incr > 1)
     i+=incr;
else
     i++;
```

The file **question4.cpp**

```
#include <iostream>
#include <iomanip> √
#include "math.h"

   using  namespace std;

   void selTabs(int  st,  int stp, int incr) ½ ½ ½
   {
        int i =  st; √

        while(stp!=0) √
        {
            cout   <<  i <<  "\t"<<  pow(i, 2)<< "\t"<<  pow(i,3)<<endl; √
            stp--;√
            (incr > 1)? i+=incr : i++;√√
        }
   }

   int main()
   {
        int start,  stop, increment;
        cout   << "Enter the starting  value of the table"<<endl;
        cin    >> start; ½
        cout   << "Enter the number of values to be displayed - " << endl;
        cin    >> stop; ½
        cout   << "Enter the increment between the values -  " << endl;
        cin    >>  increment; ½

        selTabs(start,  stop, increment); √
        return 0;
   }
```

Output 4 marks

Question 5

The manager of the Crosswell Carpet Store has asked you to write a program to print customers' bills.

```
#include <iostream>
#include <iomanip>
using namespace std;

const float labourPerHour = 24.0;
const float salesTax = 0.14;

// The function calculateCarpetSize takes as formal parameters the length and width
// of a room to be carpeted as floating point values and returns an integer
// representing the size of the room in square meters.
int calculateCarpetSize( float lengthP, float widthP)
{
    int roomSize;
    roomSize = lengthP * widthP;                              √
    return roomSize;
}

// The function calculateCarpetCost takes as formal parameters the size of the room
// as well as the selling price of the carpet and returns the cost of the carpet.
float calculateCarpetCost( int sizeP, float sellP )   {
        float carpetCost;
        carpetCost = sizeP * sellP;                                      √
        return carpetCost;
}

// The function calculateLabourCost takes the size of the room as formal parameter
// and calculates the labour cost. The labour cost is equal to the number of square
// meters purchased times R24.00. Note that no tax is charged on labour
float calculateLabourCost( int sizeP)
{
        float labour;
        labour = sizeP * labourPerHour;           √
        return labour;
}

// The boolean function qualifyForDiscount uses the customer number to determine if
// the customer qualifies for a discount. All customers identified by a customer
//number starting with a zero ('0') qualify for discount.
bool qualifyForDiscout(string custNoP)
{
    if (custNoP[0] == '0')                     √
    else
        return false;
}

// The function computeDiscount returns the discount amount for which the customer
// qualifies. It first prompts the user for the discount percentage and then
// calculates the discount.
float computeDiscount()
{
    float discountAmt;
    cout << "\n Please enter the discount amount: ";
    cin  >> discountAmt;                                      √
    return discountAmt;
 }
```

```
// The function printCustomerStatement takes as formal parameters the customer name,
// the customer number, the carpet and labour cost as well as the discount and prints
// the statement.
void printCustomerStatement( string custNameP, string custNoP, float cCostP,
                             float lCostP, float discountP)       √
{

        float subTotal1  = cCostP + lCostP;
        float subTotal2  = subTotal1 - discountP;                  √
        float tax = (subTotal2 * salesTax);

        cout.width(25);cout << "CROSWELL CARPET STORE" << endl;
        cout.width(20);cout << "STATEMENT" << endl;

        cout.width(20);cout << "Customer name : " << custNameP << endl;
        cout.width(20);cout << "Customer number : " << custNoP << endl;
        cout.width(20);cout << "Carpet prize : " << cCostP << endl;
        cout.width(20);cout << "Labour : " << lCostP << endl;
        cout.width(20);cout << "Subtotal : " << subTotal1 << endl;
        cout.width(20);cout << "Less disocunt : " << discountP << endl;
        cout.width(20);cout << "Subtotal : " << subTotal2 << endl;
        cout.width(20);cout << "Plus tax : " << tax << endl;
        cout.width(20);cout << "Total : " << subTotal2 + tax << endl;

}

int main()
{
    // Declare variables
    string customerNo;
    string customerName;
    float roomLength;
    float roomWidth;
    float sellingPrice;
    float carpetCost;
    float labourCost;
    float discount;
    int carpetSize;

    cout << setprecision(2) << fixed;
    cout << "\n Please enter the following information: ";
    cout << "\n Customer name: ";
    cin >> customerName;
    cout << "\n Customer number: ";
    cin >> customerNo;
    cout << "\n The length of the room: ";
    cin >> roomLength;
    cout << "\n The width of the room: ";
    cin >> roomWidth;
    cout << "\n The carpet selling price: ";
    cin >> sellingPrice;
    carpetSize = calculateCarpetSize( roomLength, roomWidth );      ½
    carpetCost = calculateCarpetCost( carpetSize, sellingPrice ); ½
    labourCost = calculateLabourCost( carpetSize );                 ½

    if (qualifyForDiscout(customerNo))    ½
        discount = computeDiscount();     ½
    else
        discount = 0.0;                   ½
```

```
      printCustomerStatement(customerName, customerNo, carpetCost,
                                   labourCost, discount);
     return 0;

} // end main
```

Output 5 marks

Question 6

Question 6a

In this question you had to write three functions. The void function `getData` in our program below has three reference parameters as it is changing the values of the three parameters. The second function `calculatePay` has two value parameters and uses the values of these parameters to calculate the pay and then returns the pay as a `float` value. Function `printPaySlip` has four value parameters as it only prints (displays) the values of the parameters.

