Assignment - 31 **A Job Ready Bootcamp in C++, DSA and IOT** **MySirG**

new and delete operator, Inheritance

1. Define a class Person with instance members name and age. Also define member functions setName(), setAge(), getName(), getAge(). Now define class Employee by inheriting Person class. In the Employee class define empid and salary as instance members. Also define setEmpid, setSalary, getEmpid, getSalary.

```cpp
#include <iostream>
using namespace std;

class person
{
protected:
    string name;
    int age;

public:
    void setName(string name)
    {
        this->name = name;
    }
    void setAge(int age)
    {
        this->age = age;
    }
    string getName()
    {
        return (name);
    }
    int getAge()
    {
        return (age);
    }
};
class employee : public person
{
private:
    int empid;
    double salary;

public:
    void setEmpid(int x)
    {
        empid = x;
    }
    void setSalary(double y)
    {
        salary = y;
    }
    int getEmpid()
    {
        return empid;
    }
    double getSalary()
    {
        return salary;
    }
};
int main()
{
    employee e1;
    e1.setName("Shekh Akhtar");
    e1.setAge(26);
    e1.setEmpid(123);
    e1.setSalary(15356.32);
    cout << "Employee id: " << e1.getEmpid() << endl;
    cout << "Employee Name: " << e1.getName() << endl;
    cout << "Employee age: " << e1.getAge() << endl;
    cout << "Employee salary: " << e1.getSalary() << endl;
    return 0;
}
========================================================================
Output:
Employee id: 123
Employee Name: Shekh Akhtar
Employee age: 26
Employee salary: 15356.3
```

2. Write a C++ program to add two numbers using single inheritance. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.

```
#include <iostream>
using namespace std;

class Number
{
protected:
    int a, b;

public:
    void setData(int x, int y)
    {
        a = x;
        b = y;
    }
};
class Sum : public Number
{
public:
    void displayData()
    {
         cout << "Sum of " << a << " and " << b << " is " << (a + b) <<
endl;
    }
};

int main()
{
    Sum s;
    s.setData(5, 15);
    s.displayData();
    return 0;
}
```

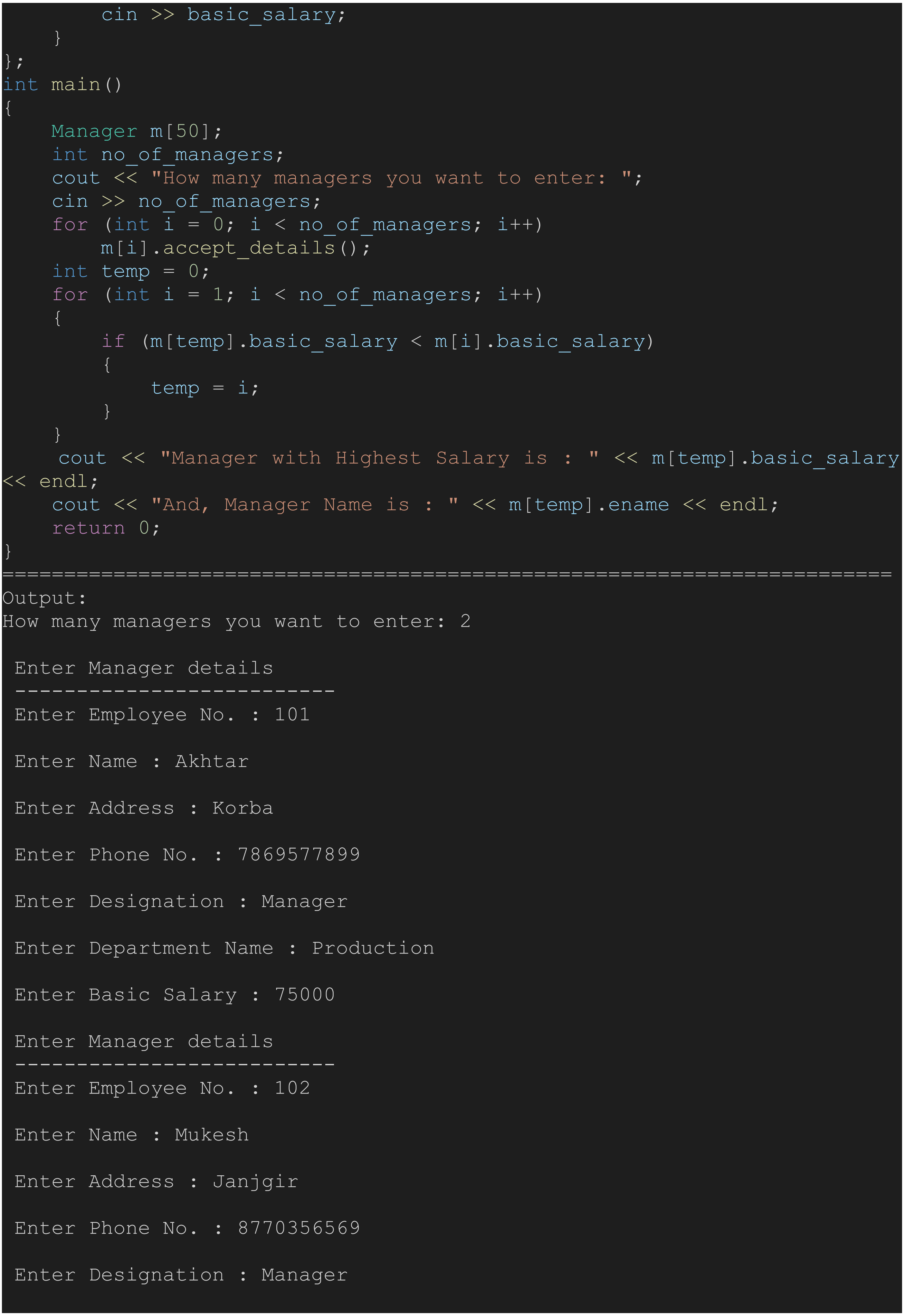
```
========================================================================
Output:
Sum of 5 and 15 is 20
```

3. Write a C++ program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in base class. A class will be derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of students.

```
#include <iostream>
using namespace std;

class A
{
protected:
    int a, b, c;

public:
    void setMarks()
    {
        cout << "Enter marks details" << endl;
        cout << "Physics : ";
        cin >> a;
        cout << "Chemistry : ";
        cin >> b;
        cout << "Mathematics: ";
        cin >> c;
    }
};
class B : public A
{
protected:
    float sum;

public:
    void total()
    {
        sum = (a + b + c);
    }
};
class C : public B
{
private:
    float percentage;

public:
    void displayPercent()
    {
        percentage = (sum/3);
        cout << "Percentage: " << percentage << endl;
    }
};

int main()
{
    C m;
    m.setMarks();
    m.total();
    m.displayPercent();
    return 0;
}
========================================================================
Output:
Enter marks details
Physics : 70
Chemistry : 80
Mathematics: 90
Percentage: 80
```

4. Write a C++ program to design a base class Person (name, address, phone_no). Derive a class Employee (eno, ename) from Person. Derive a class Manager (designation, department name, basic-salary) from Employee. Write a menu driven program to:
a. Accept all details of 'n' managers.
b. Display manager having highest salary

```
#include <iostream>
using namespace std;

class Person
{
protected:
    char name[50], address[100];
    string phone;
};
class Employee : public Person
{
public:
    int eno;
    char ename[50];
};
class Manager : public Employee
{
public:
    char designation[50], deptname[100];
    float basic_salary;
    void accept_details()
    {
        cout << "\n Enter Manager details ";
        cout << "\n -------------------------- ";
        cout << "\n Enter Employee No. : ";
        cin >> eno;
        cout << "\n Enter Name : ";
        cin >> ename;
        cout << "\n Enter Address : ";
        cin >> address;
        cout << "\n Enter Phone No. : ";
        cin >> phone;
        cout << "\n Enter Designation : ";
        cin >> designation;
        cout << "\n Enter Department Name : ";
        cin >> deptname;
        cout << "\n Enter Basic Salary : ";
```



```
        cin >> basic_salary;
    }
};
int main()
{
    Manager m[50];
    int no_of_managers;
    cout << "How many managers you want to enter: ";
    cin >> no_of_managers;
    for (int i = 0; i < no_of_managers; i++)
        m[i].accept_details();
    int temp = 0;
    for (int i = 1; i < no_of_managers; i++)
    {
        if (m[temp].basic_salary < m[i].basic_salary)
        {
            temp = i;
        }
    }
     cout << "Manager with Highest Salary is : " << m[temp].basic_salary
<< endl;
    cout << "And, Manager Name is : " << m[temp].ename << endl;
    return 0;
}
========================================================================
Output:
How many managers you want to enter: 2

 Enter Manager details
 --------------------------
 Enter Employee No. : 101

 Enter Name : Akhtar

 Enter Address : Korba

 Enter Phone No. : 7869577899

 Enter Designation : Manager

 Enter Department Name : Production

 Enter Basic Salary : 75000

 Enter Manager details
 --------------------------
 Enter Employee No. : 102

 Enter Name : Mukesh

 Enter Address : Janjgir

 Enter Phone No. : 8770356569

 Enter Designation : Manager
 Enter Department Name : Animation

 Enter Basic Salary : 85000
 Manager with Highest Salary is : 85000
 And, Manager Name is : Mukesh
```

5. Write a C++ program to define a base class Item (item-no, name, price). Derive a class Discounted-Item (discount-percent). A customer purchases 'n' items. Display the item-wise bill and total amount using appropriate format.

```
#include <iostream>
using namespace std;
class Item
{
    public:
        int item_no;
        float item_price;
        string item_name;
};
class DiscountItem : public Item
{
    public:
        float discount_price;
        float discount_percent;
        void inputData()
        {
            cout<<"\n Enter Item Name : ";
            cin>>item_name;
            cout<<"\n Enter Item No. : ";
            cin>>item_no;
            cout<<"\n Enter Item Price : ";
            cin>>item_price;
            cout<<"\n Enter Discount Percent : ";
            cin>>discount_percent;
            cout<<"\n ----------------------\n";
                        discount_price = item_price - ((item_price *
discount_percent)/100);
        }
        void displayData()
        {
                cout<<"\n Item Name : "<<item_name;
                cout<<"\n Item No. : "<<item_no;
                cout<<"\n Item Price : "<<item_price;
                cout<<"\n Discount Percent : "<<discount_percent;
                cout<<"\n Discounted Price : "<<discount_price;
                cout<<"\n ----------------------\n";
        }
};
int main()
{
    DiscountItem d[100];
     int number_of_item, total_price = 0, total_discount = 0, net_cost =
0;

    cout << "How many items you want to enter? : ";
    cin >> number_of_item;

    for(int i = 0; i < number_of_item; i++)
        d[i].inputData();

    for(int i = 0; i < number_of_item; i++)
        d[i].displayData();

    for(int i = 0; i < number_of_item; i++)
    {
        total_price = total_price + d[i].item_price;
                total_discount = total_discount + (d[i].item_price -
d[i].discount_price);
        net_cost = net_cost + d[i].discount_price;
    }

    cout<<"Total Price : "<<total_price << endl;
    cout<<"Total Discount : "<<total_discount << endl;
    cout<<"Net Cost : "<< net_cost << endl;

    return 0;
}
========================================================================
Output:
How many items you want to enter? : 2

 Enter Item Name : Shirt

 Enter Item No. : 1002

 Enter Item Price : 500

 Enter Discount Percent : 20

 ----------------------

 Enter Item Name : Jeans

 Enter Item No. : 2002

 Enter Item Price : 700

 Enter Discount Percent : 20

 ----------------------

 Item Name : Shirt
 Item No. : 1002
 Item Price : 500
 Discount Percent : 20
 Discounted Price : 400
 ----------------------

 Item Name : Jeans
 Item No. : 2002
 Item Price : 700
 Discount Percent : 20
 Discounted Price : 560
 ----------------------
Total Price : 1200
Total Discount : 240
Net Cost : 960
```

