

## Question No 1:

### Employee.h

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Employee
{
private:
    string firstName, lastName;
    int empId;
public:
    Employee()
    {
        firstName = "";
        lastName = "";
        empId = 0;
    }
    Employee(string fn, string ln, int ei)
    {
        firstName = fn;
        lastName = ln;
        empId = ei;
    }
    void setLastName(string ln)
    {
        lastName = ln;
    }
    void setFirstName(string fn)
    {
        firstName = fn;
    }
    void setempID(int ei)
    {
        empId = ei;
    }
    string getFirstName()
    {
        return firstName;
    }
    string getLastname()
    {
        return lastName;
    }
    int getempID()
    {
        return empId;
    }
    void print()
    {
        cout << endl << endl;
        cout << "ID : " << getempID() << endl;
        cout << "\nName : " << getFirstName() << " " << getLastname() << endl;
    }
    virtual double earnings() = 0;
};
```

### SalariedEmployee.h

```
#pragma once
#include "Employee.h"
#include<iostream>
using namespace std;
class SalariedEmployee :
    public Employee
{
private:
    double weeklySalary;
public:
    SalariedEmployee() :Employee()
    {
        weeklySalary = 0.0;
    }
    SalariedEmployee(string fn, string ln, int ei, double ws) : Employee(fn, ln, i)
    {
        weeklySalary = ws;
    }
    void setWeeklySalary(double ws)
    {
        if (ws < 0.0)
        {
            cout << "\nError.....\nSalary can't be Negative.\n" << endl;
            weeklySalary = 0.0;
        }
        else
            weeklySalary = ws;
    }
    double getWeeklySalary()
    {
        return weeklySalary;
    }
    double earnings()
    {
        return getWeeklySalary();
    }
    void printWeeklySalary()
    {
        Employee::print();
        cout << "\nSalary : " << earnings() << endl << endl;
        cout << "-----" << endl;
        cout << "-----" << endl;
    }
};
```

### HourlyEmployee.h

```
#pragma once
#include "Employee.h"
#include<iostream>
using namespace std;
class HourlyEmployee :
    public Employee
{
private:
```

```
    double wage, hours;
public:
    HourlyEmployee() :Employee()
    {
        wage = 0.0;
        hours = 0.0;
    }
    HourlyEmployee(string fn, string ln, int ei, double w, double h) : Employee(fn,ln,ei)
    {
        wage = w;
        hours = h;
    }
    void setwage(double w)
    {
        if (w < 0.0)
        {
            cout << "/nError...../nWage can't be negative...../n" << endl;
            wage = 0.0;
        }
        else
            wage = w;
    }
    void sethours(double h)
    {
        if (h < 168.0 && h>0.0)
            hours = h;
        else
        {
            cout << "/nError...../nWrong hours...../n" << endl;
            hours = 0.0;
        }
    }
    double getwage()
    {
        return wage;
    }
    double gethours()
    {
        return hours;
    }
    double earnings()
    {
        if (hours <= 40.0)
        {
            return wage*hours;
        }
        else
        {
            return wage*40 + (hours - 40.0)*wage*1.5;
        }
    }
    void printHourlySalary()
    {
        Employee::print();
        cout << "\nSalary : " << earnings() << endl << endl;
        cout << "-----" << endl;
        cout << "-----" << endl;
    }
}
```

```
};
```

### ComissionEmployee.h

```
#pragma once
#include "Employee.h"
#include<iostream>
using namespace std;
class ComissionEmployee : 
    public Employee
{
private:
    double grossSales, comissionRates;
public:
    ComissionEmployee() :Employee()
    {
        grossSales = 0.0;
        comissionRates = 0.0;
    }
    ComissionEmployee(string fn, string ln, int ei, double gs, double cr) : Employee(fn,
ln, ei)
    {
        grossSales = gs;
        comissionRates = cr;
    }
    void setGrossSales(double gs)
    {
        if (gs < 0.0)
        {
            cout << "\nError.....\nGross Sale can't be Negative.\n" << endl;
            grossSales = 0.0;
        }
        else
            grossSales = gs;
    }
    void setComRate(double cr)
    {
        if (cr>0.0 && cr < 1.0)
        {
            comissionRates = cr;
        }
        else
        {
            cout << "/nError...../nWrong Comission Rate...../n" << endl;
            comissionRates = 0.0;
        }
    }
    double getGrossSales()
    {
        return grossSales;
    }
    double getComRate()
    {
        return comissionRates;
    }
    double earnings()
    {
```

```

        return getGrossSales()*getComRate();
    }
    void printComSalary()
    {
        Employee::print();
        cout << "\nSalary : " << earnings() << endl << endl;
        cout << "-----" << endl;
        cout << "-----" << endl;
    }
};


```

### TestEmployee.h

```

#pragma once
#include "SalariedEmployee.h"
#include "HourlyEmployee.h"
#include "ComissionEmployee.h"
#include <iostream>
using namespace std;
int se_count=0, he_count=0, ce_count=0;
string fname, lname;
int eID;
class TestEmployee
{
    SalariedEmployee se[20];
    HourlyEmployee he[20];
    ComissionEmployee ce[20];

public:
    void basicInfo()
    {
        cout << "\n\nEnter First Name : ";
        cin >> fname;
        cout << "Enter Last Name : ";
        cin >> lname;
        cout << "Enter Employee ID : ";
        cin >> eID;
    }
    void read(int ch)
    {
        switch (ch)
        {
        case 1:
            system("cls");
            double weeklySal;
            cout << "\n\t*** SALARIED EMPLOYEE ***" << endl;
            cout << "\t-----" << endl;
            basicInfo();
            se[se_count].setFirstName(fname);
            se[se_count].setLastName(lname);
            se[se_count].setempID(eID);
            cout << "Enter Weekly Salary : ";
            cin >> weeklySal;
            se[se_count].setWeeklySalary(weeklySal);
            se_count++;
            cout << endl << endl;
            system("pause");
            break;
        }
    }
};


```

