1. 定义一个CPerson类，有姓名、ID、性别等属性，有一个显示各个属性的方法Display，定义该类的构造函数和析构函数，用此类定义派生类： CEmployee，并给派生类添加Salary属性，定义派生类的构造和析构函数，定义一个显示各个属性的方法Display和调整工资成员函数Raisesalary，在main函数中定义派生类的对象，查看构造函数和析构函数定义和执行情况，并调用基类和派生类的Display函数。（**编程上机验证，要求用public 和private两种继承方式验证**）

单文件实现方式：

class Cperson

{

protected:

char \*name;

char \*id;

char sex;

public:

Cperson(char \*name1,char \*id1,char sex1);

~Cperson();

void Display();

};

Cperson::Cperson(char \*name1,char \*id1,char sex1)

{

name=new char[strlen(name1)+1];

strcpy(name,name1);

id=new char[strlen(id1)+1];

strcpy(id,id1);

sex=sex1;

cout<<"类Cperson构造函数被调用"<<endl;

}

void Cperson::Display()

{

cout<<"name："<<name<<endl;

cout<<"id:"<<id<<endl;

cout<<"sex:"<<(sex=='F'?" 女":"男")<<endl;

}

Cperson::~Cperson()

{

delete name;

delete id;

cout<<"类Cperson析构函数被调用"<<endl;

}

class Cemployee:public Cperson

{

public:

Cemployee(char \*name1,char \*id1,char sex,double salary1);

~Cemployee();

void Display();

void RaiseSalary(double s);

protected:

double salary;

};

Cemployee::Cemployee(char \*name1,char \*id1,char sex,double salary1):Cperson(name1,id1,sex)

{

salary=salary1;

cout<<"类Cemployee构造函数被调用"<<endl;

}

Cemployee::~Cemployee()

{

cout<<"类Cemployee析构函数被调用"<<endl;

}

void Cemployee::Display()

{

Cperson::Display();

cout<<"salary:"<<salary<<endl;

}

void Cemployee::RaiseSalary(double s)

{

salary+=s;

}

void main()

{

Cperson cp("王红","3001",'F');

cp.Display();

Cemployee ce("张山","3002",'M',4756.1);

ce.Display();

cout<<"请输入你想增加的工资数:";

double s1;

cin>>s1;

ce.RaiseSalary(s1);

cout<<"调整工资后的职工信息："<<endl;

ce.Display();

}

多文件实现方式：

**Cperson.h:**

class Cperson

{

protected:

char \*name;

char \*id;

char sex;

public:

Cperson(char \*name1,char \*id1,char sex1);

~Cperson();

char\* Getname();

void Display();

};

**Cemployee.h:**

#include "Cperson.h"

class Cemployee:public Cperson

{

public:

Cemployee(char \*name1,char \*id1,char sex,double salary1);

~Cemployee();

void Display();

void RaiseSalary(double s);

private:

double salary;

};

**Cperson.cpp:**

#include <iostream.h>

#include <string.h>

#include "Cperson.h"

Cperson::Cperson(char \*name1,char \*id1,char sex1)

{

name=new char[strlen(name1)+1];

strcpy(name,name1);

id=new char[strlen(id1)+1];

strcpy(id,id1);

sex=sex1;

cout<<"类Cperson构造函数被调用"<<endl;

}

void Cperson::Display()

{

cout<<"name："<<name<<endl;

cout<<"id:"<<id<<endl;

cout<<"sex:"<<(sex=='F'?" 女":"男")<<endl;

}

Cperson::~Cperson()

{

delete name;

delete id;

cout<<"类Cperson析构函数被调用"<<endl;

}

char \* Cperson::Getname()

{

return name;

}

**Cemployee.cpp:**

#include <iostream.h>

#include <string.h>

#include "Cemployee.h"

Cemployee::Cemployee(char \*name1,char \*id1,char sex,double salary1):Cperson(name1,id1,sex)

{

salary=salary1;

cout<<"类Cemployee构造函数被调用"<<endl;

}

Cemployee::~Cemployee()

{

cout<<"类Cemployee析构函数被调用"<<endl;

}

void Cemployee::Display()

{

Cperson::Display();

cout<<"salary:"<<salary<<endl;

}

void Cemployee::RaiseSalary(double s)

{

salary+=s;

}

**Main.cpp**

#include <iostream.h>

#include "Cemployee.h"

void main()

{

Cperson cp("王红","3001",'F');

cp.Display();

Cemployee ce("张山","3002",'M',4756.1);

ce.Display();

cout<<"请输入你想增加的工资数:";

double s1;

cin>>s1;

ce.RaiseSalary(s1);

cout<<"调整"<<ce.Getname()<<"工资后的信息："<<endl;

ce.Display();

}