6. Write a C++ program to demonstrate how a common friend function can be used to exchange the private values of two classes. (Use call by reference method).

```cpp
#include <iostream>
using namespace std;
class B;
class A
{
protected:
    int a;

public:
    A() {}
    A(int x)
    {
        a = x;
    }
    void show()
    {
        cout << "A class value " << a << endl;
    }
    friend void swap(A *a, B *b);
};
class B : public A
{
protected:
    int b;

public:
    B() {}
    B(int y)
    {
        b = y;
    }
    void show()
    {
        cout << "B class value " << b << endl;
    }
    friend void swap(A *a, B *b);
};
void swap(A *no1, B *no2)
{
    int no3;
    no3 = no1->a;
    no1->a = no2->b;
    no2->b = no3;
}
int main()
```

```
{
    A a(10);
    B b(20);
    swap(&a, &b);
    a.show();
    b.show();
    return 0;
}
========================================================================
Output:
A class value 20
B class value 10
```

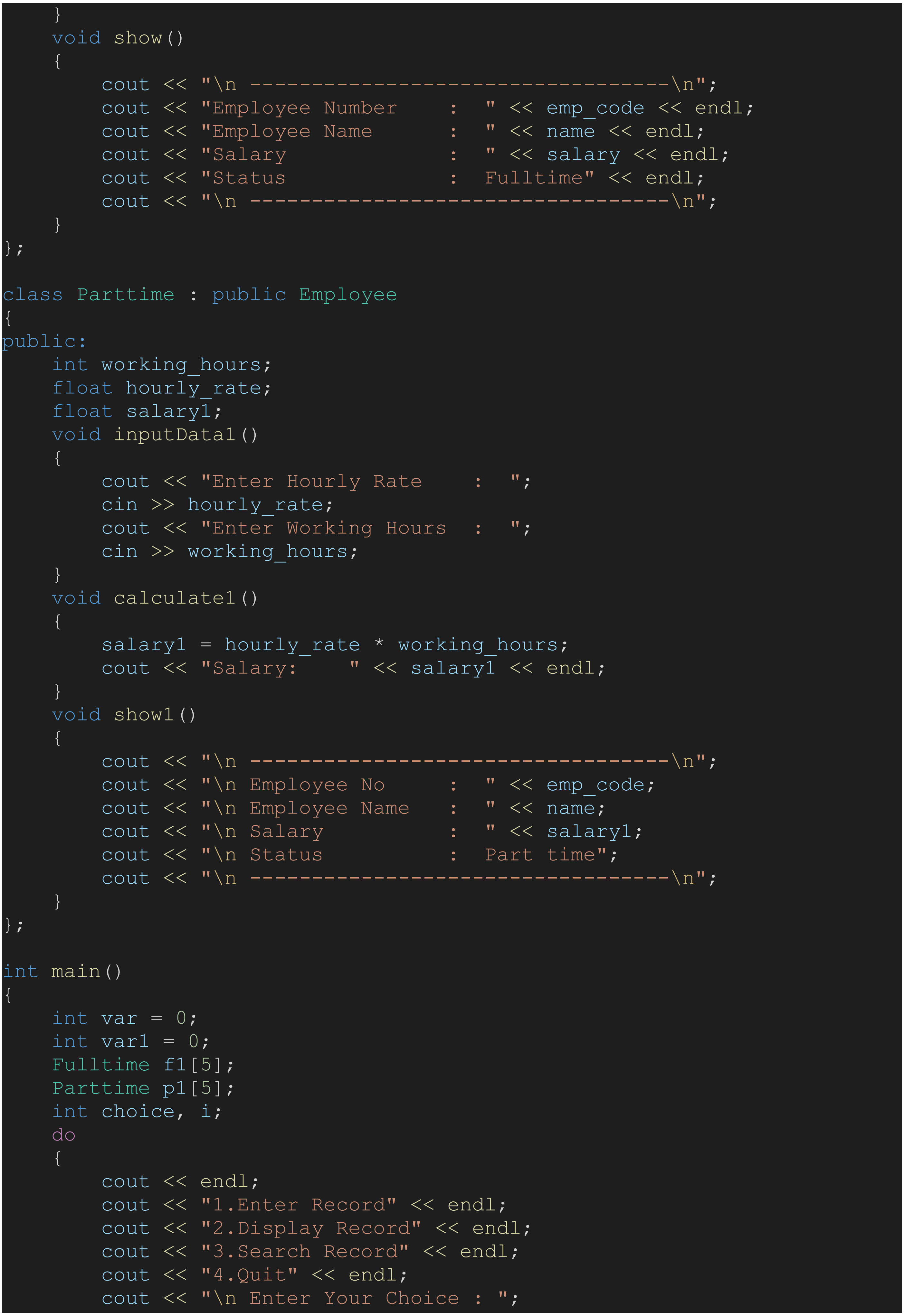
7. Write class declarations and member function definitions for a C++ base class to represent an Employee (emp-code, name).

