**Assignment-31**

**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.

#include<iostream>

using namespace std;

class *Person*{

*string* name;

    int age;

    public:

        void setName(*string* *n*){

            name = *n*;

        }

        void setAge(int *a*){

            age = *a*;

        }

*string* getName(){

            return *this*->name;

        }

        int getAge(){

            return *this*->age;

        }

};

class *Employee*:public *Person*{

    int empId,salary;

    public:

        void setEmpId(int *id*){

*this*->empId = *id*;

        }

        void setSalary(int *salary*){

*this*->salary = *salary*;

        }

        int getEmpId(){

            return empId;

        }

        int getSalary(){

            return salary;

        }

};

int main(){

*Employee* e1;

    e1.setName("Ajay Singh");

    e1.setAge(25);

    e1.setEmpId(1);

    e1.setSalary(5000);

    cout<<"Employee Name: "<<e1.getName()<<endl;

    cout<<"Employee ID: "<<e1.getEmpId()<<endl;

    cout<<"Employee Age: "<<e1.getAge()<<endl;

    cout<<"Employee Salary: "<<e1.getSalary()<<"/- Rupees only "<<endl;

    return 0;

}

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.

// 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 *Base*{

    public:

    int num1,num2;

        void setNum(){

            cout<<"Enter Two Numbers To Add: "<<endl;

            cin>>num1>>num2;

        }

};

class *Derived*: public *Base*{

    int sum;

    public:

        void Sum(){

            cout<<"sum: "<<num1+num2;

        }

};

int main(){

*Derived* a;

    a.setNum();

    a.Sum();

    return 0;

}

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.

// 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 *Marks*{

    public:

        float m1,m2,m3;

        void setMarks(float *x*=0,float *y*=0, float *z*=0){

            m1 = *x*;

            m2 = *y*;

            m3 = *z*;

        }

};

class *Total*: public *Marks*{

    public:

        float Total = 0;

        void calculateTotal(){

            Total = m1 + m2 + m3;

        }

};

class *Percentage*: public *Total*{

    public:

        float percentage = 0;

        void display(){

            percentage = (Total/300) \* 100;

            cout<<percentage<<"% "<<endl;

        }

};

int main(){

    int m1,m2,m3;

    cout<<"Enter marks of three subjects: "<<endl;

    cin>>m1>>m2>>m3;

*Percentage* p;

    //To Set the marks of student

    p.setMarks(m1,m2,m3);

    //Calculates total marks

    p.calculateTotal();

    //Displays the percentage

    p.display();

    return 0;

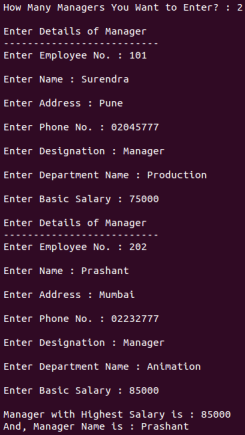
}

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

**Output -**

****

// 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

// image.png

#include<iostream>

using namespace std;

class *Person*{

    protected:

*string* name;

*string* address;

    long long int phone\_no;

    public:

        void setName(*string* *s*){

            name = *s*;

        }

        void setAddress(*string* *t*){

            address = *t*;

        }

        void setPhone(long long int *x*){

            phone\_no = *x*;

        }

*string* getName(){

            return *this*->name;

        }

*string* getAddress(){

            return *this*->address;

        }

        long long int getPhone(){

            return phone\_no;

        }

};

class *Employee*:public *Person*{

    protected:

     int empNo;

*string* ename;

    public:

      void setEmpNo(int *no*){

        empNo = *no*;

      }

      void setEmpName(*string* *name*){

        ename = *name*;

      }

      int getEmpNo(){

        return empNo;

      }

*string* getEmpName(){

        return ename;

      }

};

class *Manager*: public *Employee*{

    protected:

*string* designation, deptName;

    int salary;

    public:

        void setDesignation(*string* *s*){

            designation = *s*;

        }

        void setDepartment(*string* *a*){

            deptName = *a*;

        }

        void setSalary(int *s*){

            salary = *s*;

        }

        int getSalary(){

            return *this*->salary;

        }

*string* getDesignation(){

            return designation;

        }

*string* getDepartment(){

            return deptName;

        }

};

int main(){

    int n;

    cout<<"How many managers you want to enter: ";

    cin>>n;

*string* temp;

    int temp\_var;

    long long int x;

*Manager* \*m;

    try{

        m = **new** *Manager* [n];

