Lab No.0 Procedural Programming Refresher

0.1 Introduction

Computer Programming course in previous semester is based on writing set of instruction in a particular sequence such that the output is based on the sequence of written statements. If this sequence of instructions is modified then it could have a large impact on the resulting program. Often a situation arises where we intend to use a particular chuck of code again and again in a single program. In this scenario using procedural programming, the programmer is forced to copy-paste the code in various locations of the source file.

Although this is a solution, it is not practical one because it causes the code to become length, unmanaged, poorly structured, and difficult to understand. In such a situation, function handling offers an attractive and easy to implement alternative that is widely accepted and promoted by programmers of all languages.

This is pre-lab for the object-oriented programming course on the use of functions, pointers, structures, and building logic. The functions that are designed will receive variables and arrays from the user. The function will perform some processing on the provided data and return the results back to the main program. The importance of this pre-lab can be highlighted through the fact that functions play critical role in establishment of classes, objects and their implementation. A large portion of object oriented programming is based on manipulating variable using functions. Also in this pre-lab, the use of use of pointes has been emphasized. Concepts relating to pointers, memory access, using arrays with pointer have been covered.

0.2 Objectives of the lab

Reinstate the concepts of structured programming such as

1. decision making
2. iterations
3. functions
4. structures and arrays
5. pointers

0.3 Activities

0.3.1 Activity

Write a program that takes integer array and reverses its elements. Take size of array and contents of array from user in main(). Write function **printArray()** for printing and **revArray()** for reversing elements of the array. Both functions take two arguments: size and integer array in array-notation. Note that actual input array elements are to be swapped in **revArray()** i.e. for array having size 5, swap array element 0 with element 4, swap array element 1 with element 3, and so forth. Temporary array for reversing the elements is not allowed. Print input array before and after reversal using **printArray()**. Pointer-notation is not allowed in this activity.

0.3.2 Activity

Write a program that creates **Student** structure. It consists of three data members: name (character array of size 50), age (integer variable), and cgpa (floating-point variable). It consists of no-argument constructor **Student()**. It consists of three-argument constructor **Student(char n[], int a, float c)** to initialize the three data members. It consists of **show()** function to display the three data members. In main, create two Student variables: s1 and s2. To store data in s1, use three-argument constructor. To store data in s2, access the three Student data member directly. Once initialized, display the contents of s1 and s2 using show() function.

0.3.3 Activity

Write a program that finds the largest of randomly generated data. Specifications of program are as follows:

1. Take array length from user.
2. Declare integer array of user-specified length in step 1.
3. Use rand function to randomly generate data within range of 0 to 100 and store in an array using pointer-notation. Display the data using pointer-notation. Use *for* loop for traversing.
4. Find and display the largest number using pointer-notation.

0.3.4 Activity

Write a program to find transpose of an nxn matrix. Declare integer matrix and its transpose using calloc(). Use 2D pointers in this activity for storage, retrieval, and finding transpose. Free

the pointer variables in the end.

0.4 Testing

Test Cases for Activity 1

Sample Inputs	Sample Outputs
Array Length: 3 Array Contents: 11 22 33 Array Length: 5 Array Contents: 1 2 3 4 5	33 22 11 5 4 3 2 1

Test Cases for Activity 2

Sample Inputs	Sample Outputs
Student 1 Name: Ali Khan Age: 22 CGPA: 3.12 Student 2 Name: Ramiz Shah Age: 21 CGPA: 3.5	Student 1 details Student 2 details

Test Cases for Activity 3

Sample Inputs	Sample Outputs
Array Length: 5 Array Length: 10	Largest of randomly generated numbers Largest of randomly generated numbers

Test Cases for Activity 4

Sample Inputs	Sample Outputs
Matrix Length: 2x2 Matrix Contents: 5 9 4 3 Matrix Length: 3x3 Matrix Contents: 1 2 3 4 5 6 7 8 9	 Transpose: 5 4 9 3 Transpose: 1 4 7 2 5 8 3 6 9

0.5 Tools

Code::Blocks or Dev-C++ or Eclipse

0.6 References

1 **Object-Oriented Programming in C++ by *Robert Lafore***

2 **How to Program C++ by *Deitel & Deitel***