

**PRE-LAB**

1. What are the types of Inheritance & How it is helpful to implement an Application?

**Solution:**

1. Single Inheritance: A class inherits from only one base class.
2. Multiple Inheritance: A class inherits from more than one base class.
3. Multilevel Inheritance: A derived class inherits from another derived class.
4. Hierarchical Inheritance: Multiple classes inherit from a single base class.
5. Hybrid Inheritance: Combination of two or more types of inheritance.

Inheritance helps in code reusability, promotes the concept of hierarchical modeling, and allows for the creation of more specialized classes, enhancing the structure and extensibility of an application

**IN-LAB:**

**Task1: To create classes Employee, SalesPerson, Manager and Company with predefined functionality.**

**Low level requires:**

1. To create basic class **Employee** and declare following content:

- Three closed fields – text field **name** (employee last name), money fields – **salary** and **bonus**
- Public property **Name** for reading employee's last name
- Public property **Salary** for reading and recording salary field
- Constructor with parameters string **name** and money **salary** (last name and salary are set)
- Virtual method **SetBonus** that sets bonuses to salary, amount of which is delegated/conveyed as bonus
- Method **ToPay** that returns the value of summarized salary and bonus.

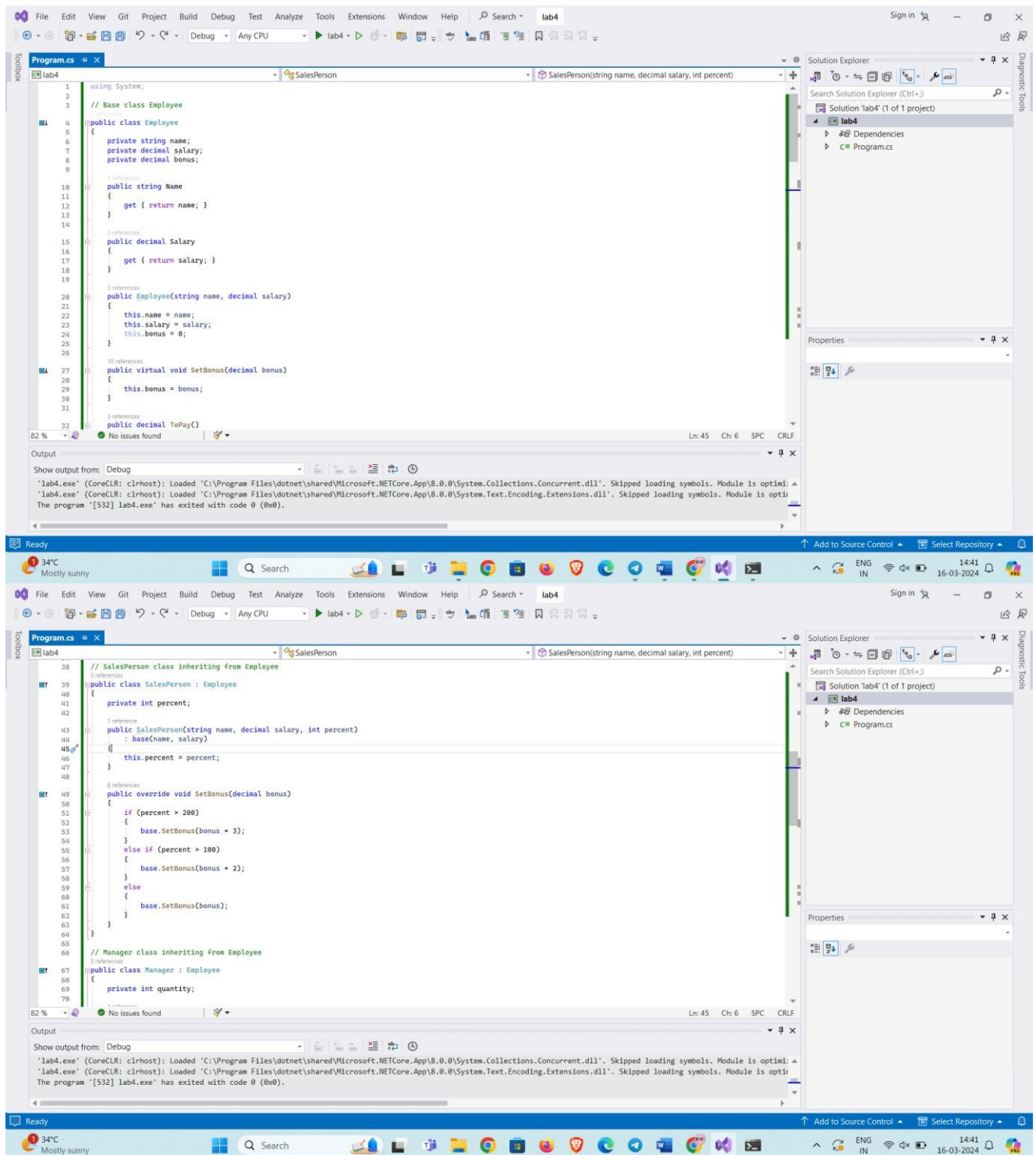
2. To create class **SalesPerson** as class **Employee** inheritor and declare within it:

