一、 1.√

2.×

3.×

4.√

5.√

6.×

7.×

8.×

9. √

10.×

二、 1.B或A

2.B

3.B

4.D

5.D

6.C

7.C

8.C

三、

1.int \*p = a;

int &q = a;

cout<<a;

2. 320

100

1000

100

100

1000

des

des

des

des

3.红色是出错语句

#include <iostream>

using namespace std;

struct a\_ST

{

public:

a\_ST( int aVal = 1000 ):\_st(aVal){}

a\_ST(const a\_ST &aRef)

{

this->\_st = aRef.\_st;

}

void display() const

{

cout<<\_st<<endl;

cout<<KST<<endl;

}

private:

int \_st;

static int KST;

};

int a\_ST::KST = 0;

int main()

{

a\_ST st;

st.display();

a\_ST \*sp = new a\_ST;

return 0;

}

四、1.

Complex::Complex(double aReal,double aImage)

{

m\_dImag = aImage;

m\_dReal = aReal;

}

Complex::~Complex()

{}

double Complex::GetReal() const

{

return m\_dReal;

}

double Complex::GetImag() const

{

return m\_dImag;

}

Complex Complex::Add(Complex &aRef) const

{

Complex result;

result.m\_dReal = m\_dReal + aRef.m\_dReal;

result.m\_dImag = m\_dImag + aRef.m\_dImag;

return result;

}

2. #include<iostream>

using namespace std;

class String{

public:

String(const char\* str=””);

String(const String &other);

~String() {delete[] m\_data;}

void display() const;

private:

char \*m\_data;

};

String::String(const char\* str)

{

m\_data = new char[strlen(str)+1];

strcpy(m\_data,str);

}

String::String(const String& other)

{

m\_data=new char[strlen(other.m\_data)+1];

strcpy(m\_data,other.m\_data);

}

void String::display() const

{

cout<<m\_data<<endl;

}