

、复习题 1

1-2 重载函数可以带有默认值参数，但是要注意二义性。

2-1 对象之间的相互作用和通信是通过消息。() 不是消息的组成部分。

- A. 接受消息的对象
- B. 要执行的函数的名字
- C. 要执行的函数的内部结构
- D. 函数需要的参数

2-5 在下面类声明中，关于生成对象不正确的是 ()。

class point { public: int x; int y; point(int a,int b) {x=a;y=b;} };

- A. point p(10,2);
- B. point *p=new point(1,2);
- C. point *p=new point[2];
- D. point *p[2]={new point(1,2), new point(3,4)};

5-2 阅读下面的程序，完成其中复制构造函数的代码。

```
#include <iostream>
using namespace std;
class CAT
{
public:
    CAT();
    CAT(const CAT&);
    ~CAT();
    int GetAge() const { return *itsAge; }
    void SetAge(int age){ *itsAge=age; }
protected:
    int* itsAge;
};
CAT::CAT()
{
    itsAge=new int;
    *itsAge =5;
}
CAT::CAT(const CAT& c)
{
    CAT new_obj() 5分 ;
    *(new_obj.itsAge)=*(c.itsAge) 5分 ;
    itsAge=new int; 5分 ;
    *itsAge = c.GetAge(); 5分 ;
}
CAT::~~CAT()
{
    delete itsAge; }
}
```

5-2 多种错误 ① (0分) 创建提问

拷贝构造函数本质还是构造函数，也要给自己的成员变量 **new** 一些必要的空间，只是初始化值由旧对象得到。
(注：默认拷贝构造函数和对象间=赋值，都只有浅拷贝)

复习题 1 答案：2-1 C; 2-5 C 没有默认构造函数;

```
1  int* p1 = new int;      //-1- int。
2  char* p2 = new char [m]; //-2- 一维char数组。
3  int** p3 = new int* [m];
4  for(int i = 0; i < m; i++){
5      p3[i] = new int [n];
6  }          //-3- 二维int数组。更高维类似。
7  //上述的int, char替换成其他数据类型或者结构/类都是类似的。
```

复习题 5

5-1 Run the following program, Enter: 1, the output is: 55 34 21 13 8 5 3 2 1 1

```
#include <iostream>
using namespace std;

enum ERROR{UnderFlow,OverFlow};
template<typename T>
class StackTemplate {
    enum { ssize = 100 };
    T stack[ssize];
    int top;
public:
    StackTemplate() top(0) {}
    void push(const T& i) {
        if (top >= ssize) throw OverFlow;
        stack[top++] = i;
    }
    T pop() {
        if (top < 0) throw UnderFlow;
        return stack[top--];
    }
    int size() const
    { return top; }
};
```

1分

1分

1分

1分

top < 0

stack[--top]

想想可能真的那么简单吗？遇到多态必

查条件：

- ①原型完全相同
- ②virtual
- ③指针/引用传入

5-2 Run the following program, the output is: B::f()

```
#include <iostream>
using namespace std;
class A{
public:
    void f() { cout<<"A::f()\n"; }
};
class B:public A{
public:
    void f() {cout<<"B::f()\n"; }
};
int main()
{
    B b;
    A &p = b;
    cout << p. f();
    return 0;
}
```

1分

1分

virtual f()

p.

```

#include <iostream>
using namespace std;
class IndexError{};

template <typename T>
class ARRAY
{
    size_t m_size;
    T *m_ptr;
public:
    ARRAY(size_t size) : m_size(size)
    {
        m_ptr = new T[size];
        memset(m_ptr, 0, size*sizeof(int));
    }
    ~ARRAY()
    {
        delete[] m_ptr;
    }
    T& at(int index);
};

template <typename T>
ARRAY::at(int index)
{
    if(index<0|| index >= m_size )
    {
        throw IndexError();
    }
    return m_ptr[index];
}

```

ARRAY<T>

```

class ERROR{};
class STRING
{
    char *m_pStr;
    int m_len;
public:
    STRING(char *str=NULL){
        if (str != NULL) {
            m_len = strlen(str);
            m_pStr = new char[m_len] 1分 ;
            strcpy(m_pStr, str 1分 );
        }
        else {
            m_len = 0;
            m_pStr = NULL;
        }
    }
    char* 1分 operator=(char *str)
    {
        char* 1分 m_pStr ;
        m_len = strlen(str)+1;
        m_pStr = new char[m_len];
        strcpy(m_pStr, str 1分 );
        return m_pStr 1分 ;
    }

    bool operator==(STRING str) const 1分
    {
        return ( strcmp 1分 (m_pStr, str.m_pStr)== 0);
    }
}

```

new char[m_len+1]

STRING &

delete []

*this

复习题 6

1-1 对单目运算符重载为友元函数时，可以说明一个形参。而重载为成员函数时，不能显式说明形参。

1-4 因为静态成员函数不能是虚函数，所以它们不能实现多态。

析构函数可以带参数

复习题 6 答案：1-1 T; 1-4 T;