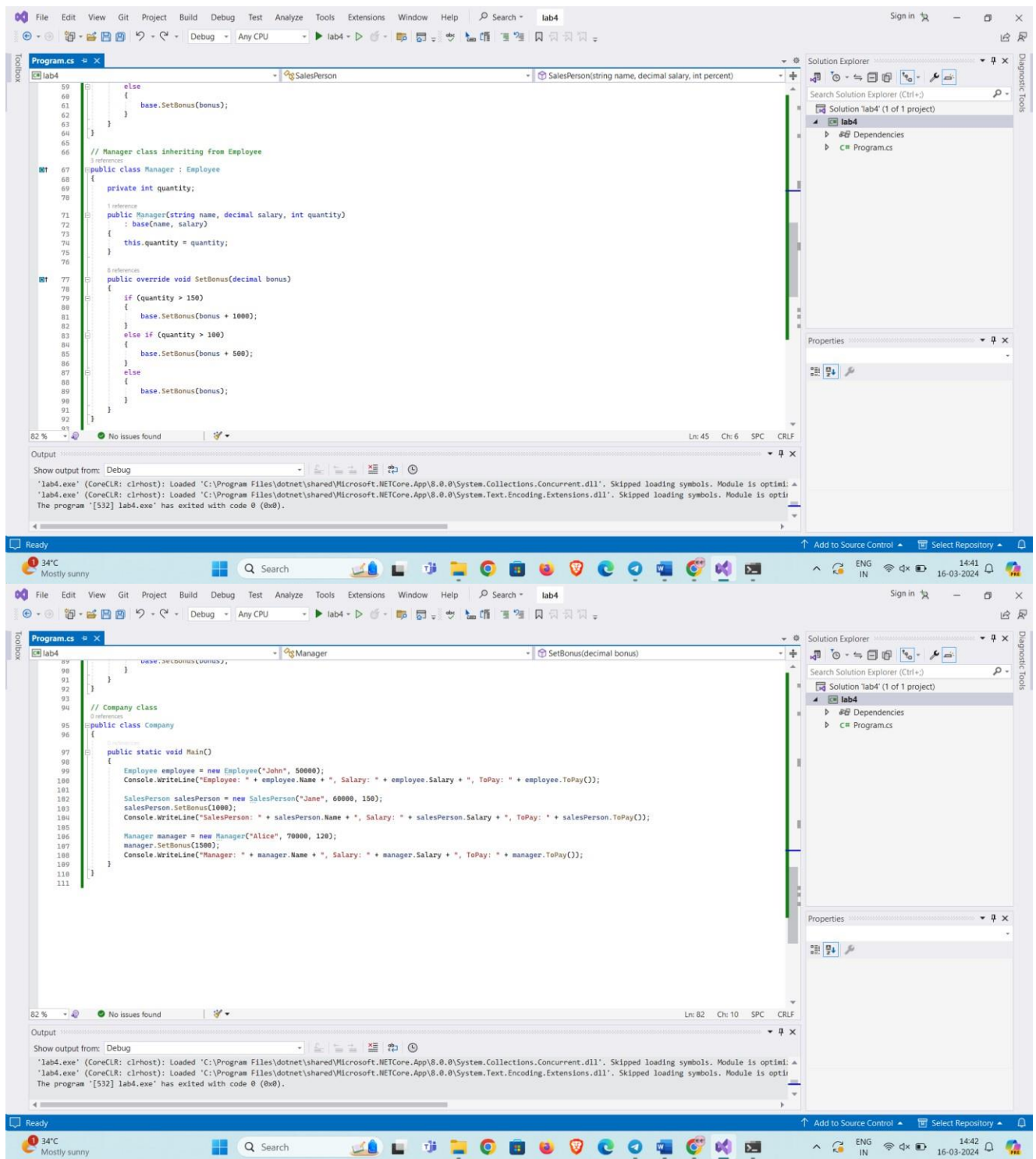
- Closed integer field **percent** (percent of sales targets plan performance/execution)
- Constructor with parameters: **name** – employee last name, **salary**, **percent** – percent of plan performance, first two of which are passed to basic class constructor
- Redefine virtual method of parent class **SetBonus** in the following way: if the sales person completed the plan more than 100%, so his bonus is doubled (is multiplied by 2), and if more than 200% - bonus is tripled (is multiplied by 3)

