CS1004 - Object-oriented Programming (OOP)
Assignment # 1

Max Points: **36**

Due Date: **Monday, September 27, 2021, 03:30 p.m.**

Carefully read the following instructions!

- It should be clear that your assignment would **not get any credit** if the assignment is submitted after the **due date**. **No** assignment will be **accepted after the due date**.
- Strict action will be taken if the submitted solution is copied from any other student.
- If you people find any mistake or confusion in the assignment (Question statement), please consult before the deadline. After the deadline no queries will be entertained in this regard.
- For any query, feel free to email at: **farah.sadia@nu.edu.pk**
- Submission: Submission will only be accepted through **GOOGLE CLASSROOM**.
- Submit all your codes with your Student ID and task number. **"K201234_Q1"**.
- Every code should be with proper **commenting**.

Question # 01:

Describe following reasons:

1. **Which Feature of OOP illustrated the code reusability? Explain with examples.**
2. **What is the difference between encapsulation and Abstraction? Explain with a real time example.**

Question # 02: *Invoice Class*

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members—a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int).

[*Note:* In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75)—called floating-point values—to represent dollar amounts.] Your class should have a constructor that

initializes the four data members. Provide a *set* and a *get* function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.3.13 (Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members—a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). [Note: In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75)—called floating-point values—to represent dollar amounts.] Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member.

In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

Question # 03:

"Universe7 is the leading exporter of Senzu Beans and has set up offices in different regions of the country. Each office is managed by a head and has a daily expenditure (electricity consumption etc). The day to day operations in the regional office include printing the profit and expenses for the day."

Tasks:

1. **Identify all the entities.**
2. **Shortlist the entities according to their relevance.**
3. **Write proper C++ code to show all the classes, variables, constructors and functions present in the given scenario. You don't need to provide any implementation for the functions**

Question # 04:*Computerization of Health Record*

A healthcare issue that has been in the news lately is the computerization of health records. This possibility is being approached cautiously because of sensitive privacy and security concerns, among others. Computerizing health records could make it easier for patients to share their health profiles and histories among their various health care professionals. This could improve the quality of health care, help avoid drug conflicts and erroneous drug prescriptions, reduce costs and in emergencies, could save lives. In this exercise, you'll design a "starter" HealthProfile class for a person. The class attributes should include the person's first name, last name, gender, date of birth (consisting of separate attributes for the month, day and year of birth), height (in inches) and weight (in pounds). Your class should have a constructor that receives this data. For each attribute, provide set and get functions. The class also should include functions that calculate and return the user's age in years, maximum heart rate and target-heart-rate range (see Exercise 3.16), and body mass index (BMI; see Exercise 2.30). Write an application that prompts for the person's information, instantiates an object of class HealthProfile for that person and prints the information from that object—including the person's first name, last name, gender, date of birth, height and

weight—then calculates and prints the person’s age in years, BMI, maximum heart rate and target-heart-rate range. It should also display the “BMI values” chart from Exercise 2.30. Use the same technique as Exercise 3.16 to calculate the person’s age.