**Name : Shahzadi Begum Shaikh Rafique**

**Assignment : C Sharp 3rd Assignment**

# Inheritance and Polymorphism

## Objective

* To use inheritance (“is-a” relationship).
* To use polymorphism.
* To create an abstract class
* To use interface.

## Assignments to be done in this session

1. Create a hierarchy of Employee, Manager, MarketingExecutive in Employee Management System. They should have the following functionality.
   1. Manager with following private members.
      * Petrol Allowance: 8 % of Salary.
      * Food Allowance : 13 % of Salary.
      * Other Allowances : 3% of Salary.

Calculate GrossSalary by adding above allowances. Override CalculateSalary() method to calculate Net Salary. NetSalary. PF calculation should not consider above allowances.

* 1. MarketingExecutive with following private members.
     + Kilometer travel
     + Tour Allowances : Rs 5/- per Kilometer (Automatically generated).
     + Telephone Allowances : Rs.1000/-

Calculate GrossSalary by adding above allowances. Override CalculateSalary(). NetSalary,PF calculation should not consider above allowances.

Implement IPrintable interface for every Employee which will allow to print details of Employee on console.

1. Write a class called MyStack with following members.
   1. integer array
   2. integer variable to store top position
   3. size of the array.

Implement Push() and Pop() operation. Implement ICloneable interface to perform cloning. Write a client application to perform cloning.

1. Create a custom exception class named StackException. The Push()and Pop() method should throw object of StackException when the stack is full or empty respectively.

Code :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AssignmentCSharp3

{

class Employee

{

public int EmpId;

public string EmpName;

public double Basesalary = 30000;

public double kilometer;

public void getInfo()

{

Console.WriteLine("\nEmployee Details: ");

Console.WriteLine("\nEnter Your Employee ID: ");

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

Console.WriteLine("Enter Your Employee Name: ");

EmpName = Console.ReadLine();

}

public void Show()

{

Console.WriteLine("\nDisplay Detail");

Console.WriteLine("\nEmployee ID : " + EmpId);

Console.WriteLine("Employee Name: " + EmpName);

}

public void getKiloMeterInfo()

{

Console.WriteLine("\nEnter No Of Kilomerters You Travled :");

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

}

public void CalculateSalary()

{

Console.WriteLine("\nYour Basic Salary : " + Basesalary);

}

}

class Manager : Employee

{

public void getEmpInfo()

{

getInfo();

base.Show();

base.CalculateSalary();

}

public void CalculateSalary()

{

double NetSal;

double GrosSal;

GrosSal =base.Basesalary + (base.Basesalary \* 0.08) +

(base.Basesalary \* 0.13) + (base.Basesalary \* 0.03);

Console.WriteLine("Your Gross Salary is : " + GrosSal);

NetSal = GrosSal - 1800;

Console.WriteLine("Your NetSalary is : " + NetSal);

}

}

class MarketingExe : Employee

{

public void getEmpInfo()

{

base.getInfo();

base.Show();

base.getKiloMeterInfo();

base.CalculateSalary();

}

public void CalculateSalary()

{

double NetSal;

double GrosSal;

GrosSal = base.Basesalary + (base.kilometer / 5) + 1000;

Console.WriteLine("Your Gross Salary is : " + GrosSal);

NetSal = GrosSal - 1800;

Console.WriteLine("Your NetSalary is : " + NetSal);

}

}

class Program

{

static void Main(string[] args)

{

Manager mngr = new Manager();

mngr.getEmpInfo();

mngr.CalculateSalary();

MarketingExe market = new MarketingExe();

market.getEmpInfo();

market.CalculateSalary();

Console.ReadKey();

}

}

}

**Output**

