

2100090095

M.Naresh

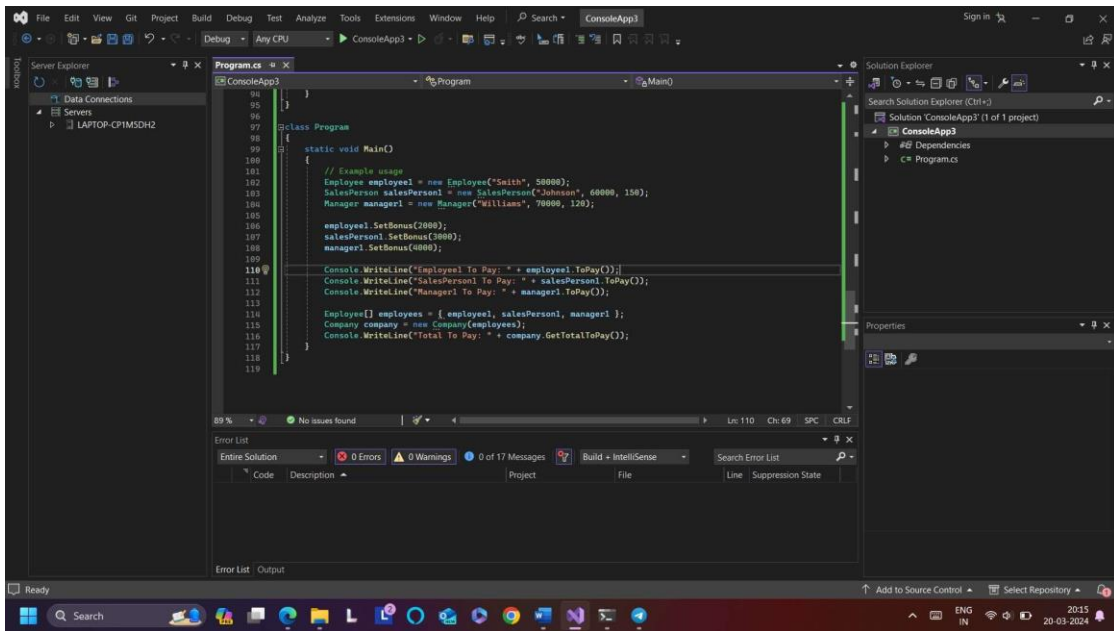
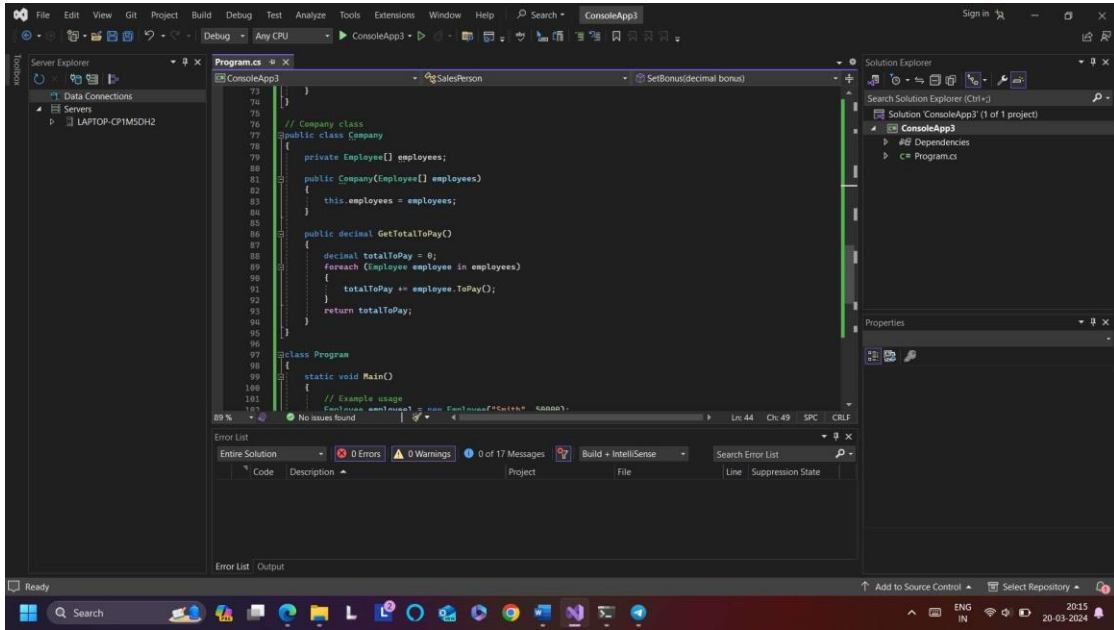
Task1: To create classes Employee, SalesPerson, Manager and Company with predefined functionality.

