

CS118 Programming Fundamentals

Monday, December 24, 2018

Course Instructor

Mr. Rizwan ul Haq, Ms. Zain Iqbal, Mr. Tahir Farooq

Serial No:

Final Term Exam

Total Time: 120 Min

Total Marks: 85

Signature of Invigilator

_____ Roll No	_____ Section	_____ Signature
------------------	------------------	--------------------

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. This paper consists of two parts (**Objective + Subjective**). **Subjective** part can be given to student during the **objective** part time after collecting the **objective** part.
2. Verify at the start of the exam that you have a total of Six (6) questions printed on Six (6) pages including this title page.
3. Attempt all questions on the question-book and in the given order.
4. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as 'useful in the paper' or else there may a charge of cheating.
5. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
6. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
7. Use only your own stationery.
8. **Use of Calculator is not allowed.**
9. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking

PART B (SUBJECTIVE)

Question	Objective	1	2	3	4	5	6	Total
Points	41	10	10	5	5	4	10	85
Score								

Vetted By: _____ Vetter Signature: _____

University Answer Sheet Required: No ☒ Yes ☐

Q1.

10

Write a function in C++ to count and display the total number of vowels (a, e, i, o, u) and consonants (other alphabets in abc that are not vowels) present in a text file "INPUT.TXT" and write your result in text file "COUNT.TXT". For Example if the file "INPUT.TXT" contains the following text. Then "COUNT.TXT" will have Vowels=12 and Consonants= 19

"INPUT.Txt"
There are 10 girls.
It is sunny outside.

"COUNT.TXT"
Vowels = 12
Consonants =17

```
void countVowels()
{
    ifstream inFile;
    ofstream outFile;
    int consonentCount = 0, vowelCount = 0;
    char ch;
    inFile.open("input.txt");
    outFile.open("count.txt");
    while(!inFile.eof())
    {
        inFile.get(ch);
        if(!inFile.eof())
            if(ch=='a' || ch=='A' || ch=='e' || ch=='E' || ch=='i' || ch=='I' || ch=='o' || ch=='O' || ch=='u' || ch=='U')
                vowelCount++;
            else if((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))
                consonentCount++;
    }
    outFile << "Vowels = " << vowelCount << endl ;
    outFile << "Consonents = " << consonentCount;
}
```

Q2.**10**

Write down a program that will take 'n' number of **positive integers from the user** and outputs the average of the numbers, first, second and third largest number from the input integers. N will be at-least 3 and will never exceed 100. The user can enter at-least 3 and at-most 100 numbers. The program **will stop taking input as soon as it finds a -ve input**. The main program will call a function "findAverage" to return the average of the numbers in double. Another function named "findLargerOnes" will return the largest, second largest and third largest numbers.

Note: User will not enter how many number of input he want to give.

Hint: Use static array or a dynamic array.

```
#include "iostream"

using namespace std;

double findAverage(int [], int );
void findLargerOnes(int[], int, int&, int&, int&);

int main()
{
    int myArray[100]; // as the maximum numbers can be 100
    int temp = 0;
    int count = 0, largest, secondlargest, thirdlargest;
    while(count < 100 && temp >= 0)
    {
        cout << "Please enter a number : " ;
        cin >> temp;
        if (temp > 0)
        {
            myArray[count] = temp;
            ++count ;
        }
    }
    if(count < 3)
        cout << "You entered too few numbers" << endl;
    else
    {
        cout << "Average = " << findAverage(myArray, count) << endl;
        findLargerOnes(myArray, count, largest, secondlargest, thirdlargest);
        cout << "Largest = " << largest << endl
             << "Second Largest = " << secondlargest << endl
             << "Third Largest = " << thirdlargest << endl;
    }
    return 0;
}

double findAverage(int arr[], int size)
{
    double sum = 0.0;
    for(int i=0 ; i<size ; i++)
        sum+=arr[i];
    return sum / size;
}

void findLargerOnes(int arr[], int size, int& largest, int& secondLargest, int& thirdLargest)
{
    largest = secondLargest = thirdLargest = -1;
    for(int i=0 ; i<size; ++i)
    {
        if(arr[i]>largest)
```

```

        {
            thirdLargest = secondLargest;
            secondLargest = largest;
            largest = arr[i];
        }
        else if(arr[i] > secondLargest)
        {
            thirdLargest = secondLargest;
            secondLargest = arr[i];
        }
        else if(arr[i] > thirdLargest)
        {
            thirdLargest = arr[i];
        }
    }
}

```

Q3.

5

Write a function named "rotate_left" that takes as its arguments the following:

- 1) An array of floating point values
- 2) An integer that tells the number of cells in the array

The function should shift the contents of each cell one place to the left, except for the contents of the first cell, which should be moved into the last. Thus, for example, if the array passed to the function looks like this:

0	1	2	3	4
1.5	2.6	1.8	9.5	5.5

Then when the function returns, the output will be following:

0	1	2	3	4
2.5	1.8	9.5	5.5	1.5

```

void rotate_left(double arr[], int size)
{
    double temp = arr[0];
    for(int i=0; i<size-1; ++i)
    {
        arr[i] = arr[i+1];
    }
    arr[size-1] = temp;
}

```

Q4.**5**

Write a C++ Function to print the half diamond star pattern using for loop(s)

Example**Input**

Input size: 5

Output

```
  *
 * *
* * *
* * * *
* * * * *
* * * *
 * * *
  * *
   *
```

```
void printHalfDiamond(int x)
{
    for(int i=0 ; i<x ; ++i)
    {
        for(int j=0 ; j<x ; ++j)
        {
            if(j < (x-i-1))
                cout << ' ';
            else
                cout << '*';
        }
        cout << endl;
    }
    for(int i=0 ; i<x-1 ; ++i)
    {
        for(int j=x ; j>0 ; --j)
        {
            if((x-j-1) < i)
                cout << ' ';
            else
                cout << '*';
        }
        cout << endl;
    }
}
```

Q5.	4
------------	----------

Write a function to swap value of two variable using Pointers.

```
void swap (int * a, int* b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Q6.

10

Write down a program that will read the data from a file "sales.txt" containing the information of sales of 2 products at 3 store. All the values are integers. The file contains the information in the form of a tabular format given underneath. Where each row represents sale at store1, store2, store3 for a product.

5	10	6
3	5	10

The program will then read the price of two products from the user and output the product wise sales at different stores.

In the above given input if the user input rate of the product1 and product2 to be 10 and 20 respectively, the output should look like.

	Store1	Store2	Store3
Product 1:	50	100	60
Product 2:	60	100	200

Hint: Use Two dimensional arrays

```
#include "iostream"
#include "iomanip"
#include "fstream"
using namespace std;

int main()
{
    int sales[2][3] = {0};
    int price1, price2;
    ifstream inFile;
    inFile.open("sales.txt");
    if(inFile.is_open())
    {
        for(int i=0 ; i<2 ; ++i)
            for( int j=0 ; j<3 ; ++j)
                inFile >> sales[i][j];
        cout << "Please enter price of Product 1 : " ;
        cin >> price1 ;
        cout << "Please enter price of Product 2 : " ;
        cin >> price2 ;
        cout << std::right<< setw(20) << "Store 1" << setw(9)
            << "Store 2" << setw(9) << "Store 3" << endl ;
        for(int i=0 ; i<2 ; ++i)
        {
            cout << std::left << "Product " << i+1 << setw(2) << ": " << std::right ;
            for( int j=0 ; j<3 ; ++j)
            {
                cout << setw(9) << sales[i][j]*price1;
            }
            cout << endl;
        }
    }
    return 0;
}
```