Derive two classes as Fulltime (daily rate, number of days, salary) and Parttime (number of working hours, hourly rate, salary).

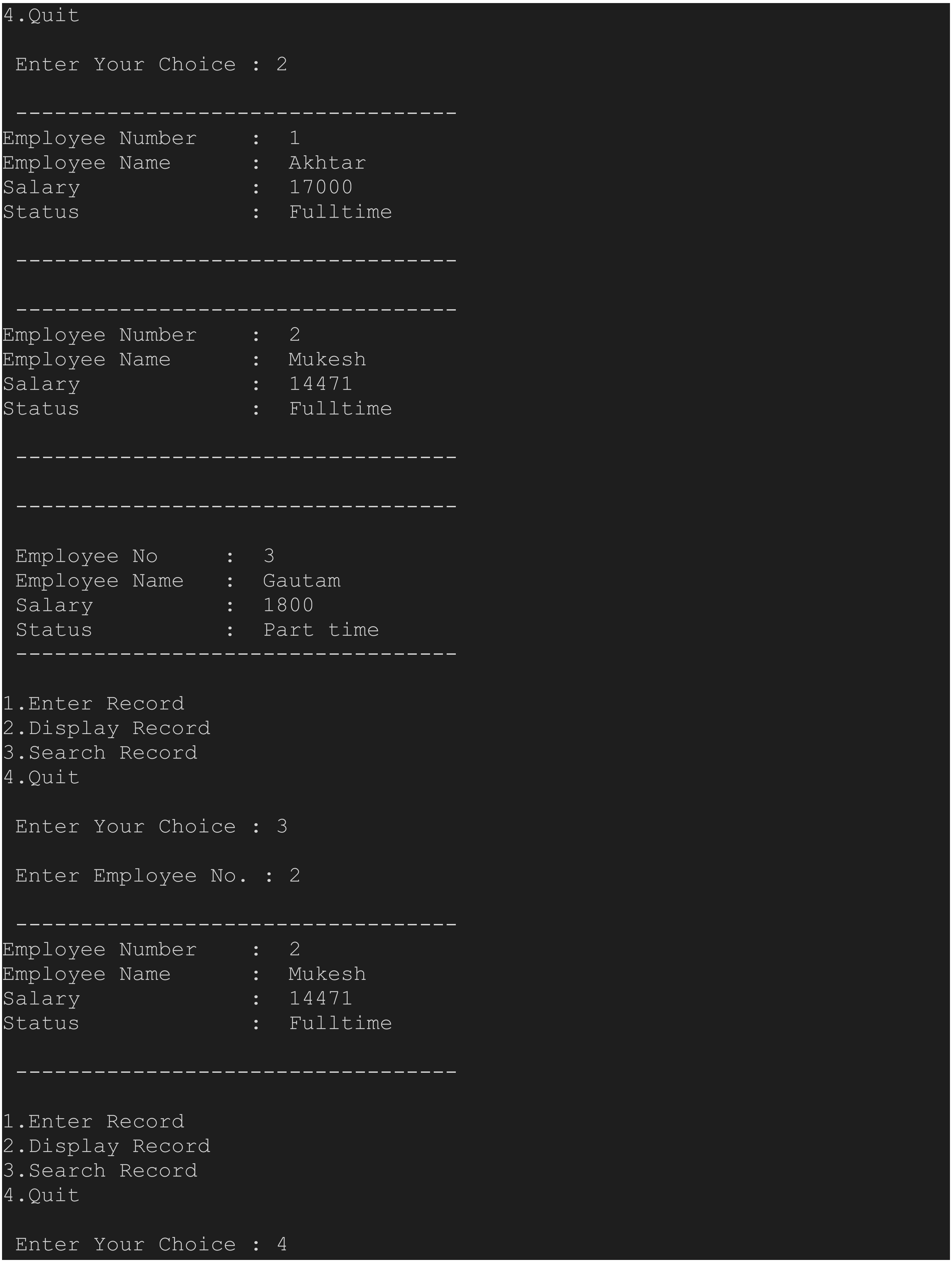
Write a menu driven program to:

