National University of Computer and Emerging Sciences

Laboratory Manual

Object Oriented Programming

Course Instructor	
Lab Instructors	Aqib Zeeshan
Section	
Date	6- March -2025
Semester	Spring-25

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. 22l0008- 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

Array of Objects: Define a class called **Student** with private attributes **name** and **age**. Implement getter and setter methods for both attributes. Create an array of pointers to **Student** objects dynamically. Prompt the user to enter the details (name and age) of each student, store them in the objects using setter methods, and then display the details of all students using getter methods.

1. 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).