```

case 2:
    system("cls");
    double enterHours, enterWage;
    cout << "\n\t*** HOURLY EMPLOYEE ***" << endl;
    cout << "\t-----" << endl;
    basicInfo();
    he[he_count].setFirstName(fname);
    he[he_count].setLastName(lname);
    he[he_count].setempID(eID);
    cout << "Enter Hours Worked : ";
    cin >> enterHours;
    cout << "Enter Wage : ";
    cin >> enterWage;
    he[he_count].sethours(enterHours);
    he[he_count].setwage(enterWage);
    he_count++;
    cout << endl << endl;
    system("pause");
    break;
case 3:
    system("cls");
    double enterSale, enterRate;
    cout << "\n\t*** COMISSION EMPLOYEE ***" << endl;
    cout << "\t-----" << endl;
    basicInfo();
    ce[ce_count].setFirstName(fname);
    ce[ce_count].setLastName(lname);
    ce[ce_count].setempID(eID);
    cout << "Enter Gross Sale : ";
    cin >> enterSale;
    cout << "Enter Comission Rate (%) : ";
    cin >> enterRate;
    ce[ce_count].setGrossSales(enterSale);
    ce[ce_count].setComRate(enterRate / 100);
    ce_count++;
    cout << endl << endl;
    system("pause");
    break;
case 4:break;
default:
    cout << "\n\nWrong Choice.....\n" << endl;
}
}
void display(int ch)
{
    switch (ch)
    {
    case 1:
        system("cls");
        cout << "\n\t*** SALARIED EMPLOYEE ***" << endl;
        cout << "\t-----" << endl;
        for (int i = 0; i < se_count; i++)
        {
            se[i].printWeeklySalary();
        }
        system("pause");
        break;
    case 2:

```

```

        system("cls");
        cout << "\n\t*** HOURLY EMPLOYEE ***" << endl;
        cout << "\t-----" << endl;
        for (int i = 0; i < he_count; i++)
        {
            he[i].printHourlySalary();
        }
        system("pause");
        break;
    case 3:
        system("cls");
        cout << "\n\t*** COMISSION EMPLOYEE ***" << endl;
        cout << "\t-----" << endl;
        for (int i = 0; i < ce_count; i++)
        {
            ce[i].printComSalary();
        }
        system("pause");
        break;
    case 4:break;
    default:
        cout << "\n\nWrong Choice.....\n" << endl;
    }
}
};


```

### TestEmployee.cpp

```

#include "TestEmployee.h"

#include<iostream>
using namespace std;
void menu()
{
    cout << "\n\n _____" << endl;
    cout << "| \t\ t\ t\ t\ t| " << endl;
    cout << "| \t\ tEMPLOYEE DATA\ t\ t| " << endl;
    cout << "| _____| " << endl;
    cout << "| \t\ t\ t\ t\ t| " << endl;
    cout << "| \t1. Add new employee\ t\ t| " << endl;
    cout << "| \t2. Display Employee\ t\ t| " << endl;
    cout << "| \t3. Exit\ t\ t\ t| " << endl;
    cout << "| _____| \n" << endl;
}
void SubMenu_2()
{
    cout << "\n\n _____" << endl;
    cout << "| \t\ t\ t\ t\ t| " << endl;
    cout << "| \t\ tNEW EMPLOYEE\ t\ t| " << endl;
    cout << "| _____| " << endl;
    cout << "| \t\ t\ t\ t\ t| " << endl;
    cout << "| \t1. Weekly Employee\ t\ t| " << endl;
    cout << "| \t2. Hourly Employee\ t\ t| " << endl;
    cout << "| \t3. Comission Employee\ t\ t| " << endl;
    cout << "| \t4. Back\ t\ t\ t\ t| " << endl;
    cout << "| _____| \n" << endl;
}
void SubMenu_1()

```

```

{
    cout << "\n\n _____" << endl;
    cout << "| \t \t \t \t \t |" << endl;
    cout << "| \t \t DISPLAY EMPLOYEE \t |" << endl;
    cout << "| _____ |" << endl;
    cout << "| \t \t \t \t \t |" << endl;
    cout << "| \t1. Weekly Employee Detail \t |" << endl;
    cout << "| \t2. Hourly Employee Detail \t |" << endl;
    cout << "| \t3. Comission Employee Detail \t |" << endl;
    cout << "| \t4. Back \t \t \t |" << endl;
    cout << "| _____ | \n" << endl;
}

int main()
{
    TestEmployee te;
    int choice, ch;
    do
    {
        system("cls");
        menu();
        cout << "Enter Choice : ";
        cin >> choice;
        switch (choice)
        {
        case 1:
            do
            {
                system("cls");
                SubMenu_2();
                cout << "\nEnter Choice : ";
                cin >> ch;
                te.read(ch);
            } while (ch != 4);
            break;
        case 2:
            do
            {
                system("cls");
                SubMenu_1();
                cout << "Enter Choice : ";
                cin >> ch;
                te.display(ch);
            } while (ch != 4);
            break;
        case 3:break;
        default:cout << "\n\nWrong Choice.....\n" << endl;
        }
    }while (choice != 3);
    return 0;
}

```