**模拟试题（第三套）**

**一、单项选择题（每小题2分，共20分）**

1. C++程序的基本功能模块为（ ）。

A. 表达式 B. 标识符 C. 语句 D. 函数

2. 当处理特定问题的循环次数已知时，通常采用（ ）来解决。

A. for循环 B. while循环 C. do循环 D. switch语句

3. 下面函数原型语句正确的是（ ）。

A. int Function(void a); B. void Function (int);

C. int Function(a); D. int(double a);

4. 假定变量m定义为“int m=7;”，则定义变量p的正确语句为（ ）。

A. int p=&m; B. int \*p=&m; C. int &p=\*m; D. int \*p=m;

5. 假定变量a和pa定义为“double a[10], \*pa=a;”，要将12.35赋值给数组a中的下标为5的元素，不正确的语句是（ ）。

A. pa[5]=12.35; B. a[5]=12.35;

C. \*(pa+5)=12.35; D. \*(a[0]+5)=12.35;

6. 假定AA为一个类，a为该类私有的数据成员，GetValue()为该类公有函数成员，它返回a的值，x为该类的一个对象，则访问x对象中数据成员a的格式为（ ）。

A. x.a B. x.a() C. x->GetValue() D. x.GetValue()

7. 类的构造函数是在定义该类的一个( )时被自动调用执行的。

A. 成员函数 B. 数据成员 C. 对象 D. 友元函数

8. 假定AB为一个类，则执行“AB x;”语句时将自动调用该类的( )。

A. 带参构造函数 B. 无参构造函数 C. 拷贝构造函数 D. 赋值重载函数

9. 假定一个类AB只含有一个整型数据成员a，当用户不定义任何构造函数时，系统为该类定义的无参构造函数为( )。

A. AB() {a=0;} B. AB(int aa=0): a(aa) {}

C. AB(int aa): a(aa) {} D. AB() {}

10. 如果表达式a++中的“++”是作为普通函数重载的运算符，若采用运算符函数调用格式，则可表示为（ ）。

A. a.operator++() B. operator++(a)

C. operator++(a,1) D. operator++(1,a)

**二、填空题（每小题2分，共20分）**

1. 存储字符’a’和字符串”a”分别需要占用\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_个字节。

2. 重载一个函数的条件是：该函数必须在参数的\_\_\_\_\_\_\_\_或参数对应的\_\_\_\_\_\_\_\_\_\_上与其它同名函数不相同。

3. 假定p是一个指向整数对象的指针，则用\_\_\_\_\_\_\_\_表示该整数对象，用\_\_\_\_\_\_\_\_表示指针变量p的地址。

4. 元素a[i]和b[i][j]的指针访问方式分别为\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_。

5. 已知有定义“int x, a[]={5,7,9}, \*pa=a;”，在执行“x=++\*pa;”语句后，x的值是\_\_\_\_\_\_\_\_，\*(pa+1)的值为\_\_\_\_\_\_\_\_。

6. 在一个用数组实现的队列类中，假定数组长度为MS，队首元素位置为first，队列长度为length，则队列为空和队列为满的条件分别为\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_。

7．对类中\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_成员的初始化只能通过在构造函数中给出的初始化表项来实现的。

8．假定AB为一个类，该类中含有一个指向动态数组空间的指针成员pa，则在该类的析构函数中应该包含有一条\_\_\_\_\_\_\_\_\_\_\_\_\_\_语句。

9. 假定派生类及其基类中都定义有同名函数成员，若要在派生类中调用基类的同名函数成员时，就需要在其函数名前加上\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_。

10. 派生类的成员函数能够直接访问所含基类的\_\_\_\_\_\_\_\_\_\_成员函数和\_\_\_\_\_\_\_\_\_\_\_成员函数。

**三、程序填充题，对程序、函数或类中划有横线的位置，根据题意按标号把合适的内容填写到程序后面的标号处。（每小题5分，共20分）**

1. 统计字符串中英文字母个数的程序。

#include<iostream.h>

int count(char str[]);

void main(){

char s1[80];

cout<<”Enter a line:”;

cin>>s1;

cout<<”count=”<<count(s1)<<endl;

}

int count(char str[]){

int num=0; //给统计变量赋初值

for(int i=0;str[i];i++)

if (str[i]>=’a’ && str[i]<=’z’ ||\_\_\_(1)\_\_\_ ) \_\_\_(2)\_\_\_;

\_\_\_(3)\_\_\_;

}

2. 对数组a[n]中的n个元素按升序进行的选择排序算法。

void SelectSort(int a[], \_\_\_(1)\_\_\_)

