I’m working on the Open-ended Project Progress.  
The project I’m developing has the purpose to solve one of the problems we have in the company I work for.  
Problem: managing company’s employee information, including their name, job title, salary, and vacation days; calculate employee bonuses based on their performance.  
  
I want to create a program that will create a list of employees, add new employees, remove employees, and calculate bonuses. The program will have the following 8 classes:

1. Program.cs: This is the main entry point for the program. It initializes the program and handles any necessary setup before starting the program loop. It also contains the Main method, which is the starting point of the program.
2. Employee.cs: This is the base class for all employee information. It defines the basic properties and methods that all employees have, such as name, job title, salary, and vacation days.
3. Manager.cs: This is a subclass of Employee that adds manager-specific information and behavior, such as managing employee performance and delegating tasks to employees.
4. HRAdmin.cs: This is another subclass of Employee that adds HR administrator-specific information and behavior, such as managing employee benefits and handling employee complaints.
5. EmployeeList.cs: This class manages a list of employees. It allows the program to add new employees, remove employees, and access information about individual employees or the entire list of employees.
6. BonusCalculator.cs: This class calculates employee bonuses based on their performance. It takes into account factors such as productivity, attendance, and overall job performance.
7. InputHandler.cs: This class handles user input, such as reading input from the console and validating user input.
8. OutputHandler.cs: This class displays program output to the console, such as displaying employee information or program messages.

The principle of abstraction will be demonstrated by having separate classes for Employee, Manager, and HRAdmin; each with their own responsibilities and methods for managing employee data.  
The principle of encapsulation will be demonstrated by using private fields for employee information and public methods for accessing and modifying that information.  
The principle of inheritance will be demonstrated by having Manager and HRAdmin classes inherit from the Employee class and sharing common methods and properties.  
The principle of polymorphism will be demonstrated by having the Manager class override the CalculateBonus method from the Employee class to provide a custom implementation.  
  
So far, I have finished the project. Right now I’m working on figuring out which other classes could be added and how to edit the Program.cs class.