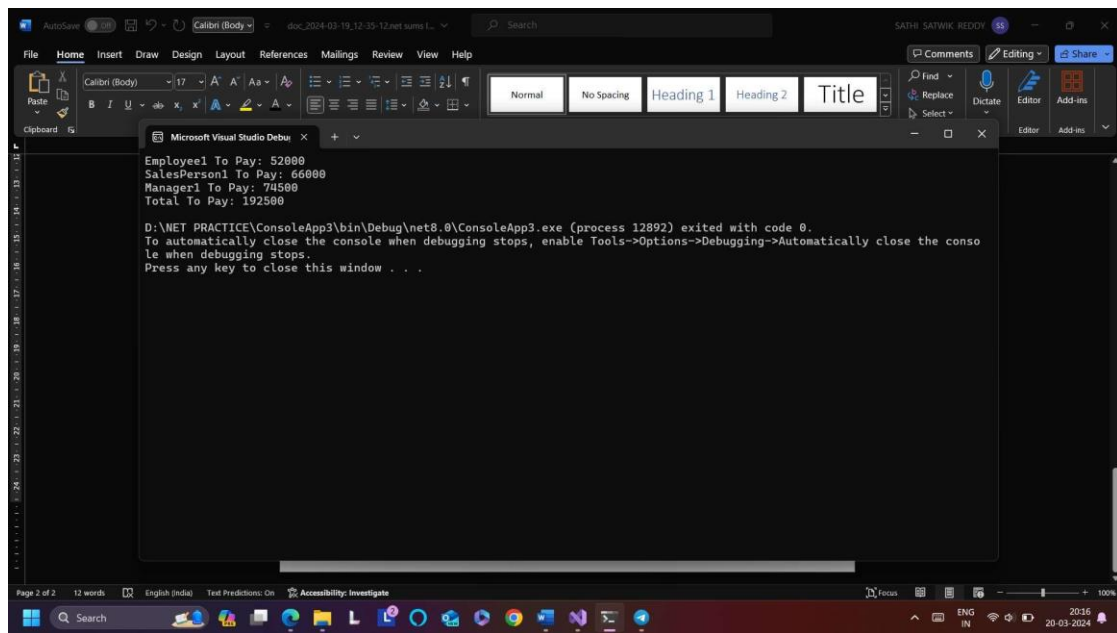
This screenshot shows the Visual Studio IDE with a C# console application. The main editor displays the `Program.cs` file, which defines a base class `Employee`. The class has private fields for `name`, `salary`, and `bonus`. It includes a `Name` property with a getter and setter, and a `SetBonus` method that updates the `bonus` field. A constructor `Employee(string name, decimal salary)` initializes the `name` and `salary` fields. The `ToPay` method is also defined, which returns the sum of `salary` and `bonus`. The `Server Explorer` on the left shows the project structure, and the `Solution Explorer` on the right shows the project files. The `Error List` at the bottom indicates no errors.

```
1 using System;
2
3 // Base class Employee
4 public class Employee
5 {
6     private string name;
7     private decimal salary;
8     private decimal bonus;
9
10    public string Name => name;
11    public decimal Salary
12    {
13        get => salary;
14        set => salary = value;
15    }
16
17    public Employee(string name, decimal salary)
18    {
19        this.name = name;
20        this.salary = salary;
21    }
22
23    public virtual void SetBonus(decimal bonus)
24    {
25        this.bonus = bonus;
26    }
27
28    public decimal ToPay()
29    {
30        return salary + bonus;
31    }
32 }
```

This screenshot shows the Visual Studio IDE with the same C# console application. The main editor displays the `Program.cs` file, which now includes the `SalesPerson` class. The `SalesPerson` class inherits from `Employee` and has a private field for `percent`. It includes a constructor `SalesPerson(string name, decimal salary, int percent)` that calls the base class constructor. The `SetBonus` method is overridden to calculate the bonus based on the `percent` field. The `Server Explorer` on the left shows the project structure, and the `Solution Explorer` on the right shows the project files. The `Error List` at the bottom indicates no errors.

```
35
36
37 public decimal ToPay()
38 {
39     return salary + bonus;
40 }
41
42
43 // SalesPerson class inheriting from Employee
44 public class SalesPerson : Employee
45 {
46     private int percent;
47
48     public SalesPerson(string name, decimal salary, int percent) : base(name, salary)
49     {
50         this.percent = percent;
51     }
52
53     public override void SetBonus(decimal bonus)
54     {
55         if (percent > 200)
56             base.SetBonus(bonus * 3);
57         else if (percent > 100)
58             base.SetBonus(bonus * 2);
59         else
60             base.SetBonus(bonus);
61     }
62 }
```





## TASK 2: Advanced level:

