**模拟试题（第四套）**

**一、单项选择题（每小题2分，共20分）**

1. 程序中主函数的名字为（ ）。

A. main B. MAIN C. Main D. 任意标识符

2. 为了提高程序的运行速度，可将不太复杂的功能用函数实现，此函数应选择（ ）。

A. 内联函数 B.重载函数 C.递归函数 D.函数模板

3. 将两个字符串连接起来组成一个字符串时，选用（ ）函数。

A. strlen() B. strcap() C. strcat() D. strcmp()

4. 用new运算符创建一维整型数组的正确语句是\_\_\_\_\_\_\_\_\_\_。

A. int \*p=new a[10]; B.int \*p=new float[10];

C. int \*p=new int[10]; D.int \*p=new int[5]={1,2,3,4,5}

5. 假定有定义“int b[10]; int \*pb;”，则不正确的赋值语句为（ ）。

A. pb=b; B. pb=&b[0]; C. pb=new int; D. pb=b[5];

6. 假定AA为一个类，a为该类公有的数据成员，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->a D. x.a

7. 假定AB为一个类，则执行“AB x(a,5);”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

8. 对于任一个类，用户所能定义的析构函数的个数至多为( )。

A. 0 B. 1 C. 2 D. 任意个

9. 对类中引用成员的初始化是通过构造函数中给出的( )实现的。

A. 函数体 B. 参数表 C. 初始化表 D. 初始化表或函数体

10. 如果表达式a==b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) B. b.operator==(a)

C. operator==(a,b) D. operator==(b,a)

**二、填空题（每小题2分，共20分）**

1. 常数-4.205和6.7E-9分别具有\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_位有效数字。

2. 元素类型为double的二维数组a[4][6]共有\_\_\_\_\_\_\_\_个元素，共占用\_\_\_\_\_\_\_\_字节的存储空间。

3. 假定对二维数组a[3][4]进行初始化的数据为{{3,5,6},{2,8},{7}}，则a[1][1]和a[2][3]分别被初始化为\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_。

4. 假定p为指向二维数组int d[4][6]的指针，则p的类型为\_\_\_\_\_\_\_\_，其中一个元素d[i][j]表示成指针访问方式为\_\_\_\_\_\_\_\_\_\_\_\_。

5. 已知变量a定义为“int a=5;”，要使ra成为a的引用，则ra应定义为\_\_\_\_\_\_\_\_\_\_\_\_，要使rb指向a，则rb应定义为\_\_\_\_\_\_\_\_\_\_\_\_。

6. 若只需要通过一个成员函数读取其数据成员的值，而不需要修改它们，则应在函数头的后面加上\_\_\_\_\_\_\_\_关键字；若只需要读取引用参数的值，不需要对其修改，则应在该参数说明的开始使用\_\_\_\_\_\_\_\_\_\_关键字。

7．假定一个类对象数组为A[N]，当定义该数组时，将自动调用该类的无参构造函数的次数为\_\_\_\_\_\_\_\_次，当离开它的作用域时，将自动调用该类析构函数的次数为\_\_\_\_\_\_\_\_次。

8．假定AB为一个类，则类定义体中的“AB(AB& x);”语句为该类\_\_\_\_\_\_\_\_\_\_\_\_\_\_的原型语句，而“operator=(AB& x);”为该类\_\_\_\_\_\_\_\_\_\_\_\_\_\_的原型语句。

9. 在定义一个派生类时，使用\_\_\_\_\_\_\_\_\_\_关键字表示为私有继承，使用\_\_\_\_\_\_\_\_\_\_关键字表示为公有继承。

10. 重载一个运算符时，该运算符的\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_以及操作符的个数不允许改变。

**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. 在输出屏幕上打印出一个由字符’\*’组成的等腰三角形，该三角形的高为5行，从上到下每行的字符数依次为1,3,5,7,9。

#include<iostream.h>

void main()

{

int i,j;

for(i=1;\_\_\_(1)\_\_\_;i++) {

for(j=1;j<=9;j++)

if(j<=5-i || \_\_\_(2)\_\_\_) cout<<’ ’;

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

cout<<endl;

}

}

2. 从一个字符串中删除所有同一字符后得到一个新字符串并输出。

#include<iostream.h>

const int len=20;

void delstr(char a[],char b[],char c);

void main() {

char str1[len],str2[len];

char ch;

cout<<"输入一个字符串:";

cin>>str1;

cout<<"输入一个待删除的字符:";

cin>>ch;

delstr(str1,str2,ch);

cout<<str2<<endl;

}

void delstr(char a[],char b[],char c)

