一、 1. √

2.×

3.×

4. √

5. √

6.×

7.×

8.×

9.×

10.×

二、 1.B

2.B

3.B

4.A

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=NULL);
        String(const String &other);
        ~String() {delete[] m_data;}
        void display() const;
private:
        char *m_data;
};
String::String(const char* str)
{
        if (!str) m_data=0;
        else
        {
                m_data = new char[strlen(str)+1];
                strcpy(m_data,str);
        }
}
String::String(const String& other)
{
        if(this!=&other)
        {
                m_data=new char[strlen(other.m_data)+1];
                strcpy(m_data,other.m_data);
        }
}
void String::display() const
{
        cout<<m_data<<endl;
}
```