**National University of Computer and Emerging Sciences **

**Laboratory Manual # 08**

**Object Oriented Programming**

| Course Instructor | Mr. Uzair Naqvi |
| --- | --- |
| Lab Instructors | Seemab Ayub , Muhammad Hashir |
| Section | BCS-2B |
| Date | 2-April-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-lab8-BCS2B.

**Objectives:**

In this lab students will practice:

● Composition

**1. Exercise: Marks: 10** a. Define classes for Engine (with horsepower and type) and Car (with model, year, and Engine object).

b. Include constructors, accessors, and mutators in both classes.

c. In main(), create a Car with a specific engine type and horsepower (using composition), and display its information.

**2. Exercise: Marks: 10** Design classes to model a simple recipe system with ingredients and instructions. Classes:

1. Ingredient:

○ Member variables:

■ Name (string) (e.g., "eggs", "flour")

■ Quantity (double) (e.g., 2.0, 1 cup)

■ Unit (optional: enum: grams, cups, tablespoons, etc.)

○ Member functions:

■ Constructor(s) (taking name, quantity, and optional unit)

■ Accessors for name, quantity, and unit (if applicable)

■ (Optional) Method to display ingredient information

2. Instruction:

○ Member variables:

■ Step number (integer)

■ Instruction text (string)

○ Member functions:

■ Constructor(s) (taking step number and instruction text)

■ Accessors for step number and instruction text

3. Recipe:

○ Member variables:

■ Name (string) (e.g., "Chocolate Chip Cookies")

■ Description (optional: string)

■ List of Ingredient objects (composition)

■ List of Instruction objects (composition)

○ Member functions:

■ Constructor(s) (taking name and optional description)

■ Accessors for name, description, ingredients, and instructions

■ Method to add an Ingredient object to the recipe (consider checking for duplicates if needed) // function will get the details for ingredient and create and the ingredient object

■ Method to add an Instruction object to the recipe // function will get the details and create/ add the object to the list

■ Method to display the entire recipe details

Implementation:

Implement the classes with constructors, accessors, and any optional methods as described. Implement the main function and show the create at least one recipe.

Note: Destructor of Recipe class should destroy all ingredients and instruction objects. **3. Exercise: Marks: 10**

You are developing a smart home management system in C++ to control various smart devices installed in a house. Implement three classes: Device, SmartHome.

Device class:

● Member variables:

● deviceId (integer, unique identifier for each device)

● deviceType (string, indicating the type of device e.g., "Light",

"Thermostat", "SecurityCamera")

● status (boolean, indicating whether the device is turned on or off) ● Constructor(s) to initialize the device's identifier, type, and status. ● Accessors to retrieve the device's identifier, type, and status.

● Method to toggle the status of the device (i.e., turn on if off, turn off if on). SmartHome class:

● Member variables:

● devices (List of Device objects)

● Method to add a new device to the smart home system.

● Method to remove a device from the system by its identifier.

● Get the device Id and remove (destroy) device object ● Method to display all devices in the smart home system.

Implement above classes and main function to demonstrate it’s working. ��