

# LAB-4

2100030723

Q1) TASK1

```
using System;

public class Employee
{
    private string name;
    private decimal salary;
    private decimal bonus;

    public string Name => name;

    public decimal Salary
    {
        get => salary;
        set => salary = value;
    }

    public Employee(string name, decimal salary)
    {
        this.name = name;
        this.salary = salary;
        this.bonus = 0;
    }

    public virtual void SetBonus(decimal bonus)
    {
        this.bonus = bonus;
    }

    public decimal ToPay()
    {
        return salary + bonus;
    }
}

public class SalesPerson : Employee
{
    private int percent;

    public SalesPerson(string name, decimal salary, int percent) : base(name, salary)
    {
        this.percent = percent;
    }

    public override void SetBonus(decimal bonus)
    {
        if (percent > 200)
        {
            base.SetBonus(bonus * 3);
        }
        else if (percent > 100)
        {
            base.SetBonus(bonus * 2);
        }
    }
}
```

```

        }
        else
        {
            base.SetBonus(bonus);
        }
    }
}

public class Manager : Employee
{
    private int quantity;

    public Manager(string name, decimal salary, int quantity) : base(name,
salary)
    {
        this.quantity = quantity;
    }

    public override void SetBonus(decimal bonus)
    {
        if (quantity > 150)
        {
            base.SetBonus(bonus + 1000);
        }
        else if (quantity > 100)
        {
            base.SetBonus(bonus + 500);
        }
        else
        {
            base.SetBonus(bonus);
        }
    }
}

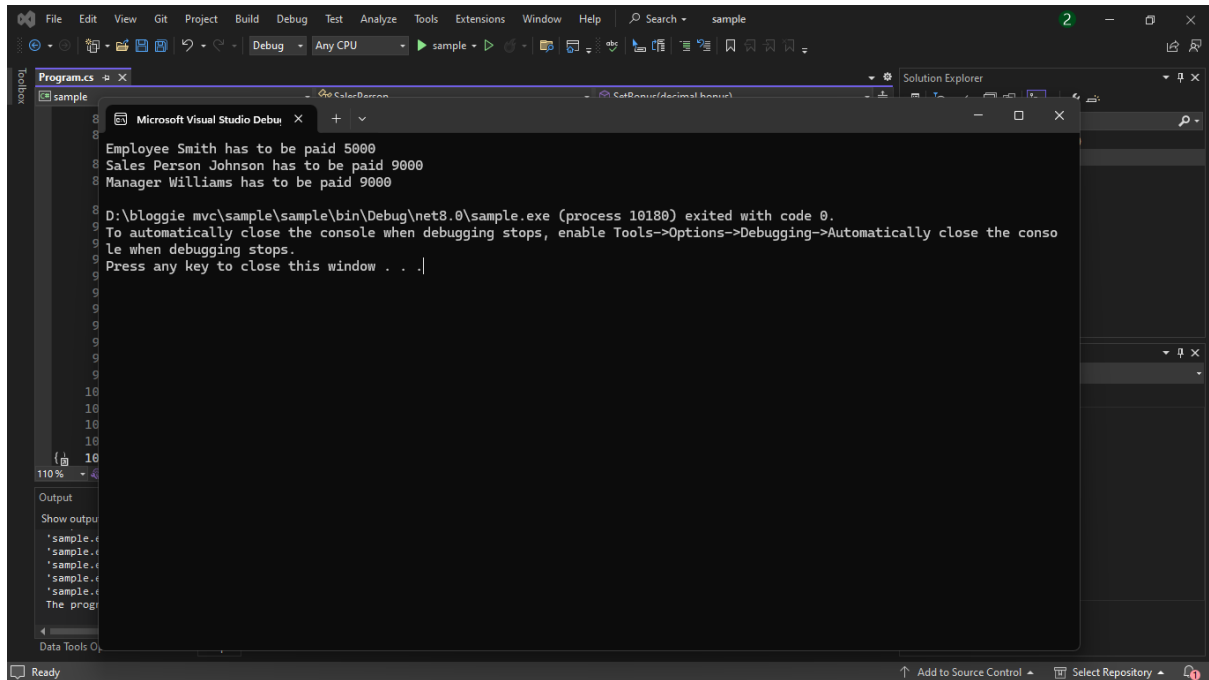
public class Company
{
    public static void Main(string[] args)
    {
        // Testing the classes
        Employee employee = new Employee("Smith", 5000);
        Console.WriteLine($"Employee {employee.Name} has to be paid
{employee.ToPay()}");

        SalesPerson salesPerson = new SalesPerson("Johnson", 6000, 250);
        salesPerson.SetBonus(1000);
        Console.WriteLine($"Sales Person {salesPerson.Name} has to be paid
{salesPerson.ToPay()}");

        Manager manager = new Manager("Williams", 7000, 120);
        manager.SetBonus(1500);
        Console.WriteLine($"Manager {manager.Name} has to be paid
{manager.ToPay()}");
    }
}

