OOP Lab Assignment 3

Jaynam Modi. PG-43. G3. Aug 14, 2020

- Problem Statement:

A company pays its employees on a weekly basis. The employees are of four types:

1. Salaried employees are paid a fixed weekly salary regardless of the number of hours worked,
2. hourly employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours,
3. commission employees are paid a percentage of their sales and
4. Salaried-commission employees receive a base salary plus a percentage of their sales.

For the current pay period, the company has decided to reward salaried-commission employees by adding 10% to their base salaries.

The company wants to implement an Object Oriented application that performs its payroll calculations polymorphically.

- Objectives:

1. To demonstrate polymorphism in inheritance
2. To assign a subclass reference to a superclass variable
3. To make systems extensible and maintainable using polymorphism

- Theory:

- Concept of Polymorphism

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to

perform different tasks. This allows us to perform a single action in different ways.

- Function Overloading

Function overloading is a feature in C++ where two or more functions can have the same name but different parameters.

Function overloading can be considered as an example of polymorphism feature in C++.

- Base class and derived class

Classes in C++ can be extended, creating new classes which retain characteristics of the base class. This process, known as inheritance, involves a base class and a derived class: The derived class inherits the members of the base class, on top of which it can add its own members.

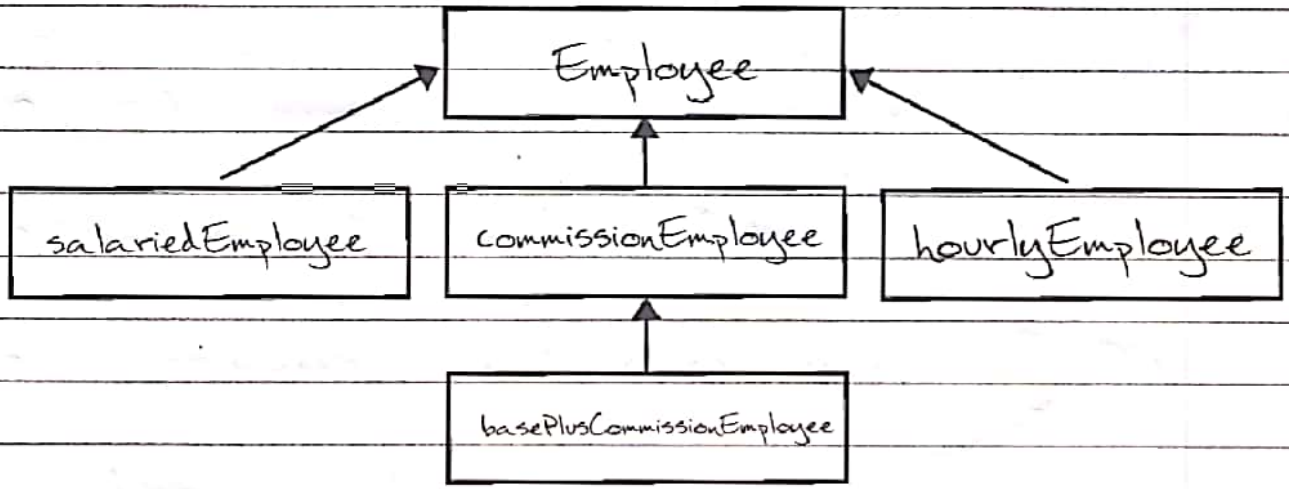
- Algorithm:

1. Receive Input from User.

2. Use Polymorphic Processing in order to calculate salary.

3. Display Results.

- Class Diagram:



- Platform:

  - 64-bit Open source Linux or its dderivatives.
  - Open Source C++ Programming tool like G++ / Eclipse Editor.

- Input:

Employee details like Employee first name, last name, social security number, earnings,

weekly salary, number of hours worked, basic salary etc.

- Output:

Polymorphically calculated payroll.

- Conclusion:

We have implemented object oriented application that performs its payroll calculations for different types of employees polymorphically.

- FAQs:

1. What is inheritance in C++ and give examples of the different types of inheritance?

> Inheritance is a mechanism of acquiring the features and behaviors of a class by another class. The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class.

