Programming Task : Employee Payroll System (C# Console Application)

Problem Statement : Develop a payroll system using Object-Oriented Programming (OOP) principles to calculate employee salaries based on their roles. The system should be implemented as a C# Console Application.

Features Implemented :

1. Object-Oriented Design (OOP Principles)
2. Role-Based Salary Calculation (Manager, Developer, Intern)
3. Menu-Driven Console Application
4. File Storage (Save and Load Employee Data)
5. Total Payroll Calculation

Program.cs

```
using System;


namespace Employees
{
   internal class Program
   {
      static void Main(string[] args)
      {
          PayrollSystem payroll = new PayrollSystem();


         while (true)
         {
            Console.WriteLine("\nEmployee Payroll System");
            Console.WriteLine("1. Add Employee");
            Console.WriteLine("2. Display All Employees");
            Console.WriteLine("3. Display Total Payroll");
            Console.WriteLine("4. Exit");
            Console.Write("Enter your choice: ");
            int choice = Convert.ToInt32(Console.ReadLine());


            switch (choice)
            {
```

```
            case 1:
                AddEmployeeMenu(payroll);
                break;
            case 2:
                payroll.DisplayAllEmployees();
                break;
            case 3:
                payroll.DisplayTotalPayroll();
                break;
            case 4:
                Console.WriteLine("Exiting program.");
                return;
                break;
        }
    }
}

static void AddEmployeeMenu(PayrollSystem payroll)
{
    Console.WriteLine("EMPLOYEE MENU");
    Console.WriteLine(" 1. Manager \n 2. Developer\n 3. Intern");
    Console.WriteLine("Enter your Choice :");
    int type = Convert.ToInt32(Console.ReadLine());

    Console.Write("Enter Name: ");
    string name = Console.ReadLine();
    Console.Write("Enter ID: ");
    int id = Convert.ToInt32(Console.ReadLine());
    Console.Write("Enter Basic Pay: ");
    double basicPay = Convert.ToDouble(Console.ReadLine());
    Console.Write("Enter Allowances: ");
```

```
            double allowances = Convert.ToDouble(Console.ReadLine());
            Console.Write("Enter Deductions: ");
            double deductions = Convert.ToDouble(Console.ReadLine());

            switch (type)
            {
                case 1:
                    Console.Write("Enter Bonus: ");
                    double bonus = Convert.ToDouble(Console.ReadLine());
                    payroll.AddEmployee(new Manager(name, id, basicPay, allowances, deductions, bonus));
                    break;
                case 2:
                    payroll.AddEmployee(new Developer(name, id, basicPay, allowances, deductions));
                    break;
                case 3:
                    payroll.AddEmployee(new Intern(name, id, basicPay, allowances, deductions));
                    break;
                default:
                    Console.WriteLine("Invalid selection! Returning to main menu.");
                    break;
            }
        }
    }
}
```

BaseEmployee.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace Employees
{
    public abstract class BaseEmployee
    {
        public string Name { get; set; }
        public int ID { get; set; }
        public string Role { get; set; }
        public double BasicPay { get; set; }
        public double Allowances { get; set; }
        public double Deductions { get; set; }

        public BaseEmployee(string name, int id, string role, double basicPay, double allowances, double deductions)
        {
            Name = name;
            ID = id;
            Role = role;
            BasicPay = basicPay;
            Allowances = allowances;
            Deductions = deductions;
        }

        public virtual double CalculateSalary()
        {
            return BasicPay + Allowances - Deductions;
        }

        public virtual void DisplayDetails()
        {
            Console.WriteLine("ID: " + ID + " \n, Name: " + Name + " \n, Role: " + Role + " \n, Salary: " + CalculateSalary().ToString("C"));
```

```
        }
    }
}
```

Manager.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Employees
{
    public class Manager : BaseEmployee
    {
        public double Bonus { get; set; }

        public Manager(string name, int id, double basicPay, double allowances, double deductions,
double bonus)
            : base(name, id, "Manager", basicPay, allowances, deductions)
        {
            Bonus = bonus;
        }

        public override double CalculateSalary()
        {
            return base.CalculateSalary() + Bonus;
        }

        public override void DisplayDetails()
```

```
        {
            base.DisplayDetails();
            Console.WriteLine("Bonus: " + Bonus.ToString("C"));

        }
    }
}
```

