National University of Computer and Emerging Sciences

OOP (CS1004)

Date: March 19th 2024

Time: 8:00 – 09:30 am

Course Instructor(s)

Shafique Rehamn, Ms Mahnoor, Ms Zainab,

, Ms Javeria

Lab Mid Exam

Total Time: 1.5 Hours

Total Marks: 50

Total Questions: 03

Semester: SP-2024

Campus: Karachi

Dept: Computer Science

Muhammad Abdur Rahman 23K0760 BCS-2A

Student Name Roll No Section Student Signature

General Instructions:

- Carefully read the following instructions before attempting the paper.
- Except your Roll No and Section, DO NOT WRITE anything on this paper.
- The Exam consists of 3 questions on 3 printed sides of 2 page.
- In case of any ambiguity, you may make assumptions, but your assumption must not contradict any question.
- **Submission Instructions:**
  - You must comment your student ID on top of each file.
  - Name the .cpp file for each question according to Roll_No e.g. k23xxxxQ1.c, k23xxxxQ2.c etc.
  - Create a folder with the name K23XXXX according to your student id.
  - Put all your .cpp files (1 for each question) in the above folder. Then paste this folder on the local share.
  - Submissions are on local storage that can be accessed using win+R keys and entering \\172.16.5.43

Paper B

Question Q1 ***Weightage: 6*** **; Marks: 15**

You have to develop a statistical calculation module for a scientific research application. The requirements for this module are outlined below:

a. Implement a C++ class named **'StatisticalAnalyzer'** with a function ***computeAverage*** that calculates the average of an array of numerical data. The function should be designed to be accessible without creating an instance of the class and should be optimized for efficiency. Describe the approach you would take to ensure flexibility for different data sets, considering the dynamic nature of scientific research data.

b. Extend the **StatisticalAnalyzer** class by implementing a member function ***calculateStandardDeviation*** that computes the standard deviation of a given array of numerical data. Optimize this function for efficiency and ensure it is resistant to unintentional modifications of the object's data.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

(Ctrl) ▾

Where:

n is the number of data points

xi represents each individual data point

$\bar{x}$ is the mean of the data set, calculated as $\frac{\sum_{i=1}^{n} xi}{n}$

Question Q2 *Weightage: 6* ; **Marks: 15**

Design a movie streaming application with the following classes: **Movie, Playlist,** and **User.**

a) Implement the **Movie** class with **Member Variables: title**: string - to store the title of the movie, **director**: string - to store the director of the movie and **duration**: int - to store the duration of the movie in minutes and **Member Functions: Movie(const string& _title, const string& _director, int _duration)**: Constructor to initialize the movie object with given title, director, and duration, **getTitle() const, getDirector() const** and **getDuration() const.**

b) Implement the Playlist class with **Member Variables: name**: string - to store the name of the playlist, **movies**: Movie* array - to store pointers to the movies in the playlist and **numMovies**: int - to store the number of movies in the playlist and **Member Functions: Playlist (const string& _name):** Constructor to initialize the playlist object with a given name. **addMovie (Movie* movie)**: Function to add a movie to the playlist, **removeMovie (Movie* movie)** and **displayInfo () const**: Function to display information about the playlist and its movies.

c) Implement the User class with **Member Variables: name**: string - to store the name of the user, **playlists**: Playlist* array - to store pointers to the playlists created by the user and **numPlaylists**: int - to store the number of playlists created by the user and **Member Functions: User (const string& _name):** Constructor to initialize the user object with a given name, **createPlaylist (const string& playlistName), addMovieToPlaylist (Playlist* playlist, Movie* movie)** Function to add a movie to a specific playlist., **removeMovieFromPlaylist (Playlist* playlist, Movie* movie) And displayUserPlaylists ()** const.

d) Figure out the relationship between the Playlist, User and Movie classes.

Now, consider the following scenario: You are working on the movie streaming application, and a User named Mujeeb Rehman has created two playlists: "Action Movies" and "Comedy Movies." The "Action Movies" playlist has the movies "The Dark Knight" directed by Christopher Nolan and "Inception" directed by Christopher Nolan. The "Comedy Movies" playlist has the movies "The Hangover" directed by Todd Phillips and "Superbad" directed by Greg Mottola. Implement the necessary steps to associate the movies with the respective playlists created by Mujeeb Rehman and display relevant information about the playlists and their movies. Additionally, Mujeeb Rehman decides to remove the movie "Inception" from the "Action Movies" playlist. Update the playlist accordingly and display the updated information.

Question Q3 ***Weightage: 8*** **; Marks: 20**

As a software engineer, you're tasked with improving a banking system to handle different types of accounts and insurance calculations. Here's the situation: the system must now manage Basic, Savings, and Checking accounts, and insurance premiums should be determined by a formula provided by the insurance department. However, insurance is only available to Savings account holders. Additionally, there's an option for customers to donate money if their account meets certain criteria: it must be a Savings account, have a balance of at least $500, and the last withdrawal amount must be between $0 and $50. The system must prioritize security using OOP concepts.

Your task is to:

a) Outlining the class structure for the banking system, specifying relationships between different account types and extending hierarchy with the special features of Savings accounts.

b) Define the necessary data members (balance, creation date, last withdrawal amount) and member functions for each account type to fulfill its purpose effectively.

c) Integrate insurance premium calculation functionality criteria is provided above in the scenario, into the system, ensuring its seamlessly incorporated into the account management process.

d) Utilizing initializer lists and getter/setter functions for encapsulation. Additionally, given the scenario, which constructor and destructor will be invoked first: the Account class or the Insurance class? Provide constructor and destructor for each class.

Please proceed with the implementation, prioritizing security alongside functionality by incorporating OOP concepts.

Wish You Best of Luck!