

## Source Code

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Employee{
```

```
private:
```

```
    string name;
```

```
    string id;
```

```
    int salary;
```

```
public:
```

```
    Employee() {}
```

```
    Employee(string n, string i, int s){
```

```
        id=i;
```

```
        name=n;
```

```
        salary = s;
```

```
    }
```

```
    ~Employee(){
```

```
    }
```

```
    void getEmployee(){
```

```
        cout<<"Employee id: "<<id<<"\nEmployee Name: "<<name<<"\nEmployee salary: "<<salary<<endl;
```

```
    }
```

```
};
```

```
class Scientist: protected Employee{
```

```
private:
```

```
    int num_publications;
```

```
    int num_awards;
```

```
    string *publications;
```

```
    string *awards;
```

public:

```
Scientist() {}
```

```
Scientist(int num_pub, string *pub, string *aw,int n_a, string name, string id, int salary):Employee(name,id,salary){
```

```
    num_publications = num_pub;
```

```
    num_awards = n_a;
```

```
    publications = new string[num_publications];
```

```
    for(int i=0;i<num_publications;i++){
```

```
        publications[i] =pub[i];
```

```
    }
```

```
    awards = new string[n_a];
```

```
    for(int i=0;i<n_a;i++){
```

```
        awards[i] =aw[i];
```

```
    }
```

```
}
```

```
~Scientist(){
```

```
    delete[] publications;
```

```
    delete[] awards;
```

```
}
```

```
void getScientist(){
```

```
    cout<<"\nScientist Details:\n";
```

```
    getEmployee();
```

```
    cout<<"\nTotal publications: "<<num_publications<<endl;
```

```
    cout<<"Publications:\n";
```

```
    for(int i=0;i<num_publications;i++){
```

```
        cout<<" "<<publications[i]<<endl;
```

```
    }
```

```
    cout<<"\nTotal Awards: "<<num_awards<<endl;
```

```
    cout<<"Awards:\n";
```

```
    for(int i=0;i<num_awards;i++){
```

```
        cout<<" "<<awards[i]<<endl;
```

```
    }cout<<"\n";
```

```
}
```

```

};

class Manager : protected Employee{
private:
    string title;
    int yrs_of_exp;
    int teams;
public:
    Manager(){}
    Manager(string t, int yrs, int tm,string name, string id, int salary):Employee(name,id,salary){
        title=t;
        yrs_of_exp=yrs;
        teams = tm;
    }
    ~Manager(){
    }
    void getManager(){
        cout<<"\nManager Details: \n";
        getEmployee();
        cout<<"Title as Manager: "<<title<<endl;
        cout<<"Years of Experience: "<<yrs_of_exp<<endl;
        cout<<"Teams Managed: "<<teams<<endl;
    }
};

```

```

class Laborer : protected Employee{
private:
    int overtime;
    int wage_over;
    int leaves;
public:

```

```

Laborer(){}

Laborer(int o, int w_o, int l,string name, string id, int salary) :Employee(name,id,salary){

    overtime = o;

    wage_over = w_o;

    leaves = l;

}

~Laborer(){

}

void getLaborer(){

    cout<<"\nLabourer Details: \n"<<endl;

    getEmployee();

    cout<<"Overtime: "<<overtime<<endl;

    cout<<"Wage in overtime: "<<wage_over<<endl;

    cout<<"Total leaves: "<<leaves<<endl;

}

};

int main(){

    int sci_num=0,man_num=0, lab_num=0;

    Scientist *arr_s[100];

    Manager *arr_m[100];

    Laborer *arr_l[100];

    while(true){

        int choice=0,c, yrs, teams;

        int salary;

        int n_p,n_a,n_o, wage, leaves;

        string name, id,title;

        cout<<"\nEnter:\n 1 to add Scientist\n 2 to add Manager\n 3 to add Laborer\n 4 to show all Scientists\n 5 to show all Managers\n 6 to show all Laborer\n any num to exit\n";

        cin>>choice;

```

```
switch(choice){  
    case 1:  
    {  
        sci_num++;  
        cout<<"Enter Employee ID, Name and Salary\n";  
        cin.clear();  
        cin.sync();  
        getline(cin, id);  
        getline(cin, name);  
        cin>>salary;  
        cout<<"Enter number of publications\t";  
        cin>>n_p;  
        string *a_p = new string[n_p];  
        cout<<"Enter publications name\n";  
        for(int i=0;i<n_p;i++){  
            string s;  
            cin.clear();  
            cin.sync();  
            getline(cin, s);  
            a_p[i]=s;  
        }  
        cout<<"Enter number of awards\t";  
        cin>>n_a;  
        string *a_a = new string[n_a];  
        cout<<"Enter awards name\n";  
        for(int i=0;i<n_a;i++){  
            string s;  
            cin.clear();  
            cin.sync();  
            getline(cin, s);  
            a_a[i] = s;  
        }  
    }
```

