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 Final Exam consists of 4 questions on 2 printed sides of 1 page.
- In case of any ambiguity, you may make assumptions, but your assumption must not contradict any statement in the question paper.
- **DON'T** share your program, if your code is matched to any member of your class, both will get straight F in the course without asking who shared or who magically copied.

Submission Instructions:

- You must comment your student ID on top of each file. (Line#1 of your code).
- Name the .c file for each question according to Roll_No e.g. **k22-xxxx_Q1.c**, k22-xxxx_Q2.c etc.
- Create a ZIP folder of all your solutions and copy it in the local storage with the title K22-xxxx_A.
- Submission are on local storage that can be accessed using win+r keys and entering \\172.16.5.41 address in the dialog box.
- Enter your username as **khifast\K22xxxx** and its assigned password.
- Zip folder needs to be pasted in the "**Exam Submission\teacherName**" folder

Time: 120 Minutes **Max Points:** 100 Points

Task 1: [30 mins, 25 points]

Create a base class called Employee that contains common properties and methods for all employees. The class has attributes such as name, employee_id, salary, and designation. You need to define methods such as print_details() which will be override in the derived class.

Next, you create several specific employee classes that inherit from the Employee class. For example, you create a Manager class that has additional attributes such as team_size and bonus, and a Developer class that has attributes such as programming_languages and experience_years. Both classes have get_performance() function which generates performance score based on their evaluation criteria.

For the get_performance() method in the employee class:Attendance: 20%,Punctuality: 15%,Quality of Work: 30% Timeliness of Work: 25%,Communication Skills: 10%.

Once the overall percentage score is calculated, it could be mapped to a letter grade using a grading scale, such as: 90-100: A 80-89: B 70-79: C 60-69: D Below 60: F

- Create a generic function called display_performance() that takes any Employee object either Manager or Developer as an argument and calls the get_performance() function.
- Define a generic filter_employees() function to filter employees by field value. It takes an array of employees, a second parameter indicating the field to filter by (e.g., "designation" or salary). The function should call print detail functions for only those employees that match the specified value.

Task 2: [30 mins, 25 points]

For Ramadan, Imtiaz super store has made different packages for its customers. Every Ramadan package falls under the "Ramadan Package" category. There are three types of packages provided under this category: "Budget Package", "Super Package", "Mega Package".

For this question, you are required to do the following:

1. Ramadan Package: This is an abstract class that contains the name (string) of the package. By default, the name will be set as "not defined", however the name can be initialized by using its parameterized constructor. It has one function called "calculateDiscount()" which will be defined by the child classes of this class.

2. Budget Package: This class will be inheriting from the Ramadan Package class. Create a default constructor for your class which sets the name of its parent to "Budget" by using its parameterized constructor. The class has a member variables called cost (integer) and itemList (string array). You class should override the calculateDiscount() method such that there is a 5% discount applied on the cost of the package.

3. Super Bachat Package: This class will be inheriting from the Ramadan Package class. Create a default constructor for your class which sets the name of its parent to "Super Bachat" by using its parameterized constructor. The class

has a member variables called cost (integer) and itemList (string array). You class should override the calculateDiscount() method such that there is a 10% discount applied on the cost of the package.

4. Mega Bachat Package: This class will be inheriting from the Ramadan Package class. Create a default constructor for your class which sets the name of its parent to "Mega Bachat" by using its parameterized constructor. The class has a member variables called cost (integer) and itemList (string array). You class should override the calculateDiscount() method such that there is a 15% discount applied on the cost of the package.

In your main function, create three pointers of "Ramadan Package" class. Dynamically create one object of each "Budge Package", "Super Bachat Package" and "Mega Bachat Pacakge"class.

Task 3: **[30 mins, 25 points]**

Develop an online rental system for a software firm that allows users to book or rent licensing software for a specific period of time. Users can select up to five software offerings, with an additional backup software in case one of their chosen software becomes unavailable. Each software offering can have a maximum of forty copies of license, and any software offering with less than ten rent bookings will be canceled.

Create a "Software" class with the following attributes: Software_ID (string), software_name (string), and max_license_copies (integer). This class has the following methods:

1. Constructor method to initialize the attributes
2. Getter and setter methods for each attribute
3. Method to rent a customer for the software
4. Method to drop a user from the software
5. Friend function to calculate the total fee for a customer's rented software
6. Overloading of the << operator to display the software details

Create a "User" class with the following attributes: User_id (string), name (string), rented_software (Software list), and backup_software (Software e). This class has the following methods:

1. Constructor method to initialize the attributes
2. Getter and setter methods for each attribute
3. Method to rent a software (checks if the software is already full and adds the customer to the software)
4. Method to drop a software (removes the customer from the software)
5. Method to add a backup software (checks if the backup song is already fully booked)
6. Friend function to calculate the total fee for the customer's rented software (calls the friend function in the Software class to calculate the rental fees)
7. Overloading of the << operator to display the user details

Finally, create a main function that simulates the rental process. Create several User and Software objects and use the methods in the User and Software classes to simulate the rental process. Use the friend function and operator overloading to calculate rental fees and display information.

Task 4: **[30 mins, 25 points]**

You are creating a program for a small business that needs to keep track of its inventory of products. The business owner has provided you with a list of products, including their name, price, and current quantity in stock. The owner wants to be able to add new products, update existing products, and generate a report of the current inventory.

Design a scenario for your program that involves the use of file handling in C++. The scenario should include the following:

1. A menu system that allows the user to choose from options such as adding a new product, updating an existing product, generating an inventory report, and exiting the program.
2. The program should read in the existing inventory from a text file and store the data in a data structure such as an array.
3. When the user adds a new product, the program should append the data to the text file.
4. When the user updates an existing product, the program should modify the data in the text file.
5. When the user generates an inventory report, the program should read in the data from the text file and output it to the console or to another file in a formatted manner. The report should include the name, price, and current quantity in stock for each product.

Use file handling in C++ to create a program that efficiently manages inventory for this small business.