Developer.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Employees
{
    public class Developer : BaseEmployee
    {
        public Developer(string name, int id, double basicPay, double allowances, double deductions)
            : base(name, id, "Developer", basicPay, allowances, deductions)
        {
        }
    }
}
```

Intern.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

namespace Employees
{
    public class Intern : BaseEmployee
    {
        public Intern(string name, int id, double basicPay, double allowances, double deductions)
            : base(name, id, "Intern", basicPay, allowances, deductions)
        {
        }
    }
}
```

PayrollSystem.cs

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Employees
{
    public class PayrollSystem
    {
        private List<BaseEmployee> employees = new List<BaseEmployee>();
        private string filePath = "employees.txt";

        public PayrollSystem()
        {
            LoadFromFile();
        }

        public void AddEmployee(BaseEmployee employee)
```

```
{
    employees.Add(employee);
    SaveToFile();
    Console.WriteLine("Employee added and saved!");
}

public void DisplayAllEmployees()
{
    if (employees.Count == 0)
    {
        Console.WriteLine("No employees found!");
        return;
    }
    foreach (var emp in employees)
    {
        emp.DisplayDetails();
        Console.WriteLine("");
    }
}

public void DisplayTotalPayroll()
{
    double totalPayroll = 0;
    foreach (var emp in employees)
    {
        totalPayroll += emp.CalculateSalary();
    }
    Console.WriteLine("Total Payroll Amount: " + totalPayroll.ToString("C"));
}

private void SaveToFile()
```

```
    {
        using (StreamWriter writer = new StreamWriter(filePath))
        {
            foreach (var emp in employees)
            {
                writer.WriteLine(emp.Name + "," + emp.ID + "," + emp.Role + "," + emp.BasicPay + "," +
emp.Allowances + "," + emp.Deductions);


            }
        }
    }

    private void LoadFromFile()
    {
        if (File.Exists(filePath))
        {
            var lines = File.ReadAllLines(filePath);
            foreach (var line in lines)
            {
                var parts = line.Split(',');
                string name = parts[0];
                int id = int.Parse(parts[1]);
                string role = parts[2];
                double basicPay = double.Parse(parts[3]);
                double allowances = double.Parse(parts[4]);
                double deductions = double.Parse(parts[5]);

                if (role == "Manager")
                    employees.Add(new Manager(name, id, basicPay, allowances, deductions, 1500));
                else if (role == "Developer")
                    employees.Add(new Developer(name, id, basicPay, allowances, deductions));
```

```
            else if (role == "Intern")
                employees.Add(new Intern(name, id, basicPay, allowances, deductions));
        }
    }
  }
 }
}
```

Output

```
C:\Users\HP\Desktop\ASP cor

Employee Payroll System
1. Add Employee
2. Display All Employees
3. Display Total Payroll
4. Exit
Enter your choice: 1
EMPLOYEE MENU
 1. Manager
 2. Developer
 3. Intern
Enter your Choice :
2
Enter Name: Nikila
Enter ID: 589
Enter Basic Pay: 250000
Enter Allowances: 5000
Enter Deductions: 2000
Employee added and saved!
```

```
ID: 589
, Name: Ann
, Role: Manager
, Salary: ? 26,350.00
Bonus: ? 1,500.00

ID: 548
, Name: nandu
, Role: Manager
, Salary: ? 6,600.00
Bonus: ? 1,500.00

ID: 123
, Name: Jaya
, Role: Manager
, Salary: ? 6,500.00
Bonus: ? 1,500.00

ID: 589
, Name: Nikila
, Role: Developer
, Salary: ? 2,53,000.00


Employee Payroll System
1. Add Employee
2. Display All Employees
3. Display Total Payroll
4. Exit
Enter your choice:
```

```
ID: 589
, Name: Nikila
, Role: Developer
, Salary: ? 2,53,000.00


Employee Payroll System
1. Add Employee
2. Display All Employees
3. Display Total Payroll
4. Exit
Enter your choice: 3
Total Payroll Amount: ? 3,28,720.00
Total Payroll Amount: ? 3,28,720.00

Employee Payroll System
1. Add Employee
2. Display All Employees
3. Display Total Payroll
4. Exit
Enter your choice:
```

CS0162 Unreachable code detected Employees