National University of Computer and Emerging Sciences

Lab Manual

for

Object Oriented Programming

Course Instructor	Ms. Arooj Khalil
Lab Instructor(s)	Ms. Mamoona Akbar Ms. Saleha Batool
Section	OOP BSE-B
Semester	Spring 2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 3

Objectives:

After performing this lab, students shall be able to:

✔ Create overloaded constructors
✔ Create destructors
✔ Handle dynamic memory for class data members as well as declaring objects.
✔ Copy Constructor
✔ Shallow vs deep copy constructor

Problem 1

Exercise 1:

- Create a class Date having following private data members:

  Int Day
  Int Month
  Int Year
- Create an object of Date "date1" and run your program

Exercise 2 [Default Constructor]:

- Write a default Constructor of Date that initializes the object to 1st January 1926 and prints "Default Constructor Called" in start.
- Now run your program and test what does date1 prints?

Exercise 3 [Print Function]:

- Implement a function Print in Date class which prints a date in following format: dd/mm/yyyy (e.g. 1/1/1926 for date1)
- Print object date1 in your main function and run the program.
- What does it print and how can we initialize the data of date1 at the time of creation?

Exercise 4 [Input Function]:

- Write a function Input in your Date class that takes input from user to populate a Date object. • Call "date1.Input()" and "date1.Print()" in your driver program and test it.

Exercise 5 [Setters]:

- Create an object xmasDay using default constructor.
- Print xmasDay and see what it prints.
- Write Setters i.e. SetDay, SetMonth and SetYear in your class.
- Now set xmasDay to 25/12/2020 using Setters in main.

Exercise 6 [Getters]:

- Write Getters i.e. GetDay, GetMonth and GetYear in your date class.

  Now print xmasDay using Getters in your Driver program.

Problem 2

i) Write a function char** AllocateMemory(int& rows, int& cols) that takes the size of matrix (rows and columns) from the user, allocates memory for the matrix and returns its pointer.
ii) Write a function void InputMatrix(char** matrix, const int rows, const int cols) which takes input the

values in matrix from user(console).
iii) Write a function void DisplayMatrix(char** matrix, const int & rows, const int &cols) that displays the matrix in proper format.

iv) Write a function that does the following:
Creates three dynamic char arrays namely alphabets, numbers, and specialchar.(Define the sizes yourself).
Iterate the 2D array and separate alphabet elements and save them in the alphabets array,
separate number elements and save them in numbers array and separate special character
elements and save them in the specialchar array.
The function returns the three arrays alphabets, numbers, and specialchar.
Note: The three arrays should not consume any extra space. Resize the arrays
accordingly.
For example, if the Sample Matrix is
A 1 v @
+ 9 s 6
P # ^ 4
Your function will return the following arrays:
alphabets = A v s P
numbers = 1 9 6 4
specialchar = @ + # ^