1. Accept the details of ‘n’ employees.
2. Display the details of ‘n’ employees.
3. Search a given Employee by emp-code.

```
#include <iostream>
using namespace std;

class Employee
{
public:
    int emp_code;
    string name;
    void input()
    {
        cout << "Enter Employee No.   :  ";
        cin >> emp_code;
        cout << "Enter Employee Name  :  ";
        cin >> name;
    }
};

class Fulltime : public Employee
{
public:
    float daily_rate;
    int number_of_days;
    int salary;
    void inputData()
    {
        cout << "Enter Daily Rate     :  ";
        cin >> daily_rate;
        cout << "Enter No. of Days    :  ";
        cin >> number_of_days;
    }
    void calculate()
    {
        salary = daily_rate * number_of_days;
        cout << "Salary:    " << salary << endl;
```



```cpp
    }
    void show()
    {
        cout << "\n ----------------------------------\n";
        cout << "Employee Number    :  " << emp_code << endl;
        cout << "Employee Name      :  " << name << endl;
        cout << "Salary             :  " << salary << endl;
        cout << "Status             :  Fulltime" << endl;
        cout << "\n ----------------------------------\n";
    }
};

class Parttime : public Employee
{
public:
    int working_hours;
    float hourly_rate;
    float salary1;
    void inputData1()
    {
        cout << "Enter Hourly Rate    :  ";
        cin >> hourly_rate;
        cout << "Enter Working Hours  :  ";
        cin >> working_hours;
    }
    void calculate1()
    {
        salary1 = hourly_rate * working_hours;
        cout << "Salary:    " << salary1 << endl;
    }
    void show1()
    {
        cout << "\n ----------------------------------\n";
        cout << "\n Employee No     :  " << emp_code;
        cout << "\n Employee Name   :  " << name;
        cout << "\n Salary          :  " << salary1;
        cout << "\n Status          :  Part time";
        cout << "\n ----------------------------------\n";
    }
};

int main()
{
    int var = 0;
    int var1 = 0;
    Fulltime f1[5];
    Parttime p1[5];
    int choice, i;
    do
    {
        cout << endl;
        cout << "1.Enter Record" << endl;
        cout << "2.Display Record" << endl;
        cout << "3.Search Record" << endl;
        cout << "4.Quit" << endl;
        cout << "\n Enter Your Choice : ";
        cin >> choice;
        switch (choice)
        {
        case 1:
            int y;
            cout << "\n 1. Fulltime Employee";
            cout << "\n 2. Parttime Employee \n";
            cout << "\n Enter : ";
            cin >> y;
            switch (y)
            {
            case 1:
                f1[var].input();
                f1[var].inputData();
                f1[var].calculate();
                var++;
                break;
            case 2:
                p1[var1].input();
                p1[var1].inputData1();
                p1[var1].calculate1();
                var1++;
                break;
            }
            break;
        case 2:
            for (i = 0; i < var; i++)
            {
                f1[i].show();
            }
            for (i = 0; i < var1; i++)
            {
                p1[i].show1();
            }
            break;
        case 3:
            int a;
            cout << "\n Enter Employee No. : ";
            cin >> a;
            for (int i = 0; i < var; i++)
            {
                if (f1[i].emp_code == a)
                {
                    f1[i].show();
                }
                if (p1[i].emp_code == a)
                {
                    p1[i].show1();
                }
            }
        }
    } while (choice != 4);

    return 0;
}
========================================================================
Output:

1.Enter Record
2.Display Record
3.Search Record
4.Quit

 Enter Your Choice : 1

 1. Fulltime Employee
 2. Parttime Employee

 Enter : 1
Enter Employee No.   :  1
Enter Employee Name  :  Akhtar
Enter Daily Rate     :  566.67
Enter No. of Days    :  30
Salary:    17000

1.Enter Record
2.Display Record
3.Search Record
4.Quit

 Enter Your Choice : 1

 1. Fulltime Employee
 2. Parttime Employee

 Enter : 1
Enter Employee No.   :  2
Enter Employee Name  :  Mukesh
Enter Daily Rate     :  499
Enter No. of Days    :  29
Salary:    14471

1.Enter Record
2.Display Record
3.Search Record
4.Quit

 Enter Your Choice : 1

 1. Fulltime Employee
 2. Parttime Employee

 Enter : 2
Enter Employee No.   :  3
Enter Employee Name  :  Gautam
Enter Hourly Rate    :  100
Enter Working Hours  :  18
Salary:    1800

1.Enter Record
2.Display Record
3.Search Record
```



```
4.Quit

 Enter Your Choice : 2

 ----------------------------------
Employee Number    :  1
Employee Name      :  Akhtar
Salary             :  17000
Status             :  Fulltime

 ----------------------------------

 ----------------------------------
Employee Number    :  2
Employee Name      :  Mukesh
Salary             :  14471
Status             :  Fulltime

 ----------------------------------

 ----------------------------------

 Employee No     :  3
 Employee Name   :  Gautam
 Salary          :  1800
 Status          :  Part time
 ----------------------------------

1.Enter Record
2.Display Record
3.Search Record
4.Quit

 Enter Your Choice : 3

 Enter Employee No. : 2

 ----------------------------------
Employee Number    :  2
Employee Name      :  Mukesh
Salary             :  14471
Status             :  Fulltime

 ----------------------------------

1.Enter Record
2.Display Record
3.Search Record
4.Quit

 Enter Your Choice : 4
```

