<u>Q:</u> A company pays its employees on a weekly basis. The employees are of four types: Salaried employees are paid a fixed weekly salary regardless of the number of hours worked, hourly employees are paid by the hour and receive overtime pay (i.e., 1.5 times their hourly salary rate) for all hours worked in excess of 40 hours, commission employees are paid a percentage of their sales and base-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. Write a c++ program that performs payroll calculations of the company polymorphically.

<u>Answer:</u>

<u>Source Code:</u>

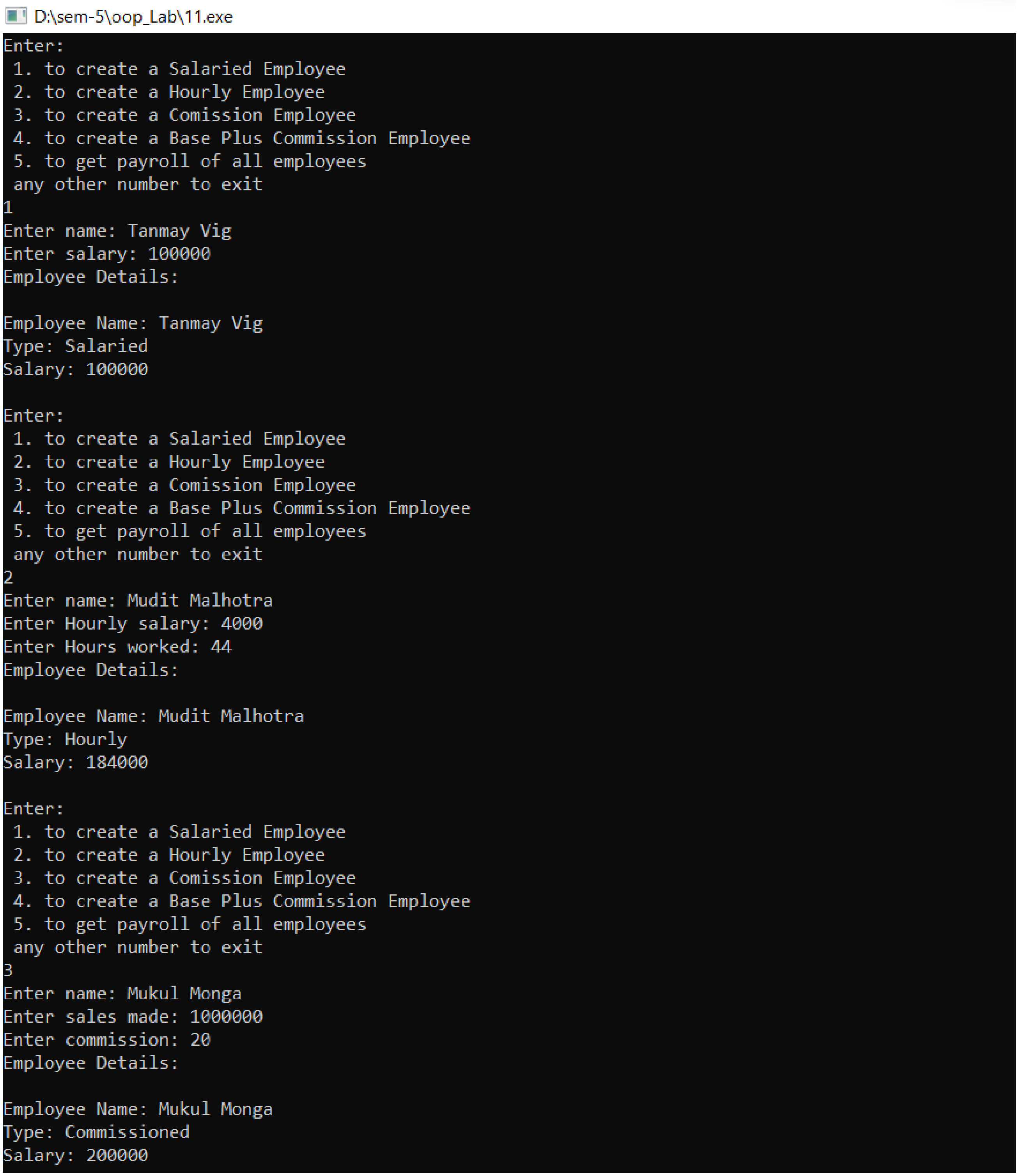
```
#include<iostream>
#include<string>

using namespace std;

class Employee {
  protected:
    string name;
    float salary;
  public:
    Employee(){ }
    Employee(string n, float s){
      name=n;salary=s;
    }
    virtual void getDetail(){
      cout<<"\nEmployee Name: "<<name<<"\nSalary: "<<salary<<"\n\n";
    }
    virtual float getSalary(){
      return salary;
    }
};

class SalariedEmployee : public Employee {
  public:
    SalariedEmployee(string name, float salary): Employee(name, salary){ }
    void getDetail(){
      cout<<"\nEmployee Name: "<<name<<"\nType: "<<"Salaried\nSalary: "<<salary<<"\n\n";
    }
};

class HourlyEmployee : public Employee
{
  protected:
    int hours;
  public:
    HourlyEmployee(string name, float salary, int hrs): Employee(name, salary){
      hours=hrs;
    }
    float getsalary(){
      if(hours<=40) return salary*hours;
      else return salary*40 +1.5*salary*(hours-40);
    }
    void getDetail(){
      cout<<"\nEmployee Name: "<<name<<"\nType: "<<"Hourly\nSalary: "<<getsalary()<<"\n\n";
    }
};

class CommissionEmployee : public Employee {
  protected:
    float commission;
  public:
    CommissionEmployee(string name,float salary,float c):Employee(name, salary){
      commission = c;
    }
    virtual float getSalary(){
      return salary*commission/100;
    }
    virtual void getDetail(){
      cout<<"\nEmployee Name: "<<name<<"\nType: Commissioned\nSalary: "<<getSalary()<<"\n\n";
    }

};

class BasePlusCommissionEmployee : public CommissionEmployee{
  protected:
    float base_amount;
    float increment;
  public:
    BasePlusCommissionEmployee(string name, float salary, float base, float c, float incr):CommissionEmployee(name, salary,c){
      base_amount = base;
      increment = incr;
    }
    float getSalary(){
      return base_amount + base_amount*increment/100 + salary*commission/100;
    }
    void getDetail(){
      cout<<"\nEmployee Name: "<<name<<"\nType: Commissioned\nSalary: "<< getSalary()<<"\n\n";
    }

};

int main(){
  Employee* arr[100];
  int count=0;
  while(true){
    int choice = 0;
```

```
cout<<"Enter:\n 1. to create a Salaried Employee\n 2. to create a Hourly Employee\n 3. to create a Comission Employee\n 4. to create a Base Plus Commission Employee\n 5. to get payroll of all employees\n any other number to exit\n";
cin>>choice;
string name;
float salary;
switch(choice){
  case 1:{
    count++;
    cout<<"Enter name: ";
    cin.clear();
    cin.sync();
    getline(cin,name);
    cout<<"Enter salary: ";
    cin>>salary;
    arr[count-1]=new SalariedEmployee(name, salary);
    cout<<"Employee Details:\n";
    arr[count-1]->getDetail();
