\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC121 PYTHON Programming**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 14 **OBJECT-ORIENTD PROGRAMING [PART 2]**

# Objectives

In this lab assignment, students will learn:

- How to create derived classes

- How to write code to access members in base class

- How to write code to override methods in base class

- How to create and use abstract classes

# Goals

In this lab assignment, students will demonstrate the abilities to:

- Create derived classes

- Write code to access members in base class

- Write code to override methods in base class

- Create and use abstract classes

# Instruction and Problems

Create a Python project for each of the following problems. Zip each Python project into a zip file. Submit the zip files to Blackboard for credit.

## Problem 1

A Chinese restaurant has two types of dinner combos for customers to choose. A regular dinner combo includes main dish and soup. A deluxe dinner combo includes main dish, soup and appetizer. There are three main dish choices: sweet and sour pork ($7), sesame chicken ($8) or shrimp fried rice ($9). There are two soup choices: egg drop soup ($1.25) or wanton soup ($1.50). There are two appetizer choices: spring roll ($1.25) or chicken wing ($1.50). They need a program to place orders. You Python project needs to follow these requirements:

1. Create a Dinner\_combo class for regular dinner combos. This class has three protected instance variables: main\_dish, soup and total. Define a choose\_dish method to choose a main dish, a choose\_soup method to choose a soup, and a displayOrder method to display items ordered and total amount due.
2. Create a Deluxe\_dinner\_combo class for deluxe dinner combos. This class is a derived class of the dinner\_combo class. It has one additional protected instance variables: appetizer. Define a choose\_appetizer method to choose an appetizer, and a displayOrder method to display items ordered and total amount due.
3. In the main module, ask user to choose either regular or deluxe dinner combo. Create an object and call its methods to input food items and display information of the order.

The following shows a sample test run. This customer chooses regular dinner combo.

Enter 1 for dinner combo or 2 for deluxe dinner combo: 1

Enter 1 for egg drop soup [$1.25] or 2 for wanton soup [$1.50]: 1

Enter 1 for sweet and sour pork [$7], 2 for sesame chicken [$8] or 3 for shrimp fired rice [$9]: 2

Items ordered: egg drop soup , sesame chicken

Please pay this amount: 9.25

The following shows a sample test run. This customer chooses deluxe dinner combo.

Enter 1 for dinner combo or 2 for deluxe dinner combo: 2

Enter 1 for spring roll [$1.25] or 2 for chicken wing [$1.50]: 1

Enter 1 for egg drop soup [$1.25] or 2 for wanton soup [$1.50]: 2

Enter 1 for sweet and sour pork [$7], 2 for sesame chicken [$8] or 3 for shrimp fired rice [$9]: 3

Items ordered: spring roll , wanton soup , shrimp fried rice

Please pay this amount: 11.75

Zip your Python project and submit it to Blackboard for credit.

## Problem 2

Write a program to create two types of utility bills: water bill and electricity bill. Both types of bills have customer’s name and address. They calculate charge differently. Your project must follow these requirements.

1. Create a Utility\_bill class. This class has two protected instance variables: name and address. The \_\_init\_\_ method takes customer’s name and address as two arguments and stores them in the instance variables. Add another protected instance variable total and initialized it to 0. Define two abstract methods calculate\_charge and display\_bill. These abstract methods have no real code. There is only one statement to raise a NotImplementedError exception.
2. Create a Water\_bill class. This class is a derived class of the Utility\_bill class. It has an additional protected instance variable to store number of gallons of water used. Define a calculate\_charge method, which asks the user to enter number of gallons of water used and uses it to calculate total charge. Customers pay $0.005 per gallon for the first 6000 gallons, and $0.007 per gallon after the first 6000 gallons. Define a display\_bill method to display customer’s name, address, number of gallons of water used and total charge.
3. Create a Electricity\_bill class. This class is a derived class of the Utility\_bill class. It has an additional protected instance variable to store kilowatt hours used. Define a calculate\_charge method, which asks the user to enter kilowatt hours used and uses it to calculate total charge. Customers pay $0.12 per kWh for the first 500 kWh, and $0.15 per kwh after the first 500 kWh. Define a display\_bill method to display customer’s name, address, number of kWh used and total charge.
4. In the main module, ask user to enter name and address. Ask the user to choose either water bill or electricity bill. Create an object and call its calculate\_charge method to calculate total charge. Call the display\_bill method to display the bill.

The following shows a sample test run. This user chooses water bill.

Enter name: John Doe

Enter address: 123 Happy Rd, Raleigh NC, 27603

Enter 1 for water bill, 2 for electricity bill: 1

How many gallons of water were used? 6002

Water Bill

Name: John Doe

Address: 123 Happy Rd, Raleigh NC, 27603

Gallons used: 6002.0

Please pay this amount: 30.014

The following shows another sample test run. This user chooses electricity bill.

Enter name: John Doe

Enter address: 123 Happy Rd, Raleigh NC, 27603

Enter 1 for water bill, 2 for electricity bill: 2

Enter kilowatt hours used: 502

Electricity Bill

Name: John Doe

Address: 123 Happy Rd, Raleigh NC, 27603

Kilowatt Hours used: 502.0

Please pay this amount: 60.3

Zip your Python project and submit it to Blackboard for credit.

# Grading rubric for problem 1

Dinner\_combo class [15 points]

Deluxe\_dinner\_combo class [15 points]

Main module [12.5 points]

# Grading rubric for problem 2

Utility\_bill class [15 points]

Water\_bill class [15 points]

Electricity\_bill class [15 points]

Main module [12.5 points]