    }catch(*bad\_alloc* a){

        cout<<"Allocation Failed"<<endl;

    }

    for(int i=0; i<n; i++){

        cout<<endl<<"Enter Details of Manager "<<i+1<<endl<<"-------------------------"<<endl;

        cout<<"Enter employee number : ";

        cin>>temp\_var;

        (m+i)->setEmpNo(temp\_var);

        cout<<"Enter Name : ";

        fflush(stdin);

        getline(cin,temp);

        (m+i)->setEmpName(temp);

        cout<<"Enter Address : ";

        fflush(stdin);

        getline(cin,temp);

        (m+i)->setAddress(temp);

        cout<<"Enter Phone no : ";

        cin>>x;

        (m+i)->setPhone(x);

        cout<<"Enter Designation : ";

        fflush(stdin);

        getline(cin,temp);

        (m+i)->setDesignation(temp);

        cout<<"Enter Department Name : ";

        fflush(stdin);

        getline(cin,temp);

        (m+i)->setDepartment(temp);

        cout<<"Enter Basic Salary : ";

        cin>>temp\_var;

        (m+i)->setSalary(temp\_var);

        cout<<endl;

    }

    for(int i=0; i<n; i++){

        cout<<"----------------------"<<endl;

        cout<<"Employee No.  : " <<(m+i)->getEmpNo()<<endl;

        cout<<"Name  : " <<(m+i)->getEmpName()<<endl;

        cout<<"Address  : " <<(m+i)->getAddress()<<endl;

        cout<<"Phone No.  : " <<(m+i)->getPhone()<<endl;

        cout<<"Designation  : " <<(m+i)->getDesignation()<<endl;

        cout<<"Department Name  : " <<(m+i)->getDepartment()<<endl;

        cout<<"Base Salary  : " <<(m+i)->getSalary()<<endl;

        cout<<endl;

    }

    int max\_sal = 0;

*string* max\_sal\_manager;

    for(int i=0; i<n; i++){

        if((m+i)->getSalary() > max\_sal){

            max\_sal = (m+i)->getSalary();

            max\_sal\_manager = (m+i)->getEmpName();

        }

    }

        cout<<"Maximum Salaried Manager is: " <<max\_sal\_manager<<" With Salary: "<<max\_sal<<endl;

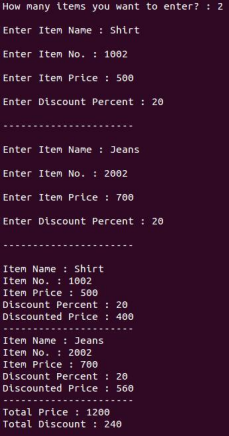
        cout<<endl;

    return 0;

}

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.

**Output -**

****

// 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.

// Output -

#include<iostream>

using namespace std;

class *Item*{

    public:

*string* name;

    int item\_no;

    float price;

};

class *Discounted\_Item*: public *Item*{

    public:

    int discount\_percent;

    float discounted\_Price(){

        float temp = (price \* discount\_percent) / 100;

        float ePrice = price - temp;

        return ePrice;

    }

};

int main(){

    int n;

    cout<<"How many Items u wanted to enter: ";

    cin>>n;

*Discounted\_Item* \* items;

    try{

        items = **new** *Discounted\_Item*[n];

    }catch(*bad\_alloc* e){

        cout<<"Allocation Failed";

    }

    for(int i = 0; i<n; i++){

        cout<<"Enter Item Name : ";

        fflush(stdin);

        getline(cin,(items+i)->name);

        cout<<"Enter Item No. : ";

        cin>>(items+i)->item\_no;

        cout<<"Enter Item Price : ";

        cin>>(items+i)->price;

        cout<<"Enter Discount Percent : ";

        cin>>(items+i)->discount\_percent;

        cout<<endl<<"----------------"<<endl;

    }

    for(int i=0;i<n;i++){

        cout<<endl<<"Item Name: "<<(items+i)->name;

        cout<<endl<<"Item No: "<<(items+i)->item\_no;

        cout<<endl<<"Item Price: "<<(items+i)->price;

        cout<<endl<<"Discount Percent: "<<(items+i)->discount\_percent;

        cout<<endl<<"Discounted Price: "<<(items+i)->discounted\_Price()<<endl;

    }

    cout<<endl<<"--------------------"<<endl;

    float total\_Price  = 0;

    float total\_Discount  = 0;

    for(int i=0; i<n; i++){

        total\_Price += (items+i)->price;

        total\_Discount += (items+i)->discounted\_Price();

    }

    cout<<"Total Price : "<<total\_Price<<endl;

    cout<<"Total Discount : "<<total\_Discount<<endl;

    return 0;

}

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).