    break;
  }
  case 2: {
    count++;
    int hrs;
    cout<<"Enter name: ";
    cin.clear();
    cin.sync();
    getline(cin,name);
    cout<<"Enter Hourly salary: ";
    cin>>salary;
    cout<<"Enter Hours worked: ";
    cin>>hrs;
    arr[count-1]=new HourlyEmployee(name, salary,hrs);
    cout<<"Employee Details:\n";
```

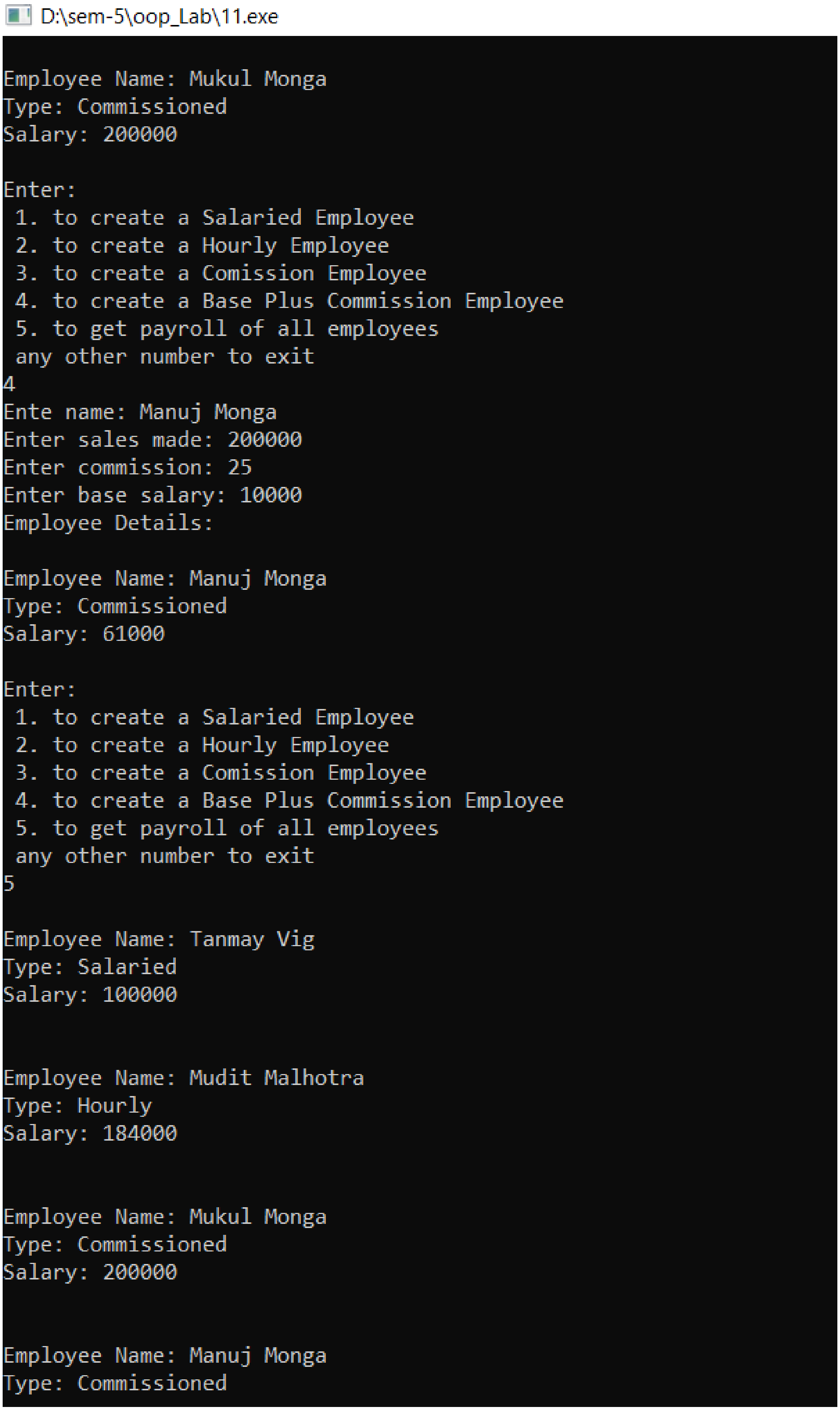
```
   arr[count-1]->getDetail();
   break;
}
case 3: {
   count++;
   float com;
   cout<<"Enter name: ";
   cin.clear();
   cin.sync();
   getline(cin,name);
   cout<<"Enter sales made: ";
   cin>>salary;
   cout<<"Enter commission: ";
   cin>>com;
   arr[count-1]=new CommissionEmployee(name, salary,com);
   cout<<"Employee Details:\n";
   arr[count-1]->getDetail();
   break;
}
case 4: {
   count++;
   float com, base;
   cout<<"Ente name: ";
   cin.clear();
   cin.sync();
   getline(cin,name);
   cout<<"Enter sales made: ";
   cin>>salary;
   cout<<"Enter commission: ";
   cin>>com;
   cout<<"Enter base salary: ";
   cin>>base;
```

```
                arr[count-1]=new BasePlusCommissionEmployee(name, salary,base, com, 10.0);
                cout<<"Employee Details:\n";
                arr[count-1]->getDetail();
                break;
            }
            case 5:{
                float sum=0;
                for(int i=0;i<count;i++){
                    arr[i]->getDetail();
                    sum+=arr[i]->getSalary();
                }
                cout<<"Total spenditure: "<<sum<<endl;
                break;
            }
            default:{
                choice=0;
                break;
            }
        }
