

## OOP Lab Task 08

Design a program in C++ that simulates a Journal (diary). The program should allow the user to enter data from the console, which will be saved in a text file representing the journal. If the program is opened again, the user should be able to see the previously saved journal entries.

The program should be menu-driven, with the following options:


**Add an entry to the journal:** When the user selects this option, they should be prompted to enter the text for the journal entry. The program should append this new entry to the end of the journal file, without deleting any previously saved entries.

**Display the journal:** When the user selects this option, the program should read the entire contents of the journal file and display them on the console.

**Hints:** Think about the logic how will you append the data instead of overwriting. You can use the append mode of fstream, manually setting the pointer to the end of file, etc. Search and understand the ways yourself.

When reading the data, take care about reading from the file. Based on the approach you used, your pointer may be at the end of the file. So try opening and closing the file again where needed (so that your pointer/marker get back to the beginning of the file, when reading the data), or manually setting the pointer to the beginning of the file.

**Output when you open the program for the first time.**

 C:\Users\hy\Desktop\Untitled1.exe

```
Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
2
Hello. First entry.

Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
1
Enter journal entry:
Hello. Second Entry.
Entry added to journal.

Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
2
Hello. First entry.
Hello. Second Entry.

Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
```

**Output when you open the program again, after writing the journal.**

```
Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
2
Hello. First entry.
Hello. Second Entry.

Please choose an option:
1. Add an entry to the journal
2. Display the journal
3. Exit
```

-----

Imagine you are working on a project for a company that wants to manage their Employees data. Your task is to create a program that can handle different types of employees such as hourly employees and salaried employees. Each employee has a name, and a salary.

To achieve this task, you need to create a class hierarchy that uses inheritance to represent the different types of employees. You should start by creating a base class called "Employee" that contains common attributes("Employee name") and methods(constructor, getters, setters, and calcSalary()) for all types of employees.

Next, create derived classes for each type of employee (HourlyEmployee and SalariedEmployee), which inherit from the base class. These derived classes should include additional attributes and methods that are specific to each type of employee.

For example, the HourlyEmployee class might have attributes for its hourly rate and hours worked, while the SalariedEmployee class might have an attribute for its annual salary. Each derived class should also have a constructor that initializes its specific attributes and calls the base class constructor.

Finally, you should override the method "calcSalary()" for each derived class. The "calcSalary()" method should calculate the salary for each type of employee based on its specific attributes.

For Hourly Employee salary calculation:

Salary= no of hours worked \* hourly rate

For Salaried Employee salary calculation:

Salary= annual salary/12

To test your program, you should create one instance of each type of employee and call the "calcSalary()" method to ensure that your program is working correctly.

**Note:** Input the employee data from user in main and pass them to the constructor. The salary will be calculated based on the info user has entered.

Enter hourly employee name: Fariba

Enter hourly rate: \$10

Enter hours worked: 5

Hourly Employee Details:

Name: Fariba

Hourly Rate: \$10

Hours Worked: 5

Hourly Employee Salary:50

Enter salaried employee name: Ali

Enter annual salary: \$100000

Salaried Employee Details:

Name: Ali

Annual Salary: \$10000

Slaried Employee Salary: \$ 833