// 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).

#include<iostream>

using namespace std;

class *B*;

class *A*{

    private:

    int num1 = 10;

    public:

        friend void swap(*A* &*x*, *B* &*y*);

        void show(){

            cout<<"A : num1 = "<<num1<<endl;

        }

};

class *B*{

    private:

    int num2 = 20;

    public:

        friend void swap(*A* &*x*, *B* &*y*);

        void show(){

            cout<<"B : num2 = "<<num2<<endl;

        }

};

void swap(*A* &*x*, *B* &*y*){

    int temp = *x*.num1;

*x*.num1 = *y*.num2;

*y*.num2 = temp;

}

int main(){

*A* a;

*B* b;

    a.show();

    b.show();

    cout<<endl;

    swap(a,b);

    a.show();

    b.show();

    return 0;

}

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).

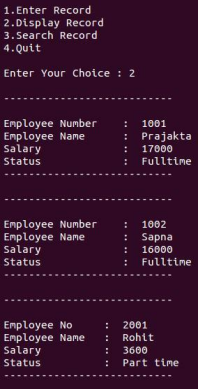
Write a menu driven program to:

1. Accept the details of ‘n’ employees.

2. Display the details of ‘n’ employees.

3. Search a given Employee by emp-code.

**Output -**

****

// 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.

// Output -

#include <iostream>

using namespace std;

class *Employee*

{

protected:

    int emp\_code;

*string* name;

public:

    void setEmpDetails()

    {

        cout << "Enter Employee Number: ";

        cin >> *this*->emp\_code;

        cout << "Enter Employee name: ";

        fflush(stdin);

        getline(cin, *this*->name);

    }

};

class *Fulltime* : public *Employee*

{

    int daily\_rate, no\_of\_days, salary;

public:

    void setFullDetails()

    {

        cout << "Enter Daily Rate: ";

        cin >> *this*->daily\_rate;

        cout << "Enter No. of days: ";

        cin >> *this*->no\_of\_days;

    }

    int getSalary()

    {

*this*->salary = daily\_rate \* no\_of\_days;

        return *this*->salary;

    }

    void showDetails()

    {

        cout << "\n------------------------\n";

        cout << "Employee No.  : " << emp\_code << endl;

        cout << "Employee Name : " << name << endl;

        cout << "Salary        : " << *this*->getSalary() << endl;

        cout << "Status        : Full-Time" << endl;

    }

    int getID(){

        return emp\_code;

    }

};

class *Parttime* : public *Employee*

{

    int no\_of\_working\_Hours, hourly\_rate, salary;

public:

    void setPartDetails()

    {

        cout << "Enter Working Hours: ";

        cin >> *this*->no\_of\_working\_Hours;

        cout << "Enter Hourly Rate: ";

        cin >> *this*->hourly\_rate;

    }

    int getSalary()

    {

*this*->salary = no\_of\_working\_Hours \* hourly\_rate;

        return *this*->salary;

    }

    void showDetails()

    {

        cout << "\n------------------------\n";

        cout << "Employee No.  : " << emp\_code << endl;

        cout << "Employee Name : " << name << endl;

        cout << "Salary        : " << *this*->getSalary() << endl;

        cout << "Status        : Part-Time" << endl;

    }

    int getID(){

        return emp\_code;

    }

};

int main()

{

    int ch;

    int fval = 0; // to access employee from fulltime array

    int pval = 0; // to access employee from parttimr array

*Fulltime* f[10];

*Parttime* p[10];

    bool flag = true;

    while (flag)

    {

        cout << "1. Enter Record" << endl;

        cout << "2. Display Record" << endl;

        cout << "3. Search Record" << endl;

        cout << "4. Quit" << endl;

        cout << "Enter Your Choice " << endl;

        cin >> ch;

        switch (ch)

        {

        case 1:

            int empNo;

            cout << "How many employees you want to enter: ";

            cin >> empNo;

            while (empNo > 0)

            {

                int choice;

                cout<<endl;

                cout << "\t1. Full-Time Employee" << endl;

                cout << "\t2. Part-Time Employee " << endl;

                cout << "Choose an Option : ";

                cin >> choice;

                if (choice == 1)

                {

                    f[fval].setEmpDetails();

                    f[fval].setFullDetails();

                    f[fval].getSalary();

                    fval++;

                }

                else if (choice == 2)