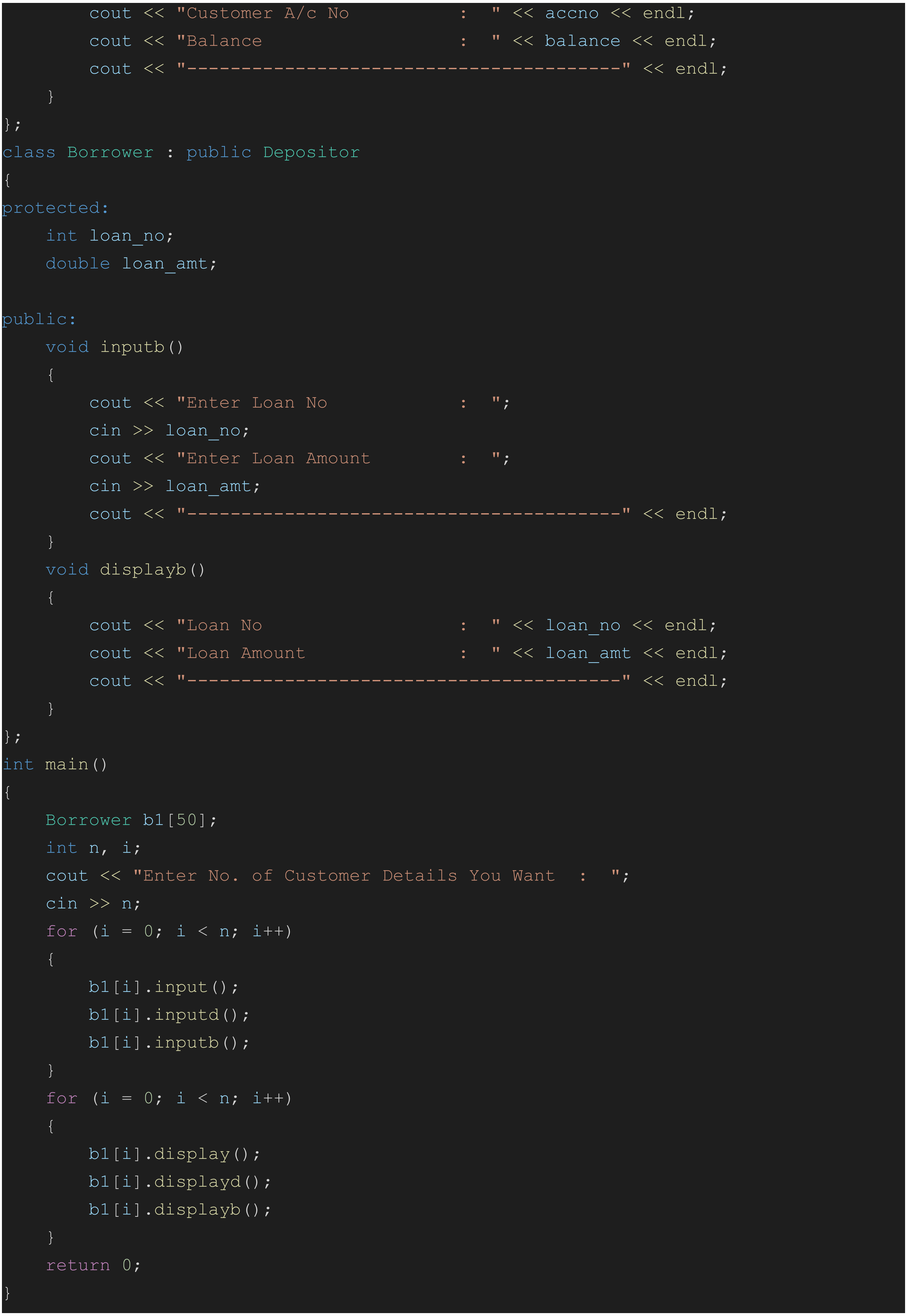
8 - In a bank, different customers have savings account. Some customers may have taken a loan from the bank. So bank always maintain information about bank depositors and borrowers.

Design a Base class Customer (name, phone-number). Derive a class Depositor(accno, balance) from Customer.

Again, derive a class Borrower (loan-no, loan-amt) from Depositor.

Write necessary member functions to read and display the details of ‘n’
customers.

```
#include <iostream>
using namespace std;
class Customer
{
protected:
    string name, phone_number;

public:
    void input()
    {
        cout << "Enter Customer Name      :  ";
        cin >> name;
        cout << "Enter Customer Phone No. :  ";
        cin >> phone_number;
    }
    void display()
    {
        cout << "Details of Customer " << endl;
        cout << "----------------------------------------" << endl;
        cout << "Customer Name            :  " << name << endl;
        cout << "Customer Phone No.       :  " << phone_number << endl;
    }
};
class Depositor : public Customer
{
protected:
    string accno;
    float balance;

public:
    void inputd()
    {
        cout << "Enter Customer A/c No    :  ";
        cin >> accno;
        cout << "Enter Balance            :  ";
        cin >> balance;
    }
    void displayd()
    {
```



```
        cout << "Customer A/c No          :  " << accno << endl;
        cout << "Balance                  :  " << balance << endl;
        cout << "----------------------------------------" << endl;
    }
};
class Borrower : public Depositor
{
protected:
    int loan_no;
    double loan_amt;

public:
    void inputb()
    {
        cout << "Enter Loan No            :  ";
        cin >> loan_no;
        cout << "Enter Loan Amount        :  ";
        cin >> loan_amt;
        cout << "----------------------------------------" << endl;
    }
    void displayb()
    {
        cout << "Loan No                  :  " << loan_no << endl;
        cout << "Loan Amount              :  " << loan_amt << endl;
        cout << "----------------------------------------" << endl;
    }
};
int main()
{
    Borrower b1[50];
    int n, i;
    cout << "Enter No. of Customer Details You Want  :  ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        b1[i].input();
        b1[i].inputd();
        b1[i].inputb();
    }
    for (i = 0; i < n; i++)
    {
        b1[i].display();
        b1[i].displayd();
        b1[i].displayb();
    }
    return 0;
}
```

```
====================================================================================
Output:
Enter No. of Customer Details You Want  :  2
Enter Customer Name      :  Akhtar
Enter Customer Phone No. :  7869577899
Enter Customer A/c No    :  60178835032
Enter Balance            :  24556
Enter Loan No            :  124
Enter Loan Amount        :  12400
----------------------------------------
Enter Customer Name      :  Mukesh
Enter Customer Phone No. :  568974215
Enter Customer A/c No    :  5321479
Enter Balance            :  24689
Enter Loan No            :  126
Enter Loan Amount        :  2560
----------------------------------------
Details of Customer
----------------------------------------
Customer Name            :  Akhtar
Customer Phone No.       :  7869577899
Customer A/c No          :  60178835032
Balance                  :  24556
----------------------------------------
Loan No                  :  124
Loan Amount              :  12400
----------------------------------------
Details of Customer
----------------------------------------
Customer Name            :  Mukesh
Customer Phone No.       :  568974215
Customer A/c No          :  5321479
Balance                  :  24689
----------------------------------------
Loan No                  :  126
Loan Amount              :  2560
----------------------------------------
```