        if(choice==0) break;
    }

    return 0;
}
```

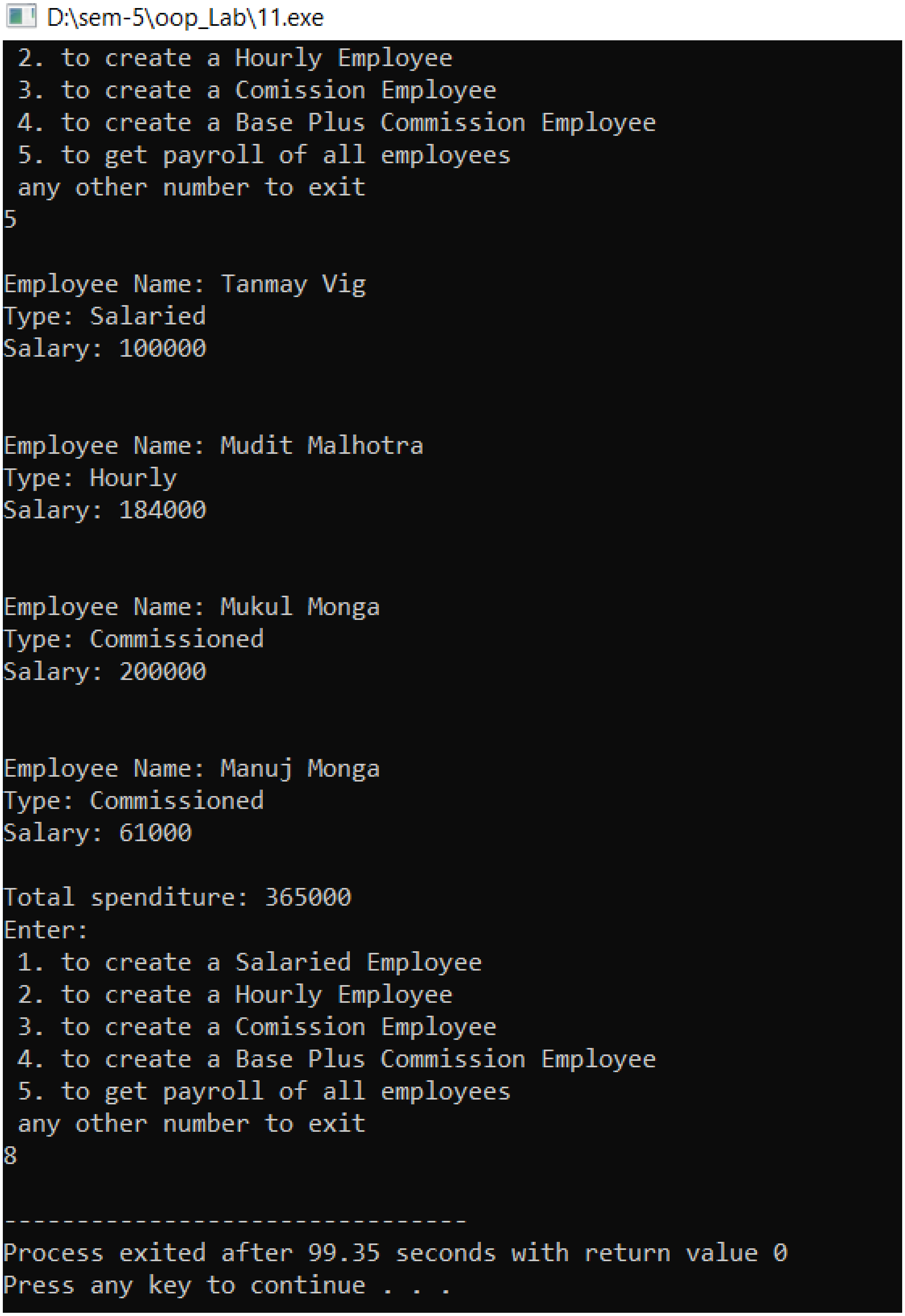
Output:



```
Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
1
Enter name: Tanmay Vig
Enter salary: 100000
Employee Details:

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
2
Enter name: Mudit Malhotra
Enter Hourly salary: 4000
Enter Hours worked: 44
Employee Details:

Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
3
Enter name: Mukul Monga
Enter sales made: 1000000
Enter commission: 20
Employee Details:

Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000
```



```

Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
4
Ente name: Manuj Monga
Enter sales made: 200000
Enter commission: 25
Enter base salary: 10000
Employee Details:

Employee Name: Manuj Monga
Type: Commissioned
Salary: 61000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
5

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000


Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000


Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000


Employee Name: Manuj Monga
Type: Commissioned
```



D:\sem-5\oop_Lab\11.exe

```
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
5

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000


Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000


Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000


Employee Name: Manuj Monga
Type: Commissioned
Salary: 61000

Total spenditure: 365000
Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
8

--------------------------------
Process exited after 99.35 seconds with return value 0
Press any key to continue . . .
```