Task 01:
Mini Employee Management System *[100 marks/ 90 minutes]*

Create a class **Date.** Class properties will be: **(5 marks)**

- Day
- Month
- Time

Methods:

- Create Default, Parameterized and copy constructor and Destructor for it.
- Create a Display function that will display in format Hours: Minute: second.

Overload the following operators:

- << : for Output in the format: Day- Month- Year.
- >> **:** for input in format: Day Month Year.
- +: Add two Date class objects. Handle Logic day sum gets bigger than a month, handle logic where month sum is greater than equal to a year.
- **- :** Add two Date class objects. Handle corner cases as well.

Create a class **Dept:** **(5 marks)**
Properties:

- Deptid
- dname
- Location

Create a Default, Parameterized constructor and display function. Make sure that each employee has unique id.

- << : for Output.
- >> **:** for input.

Create a class **Employee:** **(5 marks)**
Properties:

- Empid,
- ename,
- hiredate (Date object)
- salaray
- Deptid (dept object)
- Working Hours

Methods:

- Default, Parameterized and copy constructor, Destructor.

- Display function to display all data.
- Function to get number of years the employee has been working.

Operators:

- << : for Output.
- >> **:** for input.

➢ Create a class **Firm:** **(5 marks)**

Properties:

Firm name
Number of employees
Employees vector

Methods:

Create Default, parameterized and copy constructor, Destructor. **(20 marks)**

- Create Display function to Display whole Firm data and its employees.
- Create a function to add an employee to firm. Make sure that each employee has unique id.
- Create a function to remove an employee from firm.
- Create a function to update data of an employee of firm.
- Sort all the Employee based on their working Hours in descending order using bubble sort.
- Function to get the employee with the highest salary.

➢ Create a class **Attendance:** **(30 marks)**

This class should be able to access private data member "employee vector" of Firm class.
Additionally it will have:

- ◆ **Vector of Date object**
- ◆ **Present_employees (vector of vectors):** move the employee that are present in this day and make a copy of them in present_employees vector.

This class will store present employees against each day. For example Date objects vector will have 1 day at its first index and present_employees vector will have a vector at its first index consisting of employees that were present that day.
First ask the user the day for which attendance is to be marked. Then display all employees to the user and take input weather that employees is present or not that day.

Main Function: (20 marks)

Main function should be Menu based, presenting each function (of Firm class and Attendance class) to be selected by user. Handle all the validation checks and corner cases as well.