9. Write a C++ program to implement the following class hierarchy:
Student: id, name
StudentExam (derived from Student): Marks of 6 subjects
StudentResult (derived from StudentExam) : percentage
Define appropriate functions to accept and display details.
Create 'n' objects of the StudentResult class and display the marklist.

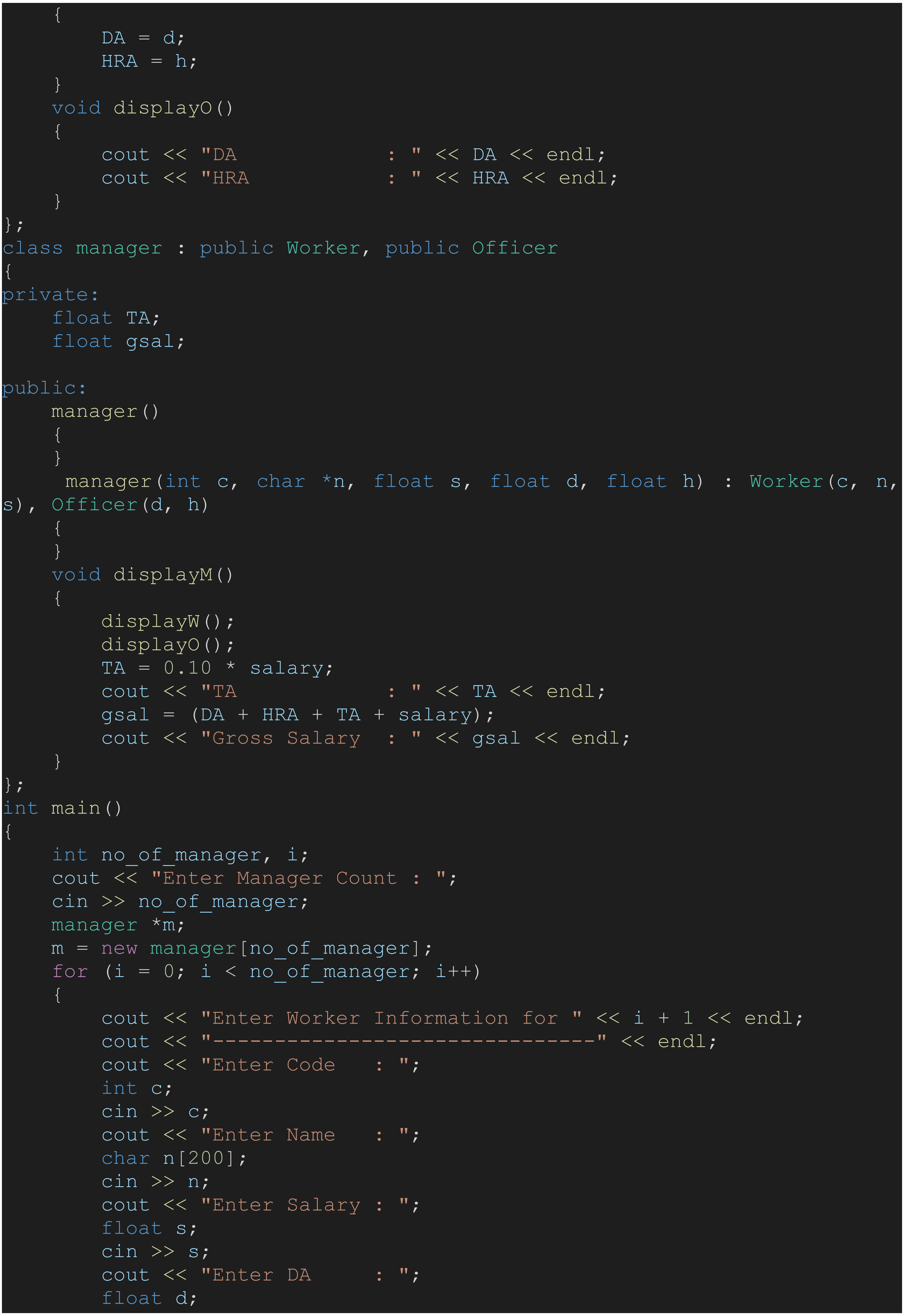
```
#include <iostream>
using namespace std;

class Student
```

```
{
protected:
    string id, name;

public:
    void inputData()
    {
        cout << "-----------------------------------------" << endl;
        cout << "Enter Roll No.            : ";
        cin >> id;
        cout << "Enter Student Name        : ";
        cin >> name;
    }
    void displayData()
    {
        cout << "-----------------------------------------" << endl;
        cout << "********** Student Marklist **********" << endl;
        cout << "-----------------------------------------" << endl;
        cout << "Roll No.             :  " << id << endl;
        cout << "Student Name         :  " << name << endl;
    }
};

class StudentExam : public Student
{
public:
    int sub1, sub2, sub3, sub4, sub5, sub6;
    float per;

    void accept_data()
    {
        inputData();
        cout << "Enter Marks for Subject 1 : ";
        cin >> sub1;
        cout << "Enter Marks for Subject 2 : ";
        cin >> sub2;
        cout << "Enter Marks for Subject 3 : ";
        cin >> sub3;
        cout << "Enter Marks for Subject 4 : ";
        cin >> sub4;
        cout << "Enter Marks for Subject 5 : ";
        cin >> sub5;
        cout << "Enter Marks for Subject 6 : ";
        cin >> sub6;
    }
    void display_data()
    {
        displayData();
        cout << "Marks of Subject 1   :  " << sub1 << endl;
        cout << "Marks of Subject 2   :  " << sub2 << endl;
        cout << "Marks of Subject 3   :  " << sub3 << endl;
        cout << "Marks of Subject 4   :  " << sub4 << endl;
        cout << "Marks of Subject 5   :  " << sub5 << endl;
        cout << "Marks of Subject 6   :  " << sub6 << endl;
    }
};

class StudentResult : public StudentExam
{
public:
    void calculate()
    {
        per = ((sub1 + sub2 + sub3 + sub4 + sub5 + sub6) / 6.0);
        cout << "Total Percentage     :  " << per << endl;
        cout << "-----------------------------------------" << endl;
    }
};

int main()
{
    StudentResult st;
    int number_of_students, i;
    cout << "Enter No. of Students You Want? : ";
    cin >> number_of_students;
    for (i = 0; i < number_of_students; i++)
    {
        st.accept_data();
        st.display_data();
        st.calculate();
    }
    return 0;
}
===================================================================================
Output:
Enter No. of Students You Want? : 1
-----------------------------------------
Enter Roll No.            : 123
Enter Student Name        : Akhtar
Enter Marks for Subject 1 : 65
Enter Marks for Subject 2 : 45
Enter Marks for Subject 3 : 75
Enter Marks for Subject 4 : 95
Enter Marks for Subject 5 : 40
Enter Marks for Subject 6 : 60
```

```
------------------------------------
********** Student Marklist **********
------------------------------------
Roll No.              :  123
Student Name          :  Akhtar
Marks of Subject 1    :  65
Marks of Subject 2    :  45
Marks of Subject 3    :  75
Marks of Subject 4    :  95
Marks of Subject 5    :  40
Marks of Subject 6    :  60
Total Percentage      :  63.3333
------------------------------------
```

