COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY
OBJECT ORIENTED PROGRAMMING – CT-260, SPRING 2023
ASSIGNMENT 1 – CLO 2, MARKS 5
RELEASE DATE: 07-05-2023, DEADLINE: 21-05-2023

Guidelines:

- 1 Mark on each question. In total there are 4 questions. 1 Mark is on proper submission of all the files, with required format by following the submission guidelines, before the deadline.
- In case of missing details or confusion, you can make assumptions. However, your assumptions must not contradict with the actual statements.
- Submission must include:
    - Code files (.cpp) zipped in a folder. Each file should have name with student ID, Assignment and Question number. For example: “CT-22034-A1-Q1.cpp”, “CT-22034-A1-Q2.cpp” format.
    - A pdf file which includes output screenshots for each question. You can take screenshot by pressing windows + print screen buttons or you can use snipping tool. You can have multiple screenshots for a single question. You can store all screen shots in a word file and then can save it as a pdf.
    - Each output must contain your ID and name on the screen.
- Plagiarism is punishable with zero marks in the task.
- Late submissions are not allowed.

1. Create a *SavingsAccount* class. Use a static data member *annualInterestRate* to store the annual interest rate for each of the savers. Each member of the class contains a private data member *savingsBalance* indicating the amount the saver currently has on deposit. Provide member function *calculateMonthlyInterest* that calculates the monthly interest by multiplying the balance by *annualInterestRate* divided by 12; this interest should be added to *savingsBalance*. Provide a **static** member function *modifyInterestRate* that sets the *static annualInterestRate* to a new value. Write a driver program to test class *SavingsAccount*. Instantiate two different objects of class *SavingsAccount*, saver1 and saver2, with balances of Rs.2000.00 and Rs.3000.00, respectively. Set the *annualInterestRate* to 3 percent. Then calculate the monthly interest and print the new balances for each of the savers. Then set the *annualInterestRate* to 4 percent, calculate the next month's interest and print the new balances for each of the savers.

2. Design a Ship class that has the following members:
    - A member variable for the name of the ship (a string)
    - A member variable for the year that the ship was built (a string)
    - A constructor and appropriate accessors/getters and mutators/setters
    - A print function that displays the ship's name and the year it was built.

    Design a CruiseShip class that is derived from the Ship class. The CruiseShip class should have the following members:

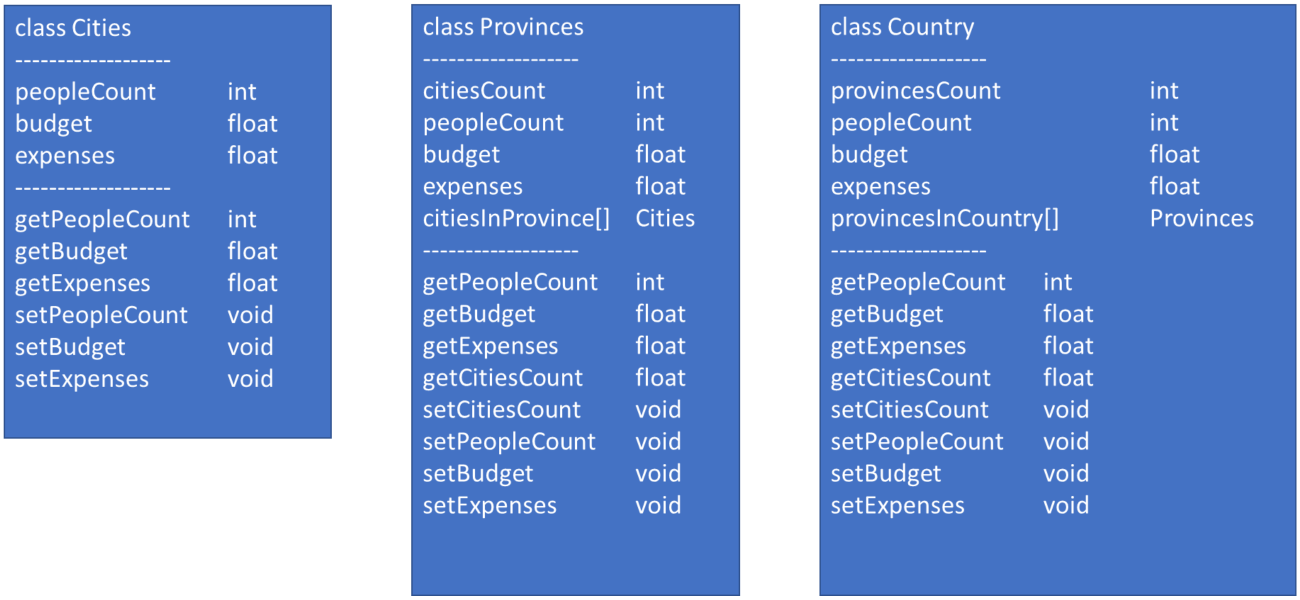
    - A member variable for the maximum number of passengers (an int)
    - A constructor and appropriate accessors and mutators
    - A print function that overrides the print function in the base class. The Cruiseship class's print function should display only the ship's name and the maximum number of passengers.

