# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BEEE** **11 (Grp1+2)**

**Lab 3:****Dynamic Memory Allocation (2)**

**Date: 19th February, 2024**

**Time: 10:00 am – 12:50 am**

# Lab Engineer: Anum Asif

# Lab 3: Dynamic Memory Allocation (2)

**Introduction**

This lab is about dynamic memory allocation. Dynamic memory allocation in C/C++ refers to performing memory allocation manually by programmer. Dynamically allocated memory is allocated on **Heap** and non-static and local variables get memory allocated on **Stack**

**Objectives**

This lab will revise the old concepts taught to the students in the previous semesters.

**Tools/Software Requirement**

Visual Studio C++

**Description : Static vs Dynamic Arrays**

Consider two variants of declaring arrays below. Memory for the first variant gets allocated on the Stack. The lifetime of an array created using the method A depends on its scope. If it is defined globally, its life is equal to the lifetime of the application. If it is declared in a function, memory for it gets allocated on the stack when the function gets called. It gets deallocated when the function call terminates. All the data related to the function call including the array gets removed from the stack. On the other hand, memory for the array created using new operator gets allocated on the heap at runtime. The lifetime of such an array is at max equal to the execution time of the application. If the array is no more required, the memory allocated for it can be freed using **delete []** command.

**Method A:**

const int size=5;

int x[size];

for (int i = 0; i < size; i++)

{

//cout << "x[" << i << "] = ";

x[i] = i + 1;

}

**Method B**

int size; // Note that size variable is const in variant A whereas it isn’t in variant B. Find out the logic behind it.

cout << "Enter size of array: ";

cin >> size;

int \*x = new int[size];

for (int i = 0; i < size; i++)

{

//cout << "x[" << i << "] = ";

x[i] = i + 1;

}

**Lab Tasks**

**Task 1**

Rewrite this program using pointers in place of arrays (use dynamic memory allocation operators new and delete). The syntax is int \*salArray= new int[size]; and for deletion delete [] salArray.

include<iostream>

using namespace std;

int main(void)

{

int salary[20];

inti;

for (i=0;i<20;++i)

{

cout<<"Enter Salary: ";

cin>>salary[i];

}

for (i=0;i<20;++i)

salary[i]=salary[i]+salary[i]/(i+1);

return 0;

}

**Task 2**

Complete the two parts for analyze pointer problem in the Lab\_3\_-\_Problem\_2.cpp file.

Part 1: Write a function void analyze\_pointer(int \*ptr) that does two things:

* Write the memory location pointed by the pointer to the console.
* Write the value of the integer (which the pointer points to) to the console.

Part 2: Use the function to complete two tasks:

* Allocate an int on the stack (e.g., “int iValue;"), assign a value to it, and get its memory location (with the reference operator—&) to pass this value to analyze\_pointer.
* Allocate an int on the heap (with the new operator). Assign a value to it, and pass it to analyze\_pointer.

Explain all the outputs (results of cout<< in the main() method) and the reasons behind getting those outputs.

**Task 3**

Define a struct Area that has two private variable members; units of type string and area\_value of type float. Modify the Lab\_3\_-\_Problem\_3.cpp program to create a dynamic variable of type Area.

* Input from the keyboard the area\_value and its units. Compute one-half and one quarter of the area and display the results
* Destroy the dynamic variable at the end

**Deliverables**

Compile a single word document by filling in the solution part and submit this Word file on LMS. The name of word document should follow this format. i.e. **YourFullName(reg)\_Lab#.** Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS.

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.