10. Consider two base classes
worker(int code, char name, float salary),
officer(float DA, HRA)
class manger(float TA(is 10% of salary), gross salary) is derived from both base classes.
Write necessary member functions.

```
#include <iostream>
#include <string.h>
using namespace std;

class Worker
{
protected:
    int code;
    char name[50];
    float salary;

public:
    Worker() {}
    Worker(int c, char *n, float s)
    {
        code = c;
        strcpy(name, n);
        salary = s;
    }
    void displayW()
    {
        cout << "Code          : " << code << endl;
        cout << "Name          : " << name << endl;
        cout << "Salary        : " << salary << endl;
    }
};
class Officer
{
protected:
    float DA, HRA;

public:
    Officer() {}
    Officer(float d, float h)
```



```
    {
        DA = d;
        HRA = h;
    }
    void displayO()
    {
        cout << "DA            : " << DA << endl;
        cout << "HRA           : " << HRA << endl;
    }
};
class manager : public Worker, public Officer
{
private:
    float TA;
    float gsal;

public:
    manager()
    {
    }
     manager(int c, char *n, float s, float d, float h) : Worker(c, n,
s), Officer(d, h)
    {
    }
    void displayM()
    {
        displayW();
        displayO();
        TA = 0.10 * salary;
        cout << "TA            : " << TA << endl;
        gsal = (DA + HRA + TA + salary);
        cout << "Gross Salary  : " << gsal << endl;
    }
};
int main()
{
    int no_of_manager, i;
    cout << "Enter Manager Count : ";
    cin >> no_of_manager;
    manager *m;
    m = new manager[no_of_manager];
    for (i = 0; i < no_of_manager; i++)
    {
        cout << "Enter Worker Information for " << i + 1 << endl;
        cout << "-------------------------------" << endl;
        cout << "Enter Code   : ";
        int c;
        cin >> c;
        cout << "Enter Name   : ";
        char n[200];
        cin >> n;
        cout << "Enter Salary : ";
        float s;
        cin >> s;
        cout << "Enter DA     : ";
        float d;
        cin >> d;
        cout << "Enter HRA     : ";
        float h;
        cin >> h;
        m[i] = manager(c, n, s, d, h);
    }
    for (i = 0; i < no_of_manager; i++)
    {
        cout << "-------------------------------" << endl;
        cout << "Manager Information " << endl;
        cout << "-------------------------------" << endl;
        m[i].displayM();
    }
    return 0;
}
========================================================================
Output:
Enter Manager Count : 1
Enter Worker Information for 1
-------------------------------
Enter Code    : 123
Enter Name    : Shekh
Enter Salary : 15000
Enter DA      : 500
Enter HRA     : 400
-------------------------------
Manager Information
-------------------------------
Code              : 123
Name              : Shekh
Salary            : 15000
DA                : 500
HRA               : 400
TA                : 1500
Gross Salary   : 17400
```