**Task Performance 01**

Syntax:

using System;

namespace EmployeeInterface{

public interface IEmployee

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Department { get; set; }

public string JobTitle { get; set; }

public double BasicSalary { get; set; }

public void computeSalary(int hoursWorked, double ratePerHour);

}

}

namespace EmployeeNamespace2

{

using EmployeeInterface;

public class PartTimeEmployee : IEmployee

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Department { get; set; }

public string JobTitle { get; set; }

public double BasicSalary { get; set; }

public PartTimeEmployee(string FName, string LName, string dept, string job)

{

FirstName = FName;

LastName = LName;

Department = dept;

JobTitle = job;

}

public void computeSalary(int hoursWorked, double ratePerHour)

{

BasicSalary = hoursWorked \* ratePerHour;

}

public double getSalary()

{

return BasicSalary;

}

}

public class MainClass()

{

public static void Main(string[] args)

{

String FName, LName, Dept, Job;

int hoursWorked;

double ratePerHour;

Console.WriteLine("Employee Salary Calculator\n");

Console.Write("First name : ");

FName = Console.ReadLine();

Console.Write("Last name : ");

LName = Console.ReadLine();

Console.Write("Department : ");

Dept = Console.ReadLine();

Console.Write("Job title : ");

Job = Console.ReadLine();

Console.WriteLine("\n----------------------------------\n");

Console.Write("Rate per hour : ");

ratePerHour = Convert.ToDouble(Console.ReadLine());

Console.Write("Total hours worked : ");

hoursWorked = Convert.ToInt32(Console.ReadLine());

PartTimeEmployee employee = new PartTimeEmployee(FName, LName, Dept, Job);

employee.computeSalary(hoursWorked, ratePerHour);

Console.WriteLine("\n----------------------------------\n");

Console.WriteLine("First name : " + employee.FirstName);

Console.WriteLine("Last name : " + employee.LastName);

Console.WriteLine("Basic Salary : " + employee.getSalary());

}

}

}

Output:

