**National University of Computer and Emerging Sciences **

**Laboratory Manual # 06**

**Object Oriented Programming**

| Course Instructor | Mr. Uzair Naqvi |
| --- | --- |
| Lab Instructors | Seemab Ayub , Muhammad Hashir |
| Section | BCS-2B |
| Date | 12-March-2024 |
| Semester | Spring-24 |

**Instructions for lab submission:**

You have to submit source code (.cpp) files along with a word document. In the word document you have to give the heading of each exercise/question, then paste your source code and output snippet. Save your word document in the following format: roll number-lab no-section i.e. 22l-0008-lab6-BCS2B.

**Objectives:**

● Constructors, destructors

● Copy Constructors

● CStrings and dynamic arrays

● Shallow and deep copy

**1. Exercise Marks: 10**

Create a class named **StringInfo** that represents a string along with its length. The class should have the following specifications:

➔ A private member attribute **char\* str** to store the string.

➔ An integer attribute **length** to store the length of the string.

➔ Implement a constructor that takes a const char\* as an argument and initializes the str member with a deep copy of the provided string. The constructor should also calculate and store the length of the string.

➔ Implement a destructor to free the dynamically allocated memory. ➔ Provide a method display() to display the stored string along with its length.

Create a main() function and show the functionality of **StringInfo** class.

**2. Exercise Marks: 10** In this exercise you have to implement a class **StringArray** which manages a dynamic array of C-strings.

#include <iostream>

#include <cstring>

class StringArray {

private:

char\*\* strings;

int size;

public:

StringArray(int arraySize) {

*// Constructor to initialize the StringArray with a given size* }

~StringArray() {

*// Destructor to free the dynamically allocated memory* }

void setString(int index, const char\* str) {

*// Method to set a string at a specific index*

}

const char\* getString(int index) {

*// Method to get the string at a specific index*

}

};

int main() {

StringArray stringArray(3);

stringArray.setString(0, "Hello");

std::cout << "String at index 0: " << stringArray.getString(0) << std::endl;

return 0;

}

**3. Exercise Marks: 10**

Create a class **StringHolder** which is designed to hold a dynamically allocated string. The class has the following specifications:

➔ It has private member attribute **char\* string\_ptr** to store a dynamically allocated string.

➔ It should have a constructor that takes a **const char\*** as an argument and dynamically allocates memory to store a copy of the string.

➔ Implement a copy constructor to perform a ***deep copy*** of the dynamically allocated string (**string\_ptr**) when an object is copied.

➔ Implement a copy constructor to perform a ***shallow copy*** of the dynamically allocated string (**string\_ptr**) when an object is copied (You can overload a constructor to differentiate).

➔ Implement a function setString(char \* str) to set or update the string\_ptr. ➔ Implement a destructor to free the dynamically allocated memory when an object is destroyed.

➔ Provide a method display() to display the string held by the object.

Your task is to write the C++ code for the StringHolder class based on the specifications above. Write a main function and create multiple objects and show the working of shallow copy and deep copy. Update the string of one object and display the string for both objects (copied by shallow and deep copy).