Program 1

Define a class Triangle and calculate the area of it.

Program code

#include<iostream>

#include<math.h>

using namespace std;

class Triangle

{

private:

double a,b,c;

public:

Triangle(double pa,double pb,double pc)

{

a=pa;b=pb;c=pc;

}

double s()

{

if(a+b>c && a+c>b && b+c>a)

{

double p=(a+b+c)/2;

return pow(p\*(p-a)\*(p-b)\*(p-c),0.5);

}

else

return 0;

}

};

int main()

{

Triangle t1(3,4,5);

if (t1.s())

cout<<t1.s()<<endl;

else

cout<<"The 3 number can't form a triangle";

return 0;

}

Program analysis

1. check whether the 3 number can form a triangle first.
2. Using Helen formula to calculate the area of the triangle

Program result

Program 2

Define a class Book to change the Press of the Book.

Program code

#include<iostream>

using namespace std;

class Book

{

public:

char \*name;

char \*author;

char \*press;

int numberOfPages;

Book()

{

}

Book(char \*pn, char \*pa, char \*pp, int pnu)

{

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

if(name!=0)

strcpy(name, pn);

author= new char[strlen(pa)+1];

if(author!=0)

strcpy(author, pa);

press= new char[strlen(pp)+1];

if(press!=0)

strcpy(press ,pp);

numberOfPages=pnu;

}

~Book()

{

delete[] name;

delete[] author;

delete[] press;

}

inline void printBook()

{

cout<<name<<endl;

cout<<author<<endl;

cout<<press<<endl;

cout<<numberOfPages<<endl;

}

void setPress(char \*pp="清华大学出版社")

{

press= new char[strlen(pp)+1];

if(press!=0)

strcpy(press ,pp);

}

};

int main()

{

Book b1("好书","张三","出版社",1024);

b1.setPress();

b1.printBook();

b1.setPress("高等教育出版社");

b1.printBook();

return 0;

}

Program analysis

Use char pointer to change the press member.

Program result

Program 3

Define a class Engineer and create two anonymous object

Program code

#include<iostream>

using namespace std;

class Engineer

{

public:

int id;

char\* name;

char\* position;

Engineer(int pid, char\* pn,char\* pp)

{

id=pid;

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

if(name!=0)

strcpy(name,pn);

strcpy(name,pn);

position=new char[strlen(pp)+1];

if(position!=0)

strcpy(position,pp);

}

Engineer()

{

}

void setEngineer(int pid, char\* pn,char\* pp)

{

id=pid;

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

if(name!=0)

strcpy(name,pn);

position=new char[strlen(pp)+1];

if(position!=0)

strcpy(position,pp);

}

void print()

{

cout<<id<<endl;

cout<<name<<endl;

cout<<position<<endl;

}

~Engineer()

{

delete[] name;

delete[] position;

}

};

int main()

{

Engineer e1[4];

int i;

for( i=0;i<4;i++)

e1[i].setEngineer(i+1,"zhangsan","mangner");

Engineer\* pe1=new Engineer(5,"zhangsan","mangner");

Engineer\* pe2=new Engineer(6,"zhangsan","mangner");

for( i=0;i<4;i++)

e1[i].print();

pe1->print();

pe2->print();

delete pe1;

delete pe2;

return 0;

}

Program analysis

Use Engineer point to point to the anonymous object.

Program result

Program 4

Define a class computer, then using copy constructor to create 4 objects

Program code

#include<iostream>

using namespace std;

class Computer

{

public:

int id;

char\* brand;

int mainFrequency;

int hardDisk;

Computer()

{

}

Computer(int pid, char\* pb, int pm, int ph)

{

id=pid;

brand=new char[strlen(pb)+1];

if(brand!=0)

strcpy(brand, pb);

mainFrequency=pm;

hardDisk=ph;

}

Computer(Computer& c)

{

id=c.id;

brand=new char[strlen(c.brand)+1];

if(brand!=0)

strcpy(brand, c.brand);

mainFrequency=c.mainFrequency;

hardDisk=c.hardDisk;

}

~Computer()

{

delete[] brand;

}

void printComputer()

{

cout<<id<<endl;

cout<<brand<<endl;

cout<<mainFrequency<<endl;

cout<<hardDisk<<endl;

}

void setid(int pid)

{

id=pid;

}

};

int main()

{

Computer c1(1001,"IBM",220,1024);

Computer c2[4];

int i;

for(i=0;i<4;i++)

{

c2[i]=c1;

c2[i].setid(1001+i+1);

}

for(i=0;i<4;i++)

c2[i].printComputer();

return 0;

}

Program analysis

Using deep copy to copy the object

Program result