```

arr_s[sci_num-1] = new Scientist(n_p,a_p,a_a, n_a,name,id, salary);

break;
}
case 4:
{
    for(int i=0;i<sci_num;i++){
        (*arr_s[i]).getScientist();
    }
    break;
}
case 2:
{
    man_num++;
    cout<<"Enter Employee ID, Name and Salary\n";
    cin.clear();
    cin.sync();
    getline(cin, id);
    getline(cin, name);
    cin>>salary;
    cout<<"Choose any position:\n 1. General Manager\n 2. Assistant Manager\n 3. Production Manager\n";
    cin>>c;
    // cin.clear();
    // cin.sync();
    switch(c){
        case 1:
            title = "General Manager";
            break;
        case 2:
            title = "Assistant Manager";
            break;
        case 3:
            title="Production Manager";

```

```

        break;
    default:
        title="Unknown Manager";
        break;
    }
    cout<<"Enter number of teams managed\t";
    cin>>teams;
    cout<<"Enter numbers of years of experience\t";
    cin>>yrs;
    arr_m[man_num-1] = new Manager(title,yrs,teams, name,id,salary);
    break;
}
case 5:
{
    for(int i=0;i<man_num;i++){
        (*arr_m[i]).getManager();
    }
    break;
}
case 3:
{
    lab_num++;
    cout<<"Enter Employee ID, Name and Salary\n";
    cin.clear();
    cin.sync();
    getline(cin, id);
    getline(cin, name);
    cin>>salary;
    cout<<"Enter overtime hours\t";
    cin>>n_o;
    cout<<"Enter wage in overtime\t";
    cin>>wage;

```

```

        cout<<"Enter number of leaves\t";

        cin>>leaves;

        arr_l[lab_num-1] = new Laborer(n_o,wage,leaves,name, id, salary);

        break;
    }
    case 6:
    {
        for(int i=0;i<lab_num;i++){
            (*arr_l[i]).getLaborer();
        }

        break;
    }
    default:
    {
        choice = 0;

        break;
    }
}

if(choice == 0) break;
}

return 0;
}

```

Output:

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

1

Enter Employee ID, Name and Salary

100

Mudit Malhotra

20000

Enter number of publications 1

Enter publications name

one

Enter number of awards 1

Enter awards name

alpha

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

1

Enter Employee ID, Name and Salary

101

Karan Kapoor

3000

Enter number of publications 0

Enter publications name

Enter number of awards 1

Enter awards name

theta

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

2

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

2

Enter Employee ID, Name and Salary

102

Tanmay Vig

1000000

Choose any position:

1. General Manager  
2. Assistant Manager  
3. Production Manager

1

Enter number of teams managed 8

Enter numbers of years of experience 5

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

6

Enter:

1 to add Scientist  
2 to add Manager  
3 to add Laborer  
4 to show all Scientists  
5 to show all Managers  
6 to show all Laborer  
any num to exit

3

Enter Employee ID, Name and Salary

103

Paras

10000

Enter overtime hours 6

Enter wage in overtime 100

Enter number of leaves 4

D:\sem-5\oop\_Lab\10.exe

```
10000
Enter overtime hours    6
Enter wage in overtime 100
Enter number of leaves  4

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
any num to exit
2
Enter Employee ID, Name and Salary
104
Manuj Monga
100000
Choose any position:
 1. General Manager
 2. Assistant Manager
 3. Production Manager
2
Enter number of teams managed    5
Enter numbers of years of experience    4

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
any num to exit
4

Scientist Details:
Employee id: 100
Employee Name: Mudit Malhotra
Employee salary: 20000

Total publications: 1
Publications:
one

Total Awards: 1
Awards:
alpha
```

D:\sem-5\oop\_Lab\10.exe

Total Awards: 1

Awards:

alpha

Scientist Details:

Employee id: 101

Employee Name: Karan Kapoor

Employee salary: 3000

Total publications: 0

Publications:

Total Awards: 1

Awards:

theta

Enter:

1 to add Scientist

2 to add Manager

3 to add Laborer

4 to show all Scientists

5 to show all Managers

6 to show all Laborer

any num to exit

5

Manager Details:

Employee id: 102

Employee Name: Tanmay Vig

Employee salary: 1000000

Title as Manager: General Manager

Years of Experience: 5

Teams Managed: 8

Manager Details:

Employee id: 104

Employee Name: Manuj Monga

Employee salary: 100000

Title as Manager: Assistant Manager

Years of Experience: 4

Teams Managed: 5

Enter:

1 to add Scientist

2 to add Manager

3 to add Laborer

4 to show all Scientists

D:\sem-5\oop\_Lab\10.exe

Teams Managed: 8

Manager Details:

Employee id: 104

Employee Name: Manuj Monga

Employee salary: 100000

Title as Manager: Assistant Manager

Years of Experience: 4

Teams Managed: 5

Enter:

1 to add Scientist

2 to add Manager

3 to add Laborer

4 to show all Scientists

5 to show all Managers

6 to show all Laborer

any num to exit

6

Labourer Details:

Employee id: 103

Employee Name: Paras

Employee salary: 10000

Overtime: 6

Wage in overtime: 100

Total leaves: 4

Enter:

1 to add Scientist

2 to add Manager

3 to add Laborer

4 to show all Scientists

5 to show all Managers

6 to show all Laborer

any num to exit

0

-----  
Process exited after 224.7 seconds with return value 0

Press any key to continue . . .