    Design a CargoShip class that is derived from the Ship class. The CargoShip class should have the following members:

    - A member variable for the cargo capacity in tonnage (an int)
    - A constructor and appropriate accessors and mutators
    - A print function that overrides the print function in the base class. The CargoShip class's print function should display only the ship's name and the ship's cargo capacity.

    Write a test program that declares an array of Ship pointers. The array elements should be initialized with the addresses of Ship, Cruiseship, and CargoShip objects. The program should then step through the array, calling each object's print function.

3. A scenario is shown in given UML diagram given below.



| class Cities | |
|---|---|
| ------------------- | |
| peopleCount | int |
| budget | float |
| expenses | float |
| ------------------- | |
| getPeopleCount | int |
| getBudget | float |
| getExpenses | float |
| setPeopleCount | void |
| setBudget | void |
| setExpenses | void |

| class Provinces | |
|---|---|
| ------------------- | |
| citiesCount | int |
| peopleCount | int |
| budget | float |
| expenses | float |
| citiesInProvince[] | Cities |
| ------------------- | |
| getPeopleCount | int |
| getBudget | float |
| getExpenses | float |
| getCitiesCount | float |
| setCitiesCount | void |
| setPeopleCount | void |
| setBudget | void |
| setExpenses | void |

| class Country | |
|---|---|
| ------------------- | |
| provincesCount | int |
| peopleCount | int |
| budget | float |
| expenses | float |
| provincesInCountry[] | Provinces |
| ------------------- | |
| getPeopleCount | int |
| getBudget | float |
| getExpenses | float |
| getCitiesCount | float |
| setCitiesCount | void |
| setPeopleCount | void |
| setBudget | void |
| setExpenses | void |

Review it to write program for each of the following tasks:

a. Create classes with defined member variables and functions. Each class must have default and parameterized constructors that assigns initial values to all members and destructors.

b. Write a main function where:

   i. Create one country "Pak" which should have five provinces.

   ii. One province has four cities.

   iii. Consider your ID as the total budget in millions for country "Pak", for example ID is "CT-22034", so 22034 million budget.

   iv. Divide that budget and assign equally to each province

   v. From assigned budget, equally divide budget to the cities.

   vi. Print all details in a tabular form that shows all provinces and their members values. All cities and their member values.

c. Write a function *totalExpenses()* which asks people count for each city. If each person has a service cost of last two digits of your id (CT-22034 → 34) then calculate total expenses for each city.

d. Write a function *totalExpenses2()* which uses expenses values of cities and calculates total expenses of provinces and the country.

e. Write a function *highestExpensesP()* which prints details of the province with highest expenses.

f. Write a function *finalFunction()* which prints profit or loss using budget and expenses members for each province in a tabular form.

4. Junaid own an electronics shop and he wants to have an inventory system for Laptops and Mobile phones. Develop an inventory system for him, and also draw its UML class diagram, by using the following guidelines.
    a. Write appropriate classes for Laptop and Mobile so that different instances may be created. There should be multiple constructors for initializing data members and destructors. Each class should have all members variable private. Setters and getters should be defined for each member variable.
    b. The main program should have following logics:
        i. Ask Junaid to enter the number of Laptops in his inventory. Create an array of that many instances (objects) of Laptop class.
        ii. Ask Junaid to enter the number of Mobiles in his inventory. Create an array of that many instances of Mobile Phone class.
        iii. Ask Junaid to enter purchase price of laptops and mobile phones by showing model number and name of the product.
        iv. Ask Junaid to enter profit margin in percentage. Change the selling price using purchase price and profit margin by using formula: SellingPrice = Cost/ (1-Margin%). Set updated value for all products.
        v. At the end of the program, the output should be Name and Model Number of each product, and total profit if all products are sold.