Different types of inheritance are :

1. Single Inheritance

2. Multiple Inheritance

3. Multilevel Inheritance

4. Hierarchical Inheritance

5. Hybrid Inheritance

2. What's the difference between public, private, and protected?

> A public member is accessible from anywhere outside the class but within a program. You can set and get the value of public variables without any member.

A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members.

A protected member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.

3. Why can't derived class access private things from base class?

> Only the class and friend functions can access private members.

```cpp
#include <iostream>

using namespace std;

class Employee {
    private:
        string name;
        string ssn;
    public:
        Employee(string a, string b){
            name = a;
            ssn = b;
        }

        virtual void print(){
            cout << " > Name : " << name << endl << " > SSN : " << ssn << endl;
        }

        virtual double payroll() = 0;
};

class salariedEmployee : Employee {
    private:
        double weeklysal;
    public:
        salariedEmployee(string a, string b, double salary): Employee(a, b){
            weeklysal = salary;
        }

        void print(){
            cout << "SALARIED EMPLOYEE : " << endl;
            Employee::print();
            cout << " > Weekly Salary : " << weeklysal << endl;
            cout << " > Payroll : " << payroll() << endl;
        }

        double payroll(){
            return weeklysal;
        }
};

class hourlyEmployee : Employee {
    private:
        double rate;
        double hours;
    public:
        hourlyEmployee(string a, string b, double r, double h): Employee(a, b){
            rate = r;
            hours = h;
        }

        void print(){
            cout << "HOURLY EMPLOYEE : " << endl;
            Employee::print();
            cout << " > Hourly Rate : " << rate << endl << " > Hours Worked : " << hours << endl << " > Payroll : " << payroll() << endl;
        }

        double payroll(){
            if(hours <= 40){
                return rate * hours;
            } else {
                return rate * 40 + ((hours - 40) * rate * 1.5);
            }
        }
};

class commissionEmployee : Employee {
    private:
        double sales;
        double rate;
    public:
        commissionEmployee(string a, string b, double s, double r): Employee(a, b){
            sales = s;
            rate = r;
        }

        void print(){
            cout << "COMMISSION EMPLOYEE : " << endl;
            Employee::print();
            cout << " > Gross Sales : " << sales << endl << " > Rate : " << rate << endl << " > Payroll : " << payroll() << endl;
        }

        double payroll(){
            return rate * sales;
        }
};

class basePlusCommissionEmployee : commissionEmployee {
    private:
        double base;
    public:
        basePlusCommissionEmployee(string a, string b, double s, double r, double bs):commissionEmployee(a, b, s, r){
            base = bs;
        }

        void print(){
            cout << "BASE + COMMISSION EMPLOYEE : " << endl;
            commissionEmployee::print();
            cout << " > Base Salary : " << base << endl;
        }

        double payroll(){
            return commissionEmployee::payroll() + base + base * 0.1;
        }
};


int main(){

    salariedEmployee s1("Will Smith", "111-11-1111", 4500.00);
    hourlyEmployee s2("Keanu Reeves", "222-22-2222", 100.00, 45.5);
    commissionEmployee s3("Ryan Reynolds", "333-33-3333", 15000.00, 0.3);
    basePlusCommissionEmployee s4("Morgan Freeman", "444-44-4444", 20000.00, 0.25, 5000.00);

    s1.print();
    s2.print();
    s3.print();
    s4.print();
}
```

```
u0_a362@localhost:~$ ./a.out
SALARIED EMPLOYEE :
 > Name : Will Smith
 > SSN : 111-11-1111
 > Weekly Salary : 4500
 > Payroll : 4500
HOURLY EMPLOYEE :
 > Name : Keanu Reeves
 > SSN : 222-22-2222
 > Hourly Rate : 100
 > Hours Worked : 45.5
 > Payroll : 4825
COMMISSION EMPLOYEE :
 > Name : Ryan Reynolds
 > SSN : 333-33-3333
 > Gross Sales : 15000
 > Rate : 0.3
 > Payroll : 4500
BASE + COMMISSION EMPLOYEE :
COMMISSION EMPLOYEE :
 > Name : Morgan Freeman
 > SSN : 444-44-4444
 > Gross Sales : 20000
 > Rate : 0.25
 > Payroll : 10500
 > Base Salary : 5000
u0_a362@localhost:~$
```