```
// Calculate and print payslips

#include <iostream>
#include <iomanip>

using namespace std;

const float workingHours = 40.0;

void getData( string &employeeP, float &hoursWorkedP, float &payRateP )  √
{

     cout << "Please enter the employee name: ";
     getline(cin, employeeP, '\n');                           ½
     cout << "Please enter the hours worked : ";
     cin  >> hoursWorkedP;                                    ½
     cout << "Please enter the pay rate     : ";
     cin  >> payRateP;

    cin.get();   // to extract the enter after reading payRateP - see activiy 7 p80
}
float calculatePay( float hoursWorkedP, float payRateP )      √
{
     float pay;

     if ( hoursWorkedP > workingHours  )
     {
         int overTime = hoursWorkedP - workingHours ;        ½
         float overTimePay = overTime * payRateP * 1.15;     ½
         pay = (workingHours * payRateP) + overTimePay;      ½
     }
     else
         pay = hoursWorkedP * payRateP;                      ½

     return pay;
}
```

```
void printPaySlip( string employeeP, float hoursWorkedP,
                      float payRateP, float payP )           √
{
     cout << endl <<"Payslip for " << employeeP << endl;
     cout << "Hours worked : " << hoursWorkedP << endl;

     if (( hoursWorkedP - workingHours) > 0 )
        cout << "Overtime hours : " << hoursWorkedP - workingHours << endl;
    else                                                                      √
        cout << "Overtime hours : 0 " << endl;

     cout << "Hourly pay rate : " << payRateP << endl;
     cout << "Pay : R" << payP << endl;
     cout << endl;
}

int main( )
{
    string theEmployee;
    float theHoursWorked;
    float thePayRate;
    float thePay;

    cout << "Calculation and printing of pay" << endl;
    cout << "--------------------------------" << endl << endl;

    for (int i = 0; i < 5; i++)√
    {
        getData(theEmployee, theHoursWorked, thePayRate);
        thePay = calculatePay(theHoursWorked, thePayRate);
        printPaySlip(theEmployee, theHoursWorked, thePayRate, thePay);
    }
    return 0;

}
```

Output 2 marks

Question 6b

```
     #include <iostream>
     using namespace std;

     float½ discountA(char customer½, char popcorn½)
     {
         float discount = 0.0;

         if (customer == 'P'|| customer == 'S')
         {
             if (popcorn == 'Y')
                 discount = 0.20; ½
             else
                 discount = 0.10; ½
         }
         return discount; ½
     }
```

```
float discountB(char popcorn½)
{
    float discount = 0.0;

    if (popcorn == 'Y') ½
       discount = 0.05; ½

    return discount; ½
}


int main ()
{

    float ticketPrice = 80.00;
    float amount, finalAmount;
    char customer, popcorn; C:\A_MODULES\COS1511\2017\Assignment 2 S1\q6b.cpp

    cout << "Please enter customer type"    << endl;
    cout << " \t\t 'p' (pensioner)"         << endl;
    cout << " \t\t 's' (student)"           << endl;
    cout << " \t\t 'r' (regular)"           << endl;
    cin  >> customer; ½

    cout << "Do you want some popcorn (y/n) ";
    cin  >> popcorn; ½

    customer = toupper(customer);
    popcorn  = toupper(popcorn);

    if (customer == 'P' || customer =='S') ½
        amount = discountA(customer,popcorn); ½
    else if (customer == 'R') ½
            amount = discountB(popcorn); ½
    else
        cout << "Incorrect customer entered" << endl;

    finalAmount = ticketPrice - (ticketPrice * amount); √

    cout.setf(ios::fixed);
    cout.precision(2);

    cout  << endl << "The amount due by the customer is : R"
          << finalAmount << endl;

    return 0;
}
```

Output 3 marks

Question 6c

```
#include <iostream>
using namespace std;

string ½ getName()
{
    string name, surname;

    cout << "Please enter your first name......" ;
    cin  >> name; ½
    cout << "Please enter your last name......." ;
    cin  >> surname; ½

    return name + ' ' + surname; ½
}

float getHours(string theName, ½ float theRate½)
{
    float grossPay, nettPay;
    int hours, totalHours = 0;

    cout << "Please enter number of hours worked for Monday......." ;
    cin  >> hours;
    totalHours += hours;

    cout << "Please enter number of hours worked for Tuesday......" ;
    cin  >> hours;
    totalHours += hours;

    cout << "Please enter number of hours worked for Wednesday...." ;
    cin  >> hours;
    totalHours += hours;                                                  √√
    cout << "Please enter number of hours worked for Thursday....." ;
    cin  >> hours;
    totalHours += hours;

    cout << "Please enter number of hours worked for Friday......." ;
    cin  >> hours;
    totalHours += hours;



    grossPay = theRate * totalHours; ½

    if (totalHours < 40) ½
       nettPay = grossPay - (grossPay * 0.10); ½
    else
        if (totalHours > 40) ½
            nettPay = grossPay + (grossPay * 0.10); ½
        else
            nettPay = grossPay; ½
    return nettPay;
}
```

```
int main ()
{

    string fullName;
    float pay;

    fullName = getName();½
    pay = getHours(fullName, 80.00); ½

    cout.setf(ios::fixed);
    cout.precision(2);

    cout    << "Employee " << fullName << " is paid : R" << pay << endl; √

    return 0;
}
```

Output 3 marks

Question 7

Note that double and float data types can be used to deal with decimal numbers. The difference between the two is the maximum number of significant digits (numbers after the comma), called precision. The float type handle 6 or 7 and double type can take 15. Thus, you can use double type if you need accuracy to more than 6 or 7 decimal places, otherwise float type will be ok.

Question 7a
Yes. The number and the type of parameters match.

Question 7b
Yes. The number and the type of parameters match.

Question 7c
No. The number of parameters does not match.

Question 7d
The first function header is invalid. The last parameter cannot be a reference as the float value R200.53 is sent in the function call.

The second function header is valid.