

```
#include<iostream>
#include<string>
using namespace std;
class Employee {
private:
int no;
string name;
public:
virtual void salary () = 0 {
cout << "Enter employee number: ";
cin >> no;
cout << "Enter employee name: ";
cin >> name;
}
virtual void display() = 0 {
cout << "Employee Number is " << no << endl;
cout << "Employee Name is " << name << endl;
}
};

class SalariedEmployee: public Employee
{
private:
double monthlySalary;
public:
void salary()
{
Employee::salary();
cout << "Enter employee monthly salary: "; cin >> monthlySalary;
}
void display()
{
```

```
cout<<endl<<"Salaried Employee:"<<endl;
Employee::display();
cout << "Salary: " << monthlySalary << endl;
}
};

class HourlyEmployee: public Employee
{
private:
    double hours;
    double rate;
public:
    void salary()
    {
        Employee::salary();
        cout << "Enter hours: ";
        cin >> hours;
        cout << "Enter hourly rate: ";
        cin >> rate;
    }
    void display()
    {
        cout<<endl<<"Hourly Employee:"<<endl;
        Employee::display();
        cout <<"Hours: " << hours << endl;
        cout <<"Hourly rate: " << rate << endl;
        cout <<"Salary: " <<(rate*hours) << endl;
    }
};

class CommissionedEmployee: public Employee
{
private:
```

```
double salarydouble;
int overtime;
public:
void salary()
{
Employee::salary();
cout << "Enter salary: ";
cin >> salarydouble;
cout << "Enter overtime: ";
cin >> overtime;
}
void display()
{
cout<<endl<<"Commissioned Employee:"<<endl;
Employee::display();
cout <<"Salary: " << salarydouble << endl;
cout <<"Overtime: " << overtime << endl;
cout <<"Total Salary: " << (salarydouble+(overtime*15)) << endl;
}
};

int main ()
{
Employee employees[3];
SalariedEmployee newSalariedEmployee;
Hourly Employee newHourlyEmployee; Commissioned Employee
newCommissionedEmployee;
employees[0]=&newSalaried Employee;
employees[1]=&newHourlyEmployee;
employees[2]=&newCommissioned Employee;
for(int i=0;i<3;i++){
employees[i]->salary();
```

```
cout<<endl;
}
for(int i=0;i<3;i++) {
employees[i]->display();
}
system ("pause");
return 0;
}
```