{

int i,j,k;

for(i=1;i<n;i++) { //进行n-1次选择和交换

k=i-1;

for(j=i;j<n;j++)

if(a[j]<a[k]) \_\_\_(2)\_\_\_;

int x=a[i-1]; a[i-1]=a[k]; \_\_\_(3)\_\_\_;

}

}

3. 下面是一维数组类ARRAY的定义，ARRAY与普通一维数组的区别是：(a)用()而不是[]进行下标访问，(b)下标从1而不是从0开始，(c)要对下标是否越界进行检查。

class ARRAY{

int \*v; //指向存放数组数据的空间

int s; //数组大小

public:

ARRAY(int a[], int n);

~ARRAY(){delete []v;}

int size(){ return s;}

int& operator()(int n);

};

\_\_\_(1)\_\_\_ operator()(int n) // ()的运算符成员函数定义

{

if(\_\_\_(2)\_\_\_) {cerr<<"下标越界！"; exit(1);}

return \_\_\_(3)\_\_\_;

}

4. 一种类定义如下:

class Goods

{

private:

char gd\_name[20]; //商品名称

int weight; //商品重量

static int totalweight; //同类商品总重量

public:

Goods (char\*str,int w){ //构造函数

strcpy(gd\_name,str);

weight=w;

totalweight+=weight;

}

~Goods (){totalweight -= weight;}

char\* GetN(){\_\_\_(1)\_\_\_;} //返回商品名称

int GetW(){return weight;}

\_\_\_(2)\_\_\_ GetTotal\_Weight() { //定义静态成员函数返回总重量

\_\_\_(3)\_\_\_;

}

}

**四、理解问答题，写出前三小题的程序运行结果和指出后两小题的程序（或函数）所能实现的功能。（每小题6分，共30分）**

1. #include<iomanip.h>

const int N=3;

void main()

{

int a[N][N]={{7,-5,3},{2,8,-6},{1,-4,-2}};

int b[N][N]={{3,6,-9},{2,-8,3},{5,-2,-7}};

int i,j,c[N][N];

for(i=0;i<N;i++) //计算矩阵C

for(j=0;j<N;j++)

c[i][j]=a[i][j]+b[i][j];

for(i=0;i<N;i++) { //输出矩阵C

for(j=0;j<N;j++)

cout<<setw(5)<<c[i][j];

cout<<endl;

}

}

运行结果：

2. #include<iostream.h>

#include<string.h>

class A{

char \*a;

public:

A(char \*s) {

a=new char[strlen(s)+1];

strcpy(a,s);

cout<<a<<endl;

}

~A(){delete []a;}

};

void main() {

A x("xuxiaokai");

A \*y=new A("weirong");

delete y;

}

运行结果：

3. #include<iostream.h>

class A {

int a,b;

public:

A() {a=b=0;}

A(int aa, int bb) {a=aa; b=bb;}

int Sum() {return a+b;}

int\* Mult() {

int \*p=new int(a\*b);

return p;

}

};

void main() {

A x(2,3), \*p;

p=new A(4,5);

cout<<x.Sum()<<’ ’<<\*(x.Mult())<<endl;

cout<<p->Sum()<<’ ’<<\*(p->Mult())<<endl;

}

运行结果：

4. void fun5(char\* a, const char\* b)

{

while(\*b) \*a++=\*b++;

\*a=0;

}

函数功能：

5. template<class T>

bool fun8(T a[], int n, T key)

{

for(int i=0;i<n;i++)

if(a[i]==key) return true;

return false ;

}

函数功能：

**五、编程题（每小题5分，共10分）**

1. 按照下面函数原型语句编写一个递归函数求出并返回数组a中n个元素的平方和。

int f(int a[],int n);

2. 根据下面类中拷贝构造函数的原型写出它的类外定义。

class Array {

int \*a; //指向动态分配的整型数组空间

int n; //记录数组长度

public:

Array(int aa[], int nn); //构造函数，利用aa数组长度nn初始化n，

//利用aa数组初始化a所指向的数组空间

Array(Array& aa); //拷贝构造函数

Array& Give(Array& aa); //实现aa赋值给\*this的功能并返回\*this

Array Uion(Array& aa); //实现\*this和aa中的数组合并的

//功能，把合并结果存入临时对象并返回

int Lenth() {return n;} //返回数组长度

void Print() { //输出数组

for(int i=0; i<n; i++)

cout<<a[i]<<’ ’;

cout<<endl;

}

};