                {

                    p[pval].setEmpDetails();

                    p[pval].setPartDetails();

                    p[pval].getSalary();

                    pval++;

                }

                else

                {

                    cout << "\tPlease enter Valid Option" << endl;

                }

                empNo--;

            }

            break;

        case 2: // Display Record

            cout<<endl;

            cout << "\t\tFull-Time Employees: " << endl;

            for (int i = 0; i < fval; i++)

            {

                f[i].showDetails();

            }

            cout << endl;

            cout << "\t\tPart-Time Employees: " << endl;

            for (int i = 0; i < pval; i++)

            {

                p[i].showDetails();

            }

            break;

        case 3: // Search Record

                int id;

                int choice2;

                cout<<"Whom do u wanna Search? "<<endl;

                cout << "\t1. Full-Time Employee" << endl;

                cout << "\t2. Part-Time Employee " << endl;

                cout << "Choose an Option : ";

                cin >> choice2;

                switch (choice2)

                {

                case 1:

                    cout<<"Enter Employee ID : ";

                    cin>>id;

                    for(int j=0; j<fval;j++){

                        if(f[j].getID() == id){

                            f[j].showDetails();

                            break;

                        }

                        else{

                            continue;

                        }

                    }

                    cout<<"FullTime Employee Not Found"<<endl;

                    break;

                case 2:

                    cout<<"Enter Employee ID : ";

                    cin>>id;

                    for(int j=0;j<pval;j++){

                        if(p[j].getID() == id){

                            p[j].showDetails();

                            break;

                        }

                        else{

                            continue;

                        }

                    }

                    cout<<"PartTime Employee Not Found"<<endl;

                    break;

                default:

                    cout<<"Invalid Choice.."<<endl;

                    break;

                }

        case 4:

            flag = false;

            break;

        default:

            flag = false;

            break;

        }

    }

    return 0;

}

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.

**Output -**

|  |  |
| --- | --- |
|  |  |

// 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.

// Output -

#include<iostream>

using namespace std;

class *Customer*{

    public:

*string* name;

        long long int phone\_no;

};

class *Depositor*:public *Customer*{

    public:

        long long int accNo;

        double balance;

};

class *Borrower* : public *Depositor*{

    public:

        int loan\_no;

        double loan\_amount;

        void setBorrowerDetails(){

                cout<<"Enter Custome Name       : ";

                fflush(stdin);

                getline(cin, *this*->name);

                cout<<"Enter Customer Phone no  : ";

                cin>>*this*->phone\_no;

                cout<<"Enter Customer A/C No    : ";

                cin>>*this*->accNo;

                cout<<"Enter Loan No            : ";

                cin>>*this*->loan\_no;

                cout<<"Enter Loan Ammount       : ";

                cin>>*this*->loan\_amount;

        }

        void showDetails(){

            cout<<"\tCUSTOMER DETAILS"<<endl;

            cout<<"----------------------------------------------"<<endl;

            cout<<"Customer Name          : "<<*this*->name<<endl;

            cout<<"Customer Phone No      : "<<*this*->phone\_no<<endl;

            cout<<"Customer A/C No        : "<<*this*->accNo<<endl;

            cout<<"Balance                : "<<*this*->accNo<<endl;

            cout<<"----------------------------------------------"<<endl;

            cout<<"Loan No                : "<<*this*->loan\_no<<endl;

            cout<<"Loan Amount            : "<<*this*->loan\_no<<endl;

            cout<<"----------------------------------------------"<<endl;

        }

};

int main(){

*Borrower* b[20];

    int borrower\_Index = 0;

    int no\_of\_entries;

    bool flag = true;

    while(flag){

        int choice;

        cout<<"\n1) Enter Details\n2) Display Details\n3) Exit. "<<endl;

        cout<<"\tChoose an Option : ";

        cin>>choice;

        cout<<endl;

        switch (choice)

        {

        case 1:

            int no\_of\_cus;

            cout<<"Enter No. of customer details you want to enter: ";

            cin>>no\_of\_cus;

            try{

                while(no\_of\_cus != 0){

                    cout<<"-----------------------------"<<endl;

                    b[borrower\_Index].setBorrowerDetails();

                    borrower\_Index++;

                    no\_of\_cus--;

                }

            }catch(...){

                cout<<"Some Exception Occured"<<endl;

                break;

            }

            break;

        case 2:

            for(int i=0; i<borrower\_Index;i++){

                cout<<endl;

                b[i].showDetails();

            }

            break;

        case 3:

            flag = false;

            break;

        default:

            cout<<"\tPlease Enter Valid Option"<<endl;

            break;

        }

    }

}

9. Write a C++ program to implement the following class hierarchy:

Student: id, name

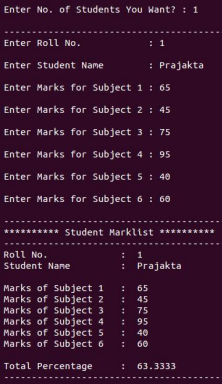
StudentExam (derived from Student): Marks of 6 subjects

StudentResult (derived from StudentExam) : percentage

Define appropriate functions to accept and display details.