```

OUTPUT :



Q2)

```
using System;
```

```
public class Employee
{
    private string name;

    private decimal salary;

    private decimal bonus;

    public string Name => name;

    public decimal Salary
```

```
{  
    get => salary;  
    set => salary = value;  
}
```

```
public Employee(string name, decimal salary)  
{  
    this.name = name;  
    this.salary = salary;  
    this.bonus = 0;  
}
```

```
public virtual void SetBonus(decimal bonus)  
{  
    this.bonus = bonus;  
}
```

```
public decimal ToPay()  
{  
    return salary + bonus;  
}  
}
```

```
public class SalesPerson : Employee  
{  
    private int percent;
```

```
    public SalesPerson(string name, decimal salary, int percent) : base(name, salary)  
    {  
        this.percent = percent;  
    }  
}
```

```

public override void SetBonus(decimal bonus)
{
    if (percent > 200)
    {
        base.SetBonus(bonus * 3);
    }
    else if (percent > 100)
    {
        base.SetBonus(bonus * 2);
    }
    else
    {
        base.SetBonus(bonus);
    }
}
}

```

```

public class Manager : Employee
{
    private int quantity;

```

```

    public Manager(string name, decimal salary, int quantity) : base(name, salary)
    {
        this.quantity = quantity;
    }

```

```

    public override void SetBonus(decimal bonus)
    {
        if (quantity > 150)
        {

```

```
        base.SetBonus(bonus + 1000);
    }
    else if (quantity > 100)
    {
        base.SetBonus(bonus + 500);
    }
    else
    {
        base.SetBonus(bonus);
    }
}
}
```

```
public class Company
{
    private Employee[] employees;

    public Company(Employee[] employees)
    {
        this.employees = employees;
    }

    public void GiveEverybodyBonus(decimal companyBonus)
    {
        foreach (Employee employee in employees)
        {
            employee.SetBonus(companyBonus);
        }
    }

    public decimal TotalToPay()
```

```

{
    decimal totalSalary = 0;
    foreach (Employee employee in employees)
    {
        totalSalary += employee.ToPay();
    }
    return totalSalary;
}

public string NameMaxSalary()
{
    decimal maxSalary = 0;
    string maxSalaryName = "";

    foreach (Employee employee in employees)
    {
        decimal employeeSalary = employee.ToPay();
        if (employeeSalary > maxSalary)
        {
            maxSalary = employeeSalary;
            maxSalaryName = employee.Name;
        }
    }

    return maxSalaryName;
}

}

public class Program
{
    public static void Main(string[] args)

```

```

{
    // Testing the classes

    Employee[] employees = new Employee[]
    {
        new Employee("Smith", 5000),
        new SalesPerson("Johnson", 6000, 250),
        new Manager("Williams", 7000, 120)
    };

    Company company = new Company(employees);

    company.GiveEverybodyBonus(1000);

    Console.WriteLine($"Total salary to pay: {company.TotalToPay()}");

    Console.WriteLine($"Employee with maximum salary: {company.NameMaxSalary()}");
}
}

```

OUTPUT :

