东南大学考试卷（A卷）

课程名称 计算机科学基础II 考试学期 11-12-3 得分

适用专业 信息工程 考试形式 闭卷 考试时间长度 120分钟

题目	一	二	三	四	总分
得分					
批阅人					

姓名 学号 密 封 线

一．单选题：（请将答案填写在下面表格中。每空1分，共10分）

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)

1. 复制构造函数可用于3个方面，其中，____(1)____不能使用复制构造函数。

A．用基类的一个对象去初始化一个派生类的对象

B．函数的形参是类的对象

C．函数的返回值是类的对象

D．用一个类的对象去初始化该类的另一个对象

2. 在一个派生类的成员函数中，试图调用其基类的成员函数"void show();"，但无法通过编译。这说明____(2)____。

A. show()是基类的保护成员　　B. show()是基类的私有成员

C. 派生类的继承方式为私有　　D. 派生类的继承方式为保护

3. 如果表达式x*y+z中，"*"是作为友元函数重载的，"+"是作为友元函数重载的,则该表达式为____(3)____。

A. y.operator*(operator+(x, y),z)　　B. x.operator+(operator*(x, y),z)

C. operator+(operator*(x, y),z)　　D. x.operator+(operator*(x, y))

4. 下列对派生类的描述中，错误的是____(4)____。

A. 派生类至少有一个基类　　B.一个派生类可以作为另一个派生类的基类

C. 派生类的缺省继承方式是private　　D.派生类只继承了基类的公有成员和保护成员

5. 关于类的输入、输出重载操作符>>和<<，描述正确的是____(5)____。

A. 必须定义为类的成员函数

B．插入符重载为类的友元函数，提取符定义必须为类的成员函数

C．提取符重载为类的友元函数，插入符定义必须为类的成员函数

D．均可重载为类的友元函数

6. 栈底至栈顶依次存放元素1、2、3、4，在第五个元素5入栈前，栈中元素可以出栈，则出栈序列可能是____(6)____。

A. 12345　　B. 42351　　C. 43251　　D. 34125

7. 关于动态内存分配，对 delete 运算符的下列说法中错误的是____(7)____。

A. 应与 new 运算符配合使用

B. 可以在类的成员函数中使用

C. 对同一个指针变量可任意多次使用该运算符

D. 若 a 是一个二维数组，则 delete[] a;可以释放 a 所占用的存储空间

8.以下类中声明了 4 个函数(原型)，它们被调用时，不分配 this 指针的是____(8)____。

```
class A{
    public:
            int  i;
            A(int);          //1)
            friend  A  data(A&);    //2)
            void  show();        //3)
            static void Print(A&);    //4)
};
```

A. 3) 4)　　B. 2) 4)　　C. 1) 3)　　D. 1) 2)

9. 下列叙述中正确的是____(9)____。

A. 循环队列有队头和队尾两个指针，因此，循环队列是非线性结构

B. 在循环队列中，只需要队头指针就能反映队列中元素的动态变化情况

C. 在循环队列中，只需要队尾指针就能反映队列中元素的动态变化情况

D. 循环队列中元素的个数是由队头指针和队尾指针共同决定

10. 当定义派生类的对象时，调用构造函数的正确顺序是____(10)____。

A. 先调用基类的构造函数，再调用派生类的构造函数

B. 先调用派生类的构造函数，再调用基类的构造函数

C. 调用基类的构造函数和调用派生类的构造函数之间的顺序无法确定

D. 调用基类的构造函数和调用派生类的构造函数是同时进行的

二.填空题（每空 2 分，共 20 分）

1. 设有以下类的定义：

```
class A
{
   int  A1;
   protected:  int A2;
   public:    int A3;
};
```

```
class B: protected A
{
    int b1;
    protected:  int b2;
    public:  int b3;
};
```

```
class C: private B
{
    int c1;
    protected:  int c2;
    public:  int c3;
};
```

请按访问权限写出派生类 C 中具有的成员。

私有成员：_____

保护成员：_____

公有成员：_____。

2. 下面程序段，1)、2)、3）三行各输出什么？

```
fstream file("ff.dat", ios::out | ios::binary);
for(char ch='b';ch<'j';ch++)     file.write(&ch,sizeof(char));
int pos=file.tellp();
cout<<"当前指针位置:: "<<pos<<endl;              //1)
for(ch='A';ch<'E';ch++)      file.write(&ch,sizeof(char));
file.seekg(pos);
file.read(&ch,sizeof(char));
cout<<"The data stored is "<<ch<<endl;           //2)
file.seekp(0,ios::beg);
for(ch='b';ch<'f';ch++)     file.write(&ch,sizeof(char));
cout<<"The data stored is "<<ch<<endl;           //3)
```

1）行：_____

2）行：_____

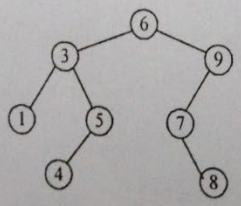
3）行：_____

3. C++有两种多态性：静态多态性和动态多态性。

静态多态性是通过_____实现的；

动态多态性是通过_____实现的；

4. 请写出下面一棵二叉树的后序遍历的结果_____。



5. 某一维数组有6个元素，它们的初始状态（第0遍）为 5 4 10 20 12 3，若用选择排序法对该数组按升序排序，请写出第三遍排序的结果。（手工排序）

_____

三. 读程序写结果（每空 2 分，共 40 分）

1.写出下面程序的运行结果（8 分）

```
#include <iostream>
using namespace std;
class   Cdata
{
    int psdata;
public:
   static int pbdata;
   Cdata(int x=0,int y=0){psdata=x;pbdata=y;}
   friend ostream& operator <<(ostream & out,Cdata &obj)
   {
       out<<obj.psdata<<'\t'<<obj.pbdata<<endl;
       return out;
   }
};
int Cdata::pbdata=0;
int main(void)
{
   Cdata data1;
   cout<<data1;
   Cdata data2(4,9);
   cout<<data1<<data2;
   cout<<Cdata::pbdata<<endl;
   return 0;
}
```

1. 运行结果为：

2. 写出下面程序的运行结果（14 分）

```
#include   <iostream>
using namespace std;
class Point
{
    int x,y;
public:
    Point(int a=0,int b=0):x(a),y(b){cout<<"Point Constructor!\n";}
    ~Point(){cout<<"Point Destructor!\n";}
    friend ostream &operator <<(ostream &out,Point &pobj)
    {
        out<<"Center coordinate:"<<pobj.x<<'\t'<<pobj.y<<'\t';
        return out;
```

```
    }
};
class Circle
{
    Point center;
    int r;
 public:
    Circle(int a,int b,int c):center(a,b),r(c)
    {
        cout<<"Circle Constructor!\n";
    }
    Circle(const Circle &t)
    {
        center=t.center;
        r=t.r;
        cout<<"Circle Copy Constructor!\n";
    }
    void Show()
    {   cout << center<<"radius:"<<r<<'\n';    }
  };
void   main(void)
{
      Circle cobj1(8,5,4);
      Circle   cobj2=cobj1;
      cobj2.Show();
}
```

2. 运行结果为：

3. 写出下面程序的运行结果（12 分）

```
#include <iostream>
using namespace std;
class CBase
{
    public:
        CBase( );
        CBase(int i);
        ~CBase( );
        virtual void Print( );
    private:
        int b;
};
```

3. 运行结果为：

```
CBase::CBase( )
{
    b=0;
    cout<<"Default constructor of CBase.\n";
}
CBase::CBase(int i)
{
    b=i;
    cout<<"Constructor of CBase." <<endl;
}
CBase::~CBase( )
{
    cout<<"Destructor of CBase."<<endl;
}
void CBase::Print( )  { cout<<b<<endl; }
class CDerive:public CBase
{
    public:
        CDerive(int i=0,int j=0);
        ~CDerive( );
        void Print( );
    private:
        int c;
};

CDerive::CDerive(int i,int j):CBase(i)
{
    c=j;
    cout<<"Constructor of CDerive."<<endl;
}
CDerive::~CDerive( )
{   cout<<"Destructor of CDerive.\n";  }
void CDerive::Print( )
{
    CBase::Print( );
    cout<<c<<endl;
}
int main( )
{
    CDerive obj(7,6);
```

```
    CBase *pb=&obj;
    pb->Print( );
    return 0;
}
```

4. 写出下面程序的运行结果（6分）

```
#include <iostream>
using namespace std;
typedef   double datatype;
datatype Div(datatype x, datatype y);
void main( )
{
    try
    {   cout<<"5/2="<<Div(5,2)<<endl;
        cout<<"8/0="<<Div(8,0)<<endl;
        cout<<"7/1="<<Div(7,1)<<endl;
    }
    catch(datatype x)
    {   cout<<"x="<<x<<"\n except of deviding zero.\n";    }
    cout<<"Just hold on.\n";
}
datatype Div(datatype x,datatype y)
{
    if(y==0)
        throw y;
    return x/y;
}
```

4．运行结果为：

四．完善程序题（每空2分，共30分）

1. 以下程序用冒泡算法实现降序排序。其中函数 sort 完成排序工作，buffer 为指向排序实数数组的首地址，length 为数组长度；函数 swap 交换两个双精度实数的值。请完善程序。

```
void swap( _____) {
    double temp = x1;

    _____;
    x2 = temp;
}
void sort(double * buffer, int length) {
    for (int i = 0; i < length - 1; i++){
        bool noswap = true;
```

```
        for (int j = length - 1; _____; j--){
            if (_____){
                swap(buffer[j], buffer[j - 1]);
                noswap = false;
            }
        }
        if (noswap) _____;
    }
}
```

2. 类 Node 实现节点基本功能：域 info 承载数据，而指针 next 指向下一个节点；单向链表类 List 采用了无头节点方式，即 head 指向的就是该链表的第一个有效节点，若该链表是空的则 head 值是 NULL，同时用 tail 指向该链表尾部，即最后一个有效节点。各函数功能描述如下：1）成员函数 append 用于在表尾部增加一个节点，同时该增加节点所承载的数据与 item 等值；2）成员函数 reverse 用于翻转该链表各节点链接顺序；3）成员函数 empty 删除链表所链接的所有节点并释放动态分配的节点内存；4）友元运算符<<实现链表各节点承载内容的顺序输出。请完善程序实现各个功能。

```
#include <iostream>
using namespace std;
class Node
{
    _____
    double info;
    Node * next;
    Node(const double & item = 0) { info = item; next = NULL; }
};

class List
{
    Node * head;
    Node * tail;
public:
    List() { _____; }
    ~List() { empty( ); }
    bool append(const double & item);
    void reverse();
    void empty();
    friend ostream & operator<<(ostream & sink, List & source);
};
```

```
bool List::append(const double & item)
{
    Node * insert = new Node (item);
    if (insert){
        if (tail){
                ____________________________;
            tail = insert;
        }
        else
            head = tail = insert;
        return true;
    }
    else return false;
}
void List::reverse()
{
    Node * p = head;
    head = NULL;

    ____________________________;
    while (p != NULL){
        Node * q = p;
        p = p->next;

        ____________________________;
        head = q;
    }
}
void List::empty()
{
    while (head){
        Node * temp = head;
        head = head->next;

        ____________________________;
    }
    head = tail = NULL;
}
```

```
ostream & operator<<(ostream & sink, List & source)
{
    int count = 0;
    Node * p = source.head;
    sink<<"List item(s) : "<<endl;
    while (p){
        count++;
        sink<<count<<'\t'<<p->info<<endl;
        p = p->next;
    }

    _____;
}
```

3. 模板函数 search 实现对已按降序排列的数组 arr 的对半递归查找：1）x 为需要查找的值；2）arr 是已按降序排列的数组首地址；3）first 与 last 分别为查找数组的首末元素索引。如果该数组中包含与 x 值相同的元素则返回该元素在数组的索引值，否则返回-1。请完善程序。

```
_____
int search(const Z & x, const Z arr [], const int first, const int last)
{
    int mid = -1;
    if (first <= last){
        mid = (first + last) / 2;
        if (arr[mid] > x)

            _____;
        else if (x > arr[mid])
            mid = search (x, arr, first, mid - 1);
    }
    return _____;
}
```