Create 'n' objects of the StudentResult class and display the marklist.

**Output -**

****

// 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.

// Output -

#include<iostream>

using namespace std;

class *Student*{

    public:

        int id;

*string* name;

        void setStdDetails(){

            cout<<"Enter Roll No.        : ";

            cin>>*this*->id;

            cout<<"Enter Student Name    : ";

            fflush(stdin);

            getline(cin,*this*->name);

        }

        void getStdDetails(){

            cout<<"Roll no               : "<<*this*->id<<endl;

            cout<<"Student Name          : "<<*this*->name<<endl;

        }

};

class *StudentExam* : public *Student*{

    public:

    //Marks of 6 subjects

    float m1,m2,m3,m4,m5,m6;

    void setMarsk(){

        cout<<"Enter Marks for Subject 1 : ";

        cin>>*this*->m1;

        cout<<"Enter Marks for Subject 2 : ";

        cin>>*this*->m2;

        cout<<"Enter Marks for Subject 3 : ";

        cin>>*this*->m3;

        cout<<"Enter Marks for Subject 4 : ";

        cin>>*this*->m4;

        cout<<"Enter Marks for Subject 5 : ";

        cin>>*this*->m5;

        cout<<"Enter Marks for Subject 6 : ";

        cin>>*this*->m6;

    }

    void getMarks(){

        cout<<endl<<"Marks Of Subject 1  : "<<m1;

        cout<<endl<<"Marks Of Subject 2  : "<<m2;

        cout<<endl<<"Marks Of Subject 3  : "<<m3;

        cout<<endl<<"Marks Of Subject 4  : "<<m4;

        cout<<endl<<"Marks Of Subject 5  : "<<m5;

        cout<<endl<<"Marks Of Subject 6  : "<<m6<<endl;

    }

};

class *StudentResult*:public *StudentExam*{

    public:

        float percentage;

        void calPercentage(){

            float total = m1 + m2 + m3 + m4 + m5 + m6;

            percentage = (total/600) \* 100;

        }

        void getPercentage(){

            cout<<" Total Percentage     : "<<percentage<<" %"<<endl;

        }

};

int main(){

    int no\_of\_stds;

    bool flag = true;

    cout<<"Enter No. of Students You Want  : ";

    cin>>no\_of\_stds;

    int temp = no\_of\_stds;

*StudentResult* \*s;

    try{

         s = **new** *StudentResult* [no\_of\_stds];

    }catch(*bad\_alloc* a){

        cout<<"Allocation Failed";

    }

    int i = 0;

    while(temp != 0){

        cout<<"---------------------"<<endl;

        (s+i)->setStdDetails();

        (s+i)->setMarsk();

        (s+i)->calPercentage();

        i++;

        temp--;

    }

     i = 0;

     cout<<endl;

    cout<<"------------------------------------------------"<<endl;

    cout<<"     \*\*\*\*\*\*\*\*\* Student MarkList \*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"------------------------------------------------"<<endl;

    while(no\_of\_stds != 0){

        (s+i)->getStdDetails();

        (s+i)->getMarks();

        (s+i)->getPercentage();

        no\_of\_stds--;

        i++;

        cout<<endl;

        cout<<"----------------------------"<<endl;

    }

    return 0;

}

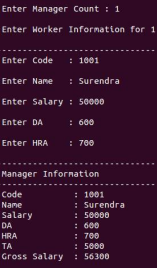
10. Consider two base classes

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.

**Output -**

****

// 10. Consider two base classes

// worker(int code, char name, float salary),

// officer(float DA, HRA)

// class manager(float TA(is 10% of salary), gross salary) is derived from both base classes. Write necessary member functions.

// Output -

#include<iostream>

using namespace std;

class *worker*{

    public:

        int code;

*string* name;

        float salary;

};

class *officer*{

    public:

        float DA, HRA;

};

class *manager*: public *worker*, public *officer*{

    public:

    float TA;

};