3. To create class **Manager** as **Employee** class inheritor, and declare with it:

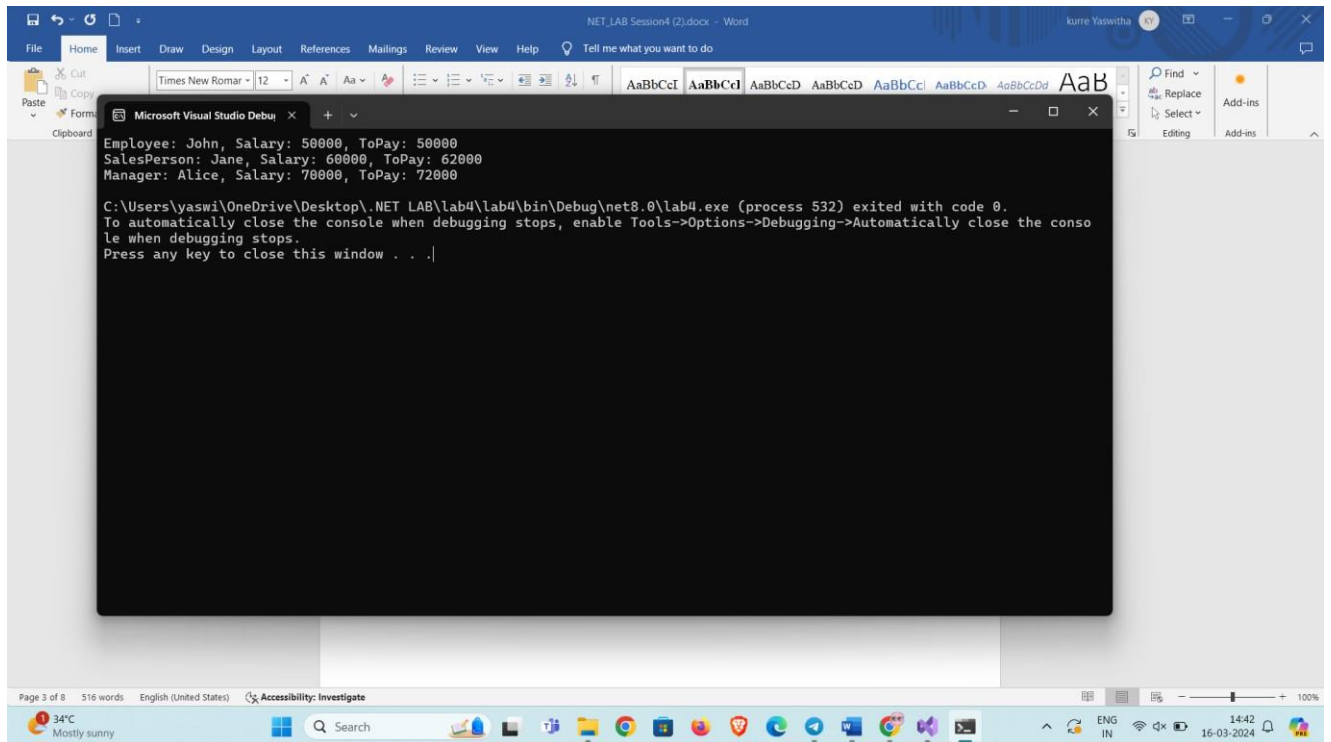
- Closed integer field **quantity** (number of clients, who were served by the manager during a month)
- Constructor with parameters string **name** – employee last name, **salary** and integer **clientAmount** – number of served clients, first two of which are passed to basic class constructor.
- Redefine virtual method of parent class **SetBonus** in the following way: if the manager served over 100 clients, his bonus is increased by 500, and if more than 150 clients – by 1000.