{

int j=0;

for(int i=0; \_\_\_(1)\_\_\_; i++)

if(\_\_\_(2)\_\_\_) b[j++]=a[i];

b[j]=\_\_\_(2)\_\_\_;

}

3. 已知一维数组类ARRAY的定义如下，ARRAY与普通一维数组区别是：其重载的运算符[ ]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(n<0 || \_\_\_(2)\_\_\_) {cerr<<"下标越界！"; exit(1);}

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

}

4. 一个类定义如下：

class Point

{

private:

int x, y;

public:

Point(){x=y=0;}

Point(int x0,int y0) {x=x0;y=y0;}

int GetX() {return x; }

int GetY() {return y; }

void Print(){cout<<"Point("<<x<<","<<y<<")"<<endl;}

\_\_\_(1)\_\_\_; //友元函数声明

\_\_\_(2)\_\_\_; //友元函数声明

};

Point operator+(Point& pt,int dd)

//加号操作符重载函数，实现Point类对象与整数的加法

{

Point temp=pt;

temp.x+=dd;

temp.y+=dd;

return temp;

}

Point operator+(Point& pt1,Point& pt2)

//加号操作符重载函数，实现两个Point类对象的加法

{

Point temp=pt1;

temp.x+=pt2.x;

temp.y+=pt2.y;

\_\_\_(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<iostream.h>

const int B=2;

void main()

{

int p=1,s=1;

while(s<50) {

p\*=B;

s+=p;

}

cout<<"s="<<s<<endl;

}

运行结果：

2. #include<iostream.h>

class CE {

private:

int a,b;

int getmin() {return (a<b? a:b);}

public:

int c;

void SetValue(int x1,int x2, int x3) {

a=x1; b=x2; c=x3;

}

int GetMin();

};

int CE::GetMin() {

int d=getmin();

return (d<c? d:c);

}

void main()

{

int x=5,y=12,z=8;

CE \*ep;

ep=new CE;

ep->SetValue(x+y,y-z,10);

cout<<ep->GetMin()<<endl;

CE a=\*ep;

cout<<a.GetMin()\*3+15<<endl;

}

运行结果：

3. #include<iostream.h>

class A {

int a[10]; int n;

public:

A(int aa[], int nn): n(nn) {

for(int i=0; i<n; i++) a[i]=aa[i];

}

int Get(int i) {return a[i];}

int SumA(int nn) {

int s=0;

for(int j=0; j<nn; j++) s+=a[j];

return s;

}

};

void main() {

int a[]={2,5,8,10,15,20};

A x(a,4);

A y(a,6);

int d=1;

for(int i=0; i<4; i++) d\*=x.Get(i);

int f=y.SumA(5);

cout<<"d="<<d<<endl;

cout<<"f="<<f<<endl;

}

运行结果：

4. #include<iostream.h>

#include<stdlib.h>

#include<time.h>

const int N=10;

int ff(int x, int y) {

int z;

cout<<x<<'+'<<y<<'=';

cin>>z;

if(x+y==z) return 1; else return 0;

}

void main()

{

int a,b,c=0;

srand(time(0)); //初始化随机数序列

for(int i=0;i<N;i++) {

a=rand()%20+1; //rand()函数产生0-32767之间的一个随机数

b=rand()%20+1;

c+=ff(a,b);

}

cout<<"得分:"<<c\*10<<endl;

}

程序功能：

5. char \*f(char \*s){

int n=strlen(s);

char\* r=new char[n+1];

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

if(s[i]>='a' && s[i]<='z') r[i]=s[i]-'a'+'A';

else r[i]=s[i];

r[n]=’\0’;

return r;

}

程序功能：

**五、编程题（每小题5分，共10分）**

1. 编写一个函数，分别求出由指针a所指向的字符串中包含的每种十进制数字出现的次数，把统计结果保存在数组b的相应元素中，该函数的原型如下。

void fun(char\* a, int b[]);

2. 根据下面类中CompareBig 函数成员的原型和注释写出它的类外定义。

class AA {

int\* a;

int n;

int MS;

public:

void InitAA(int aa[], int nn, int ms) {

if(nn>ms) {cout<<"Error!"<<endl; exit(1);}

MS=ms;

n=nn;

a=new int[MS];

for(int i=0; i<n; i++) a[i]=aa[i];

}

int CompareBig(AA& b); //比较\*this与b的大小，从前向后按两数组

//中的对应元素比较，若\*this中元素值大则返回1，若b中

//元素值大则返回-1，若相等则继续比较下一个元素，直到

//一个数组中无元素比较，此时若两者的n值相同则返回0，

//否则若\*this中的n值大则返回1，若b中的n值大则返回-1。

};