## Solution:





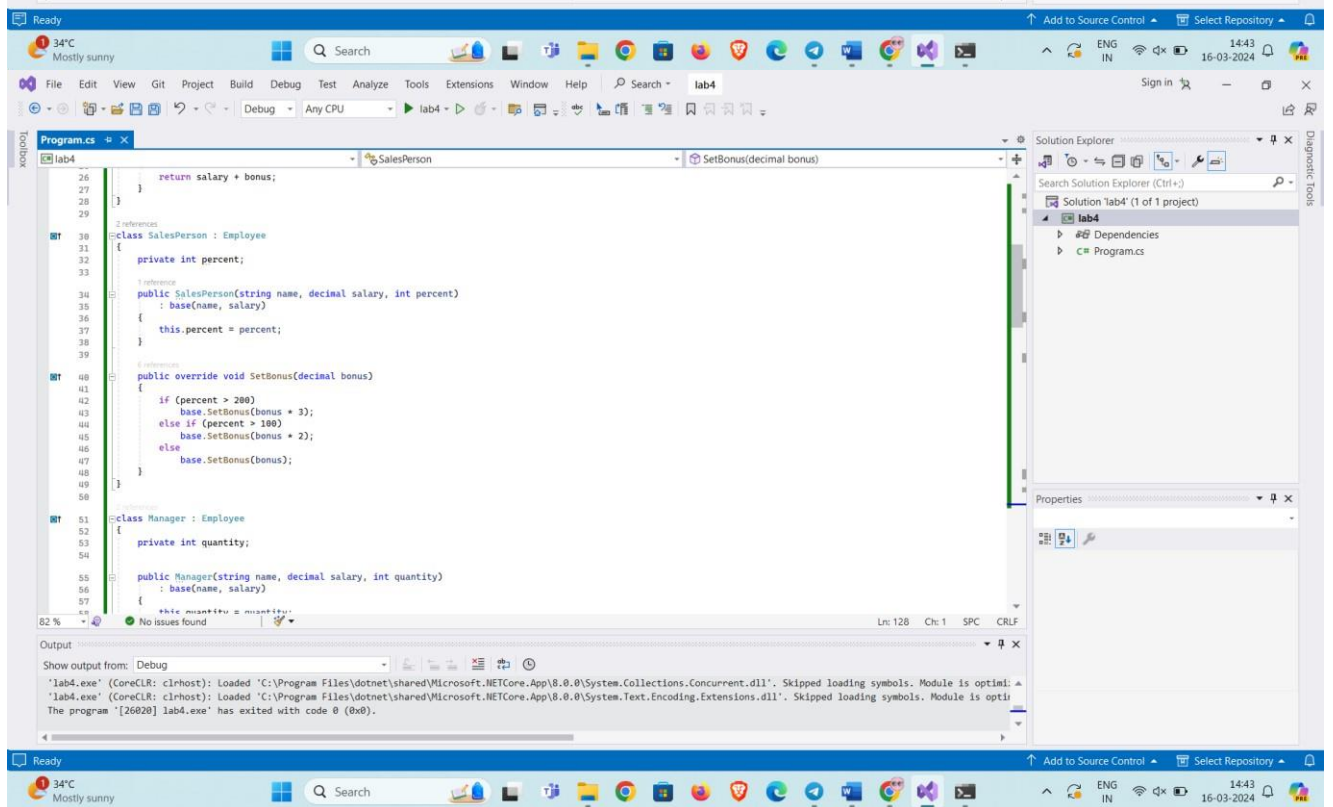
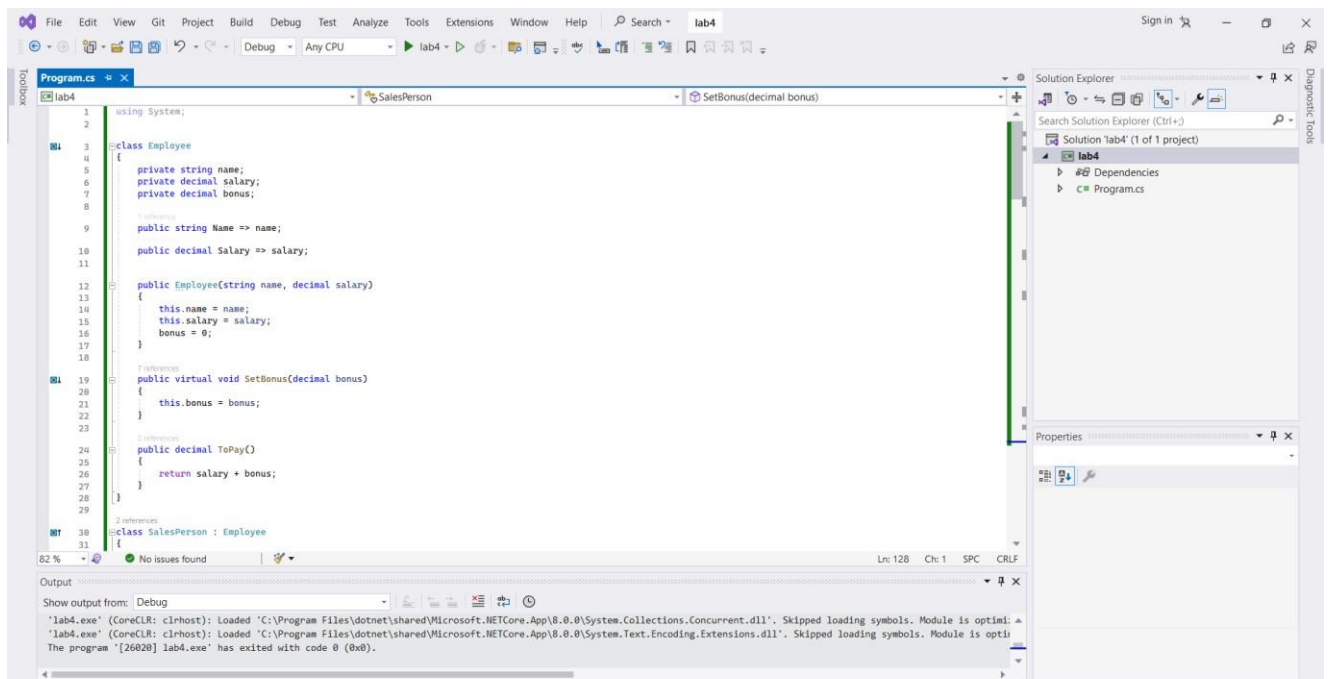
OUTPUT:



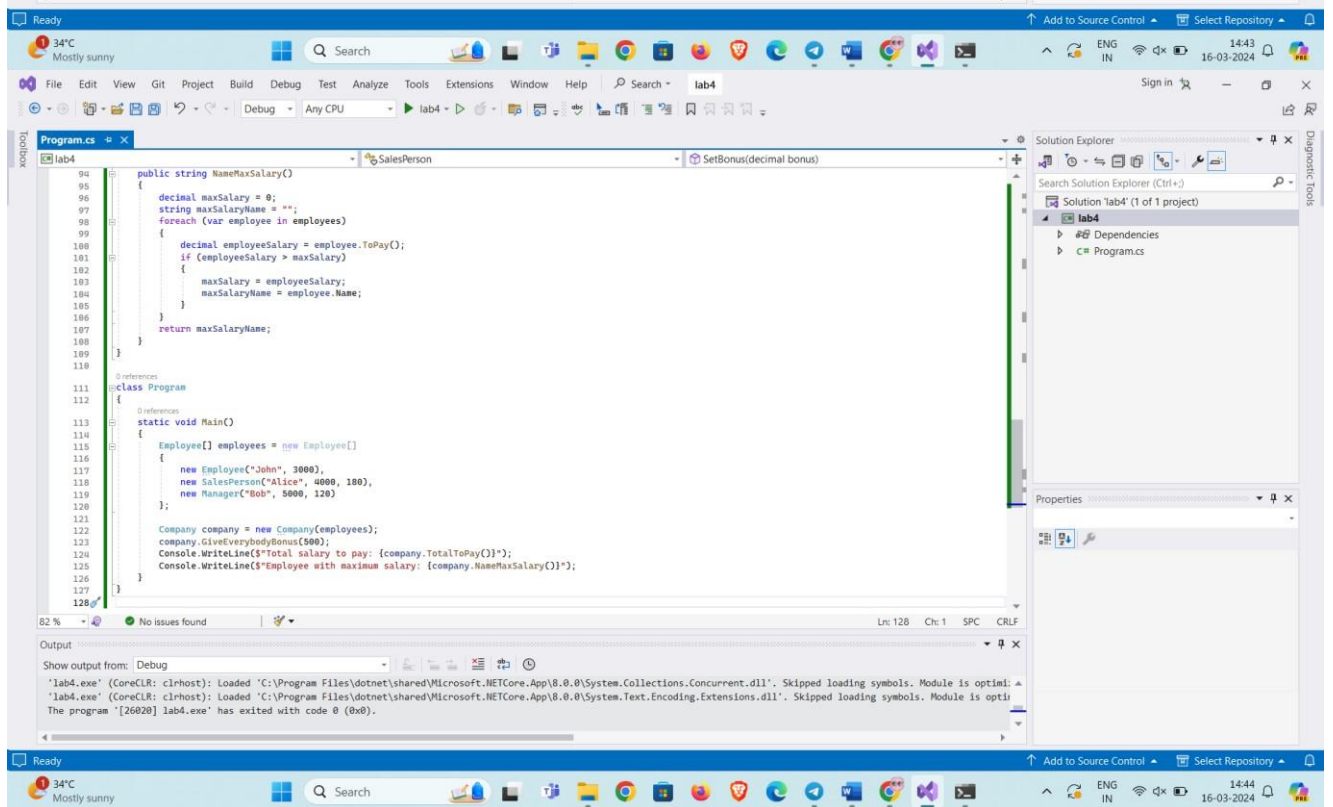
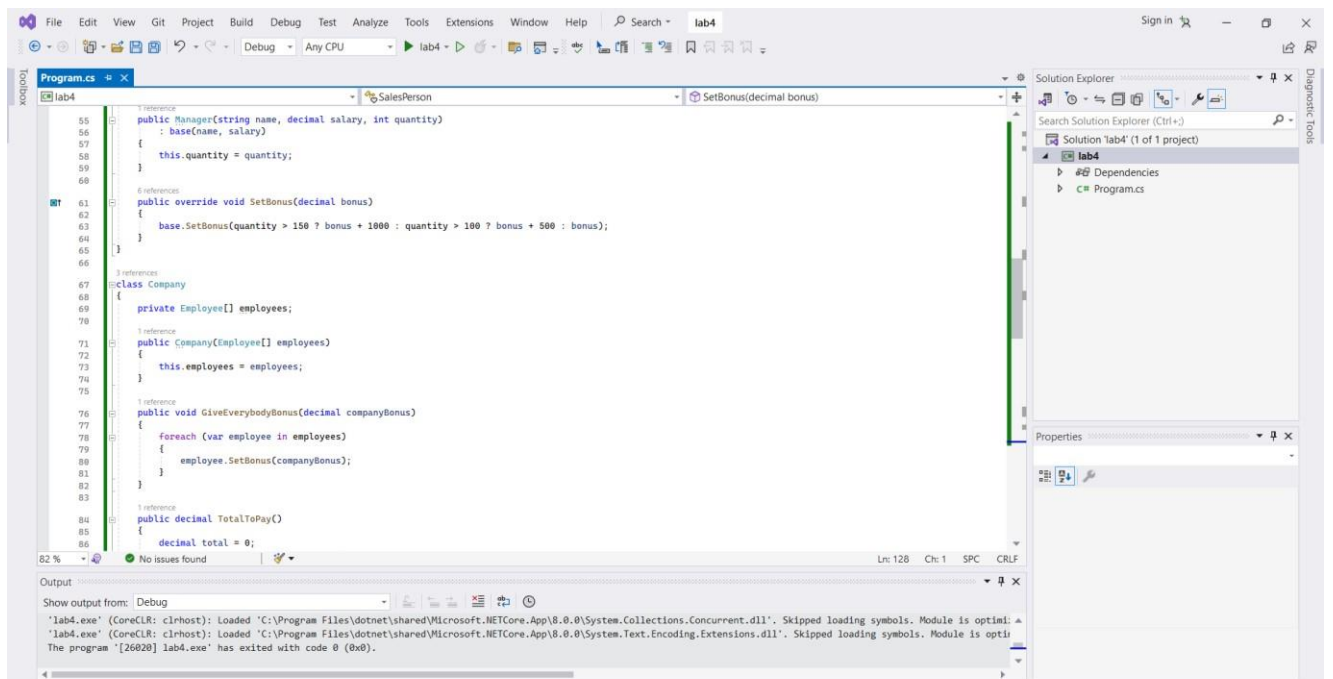
## TASK 2: Advanced level requires:

1. To fully complete Low level tasks.
2. Create class Company and declare within it:
  - Closed field **employees** (staff) – an array of Employee type.
  - Constructor that receives employee array of **Employee** type with arbitrary length
  - Method **GiveEverybodyBonus** with money parameter **companyBonus** that sets the amount of basic bonus for each employee.
  - Method **TotalToPay** that returns total amount of salary of all employees including awarded bonus
  - Method **NameMaxSalary** that returns employee last name, who received maximum salary including bonus.

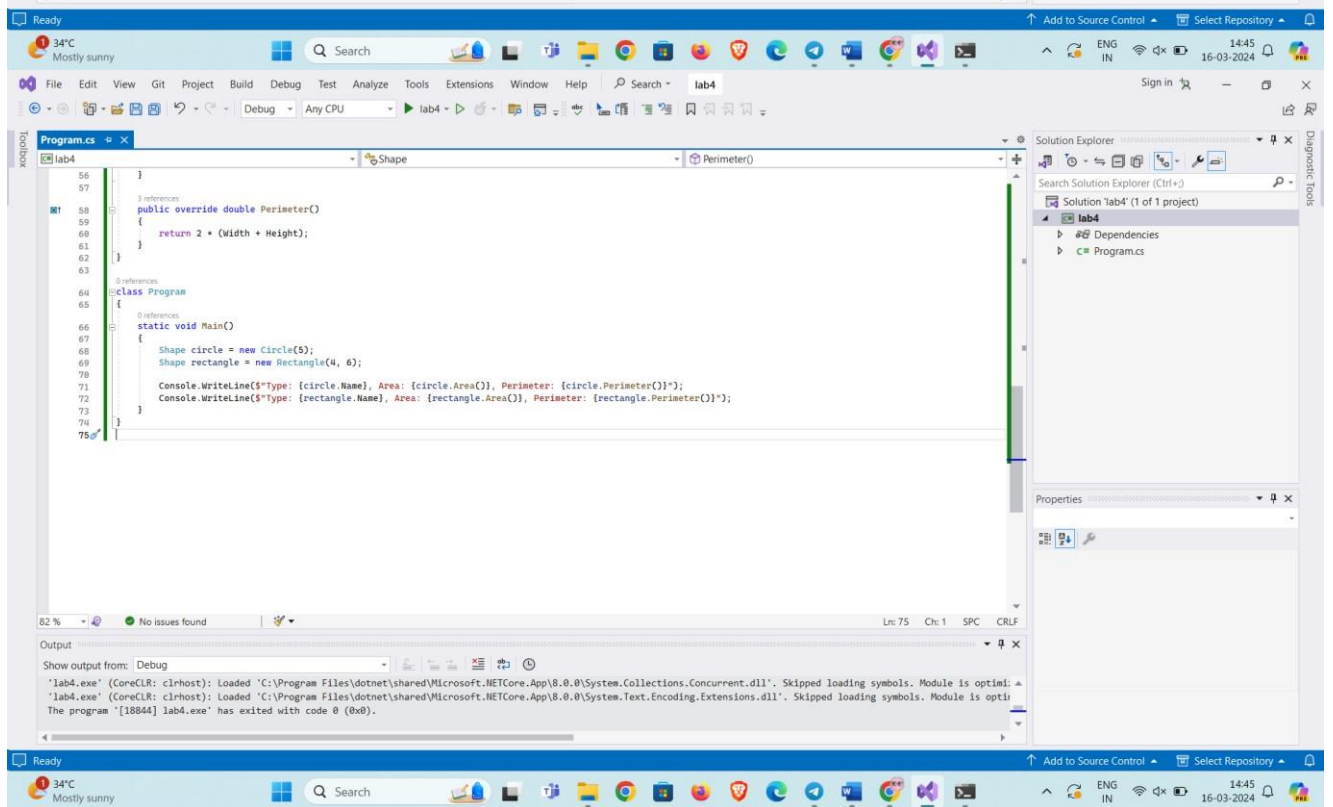
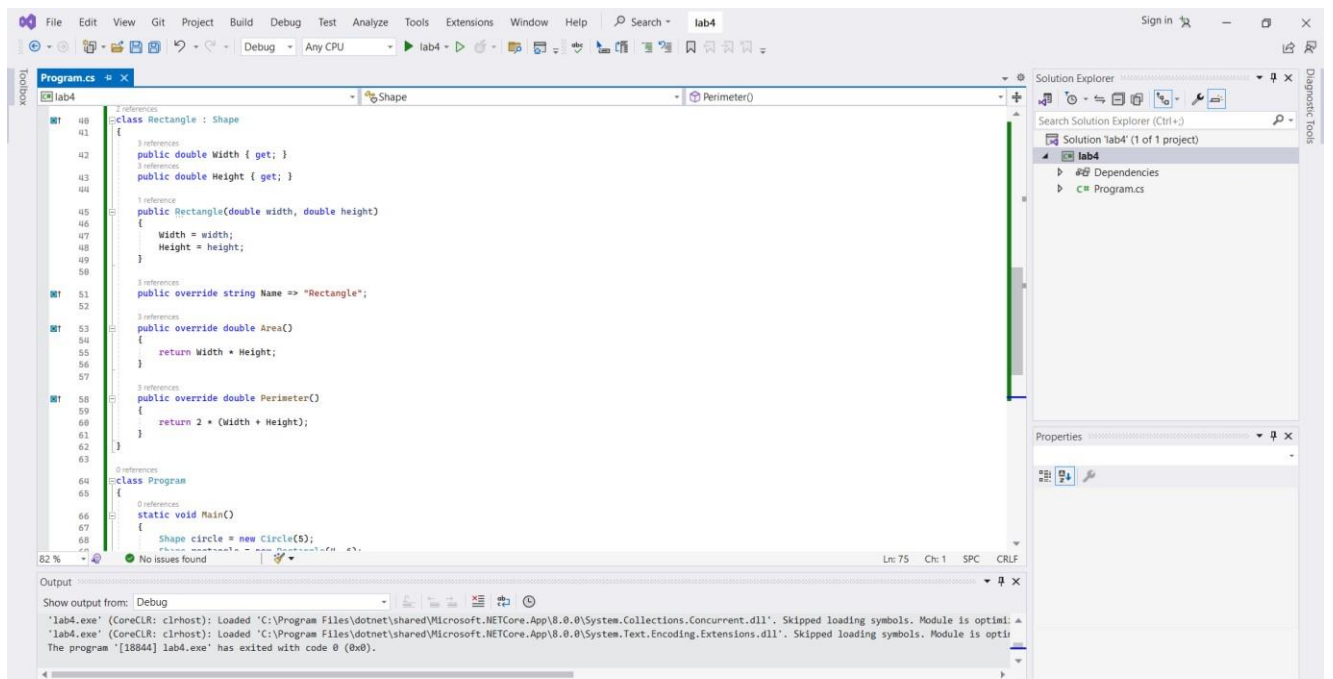
## Solution:













## OUTPUT:

