

杭州电子科技大学学生考试卷（ A ）卷

考试课程	C++程序设计 计算机语言与程序设计(乙)		考试日期	08 年 6 月 18 日		成 绩	
课程号		教师号		任课教师姓名			
考生姓名		学号（8 位）		年 级		专 业	

答案请写在答题纸上

一、单项选择题(每空 2 分，共 20 分)

1. 下列结构中， _____ 适合用来描述链表节点。
A. struct A {int a; int next}; B. struct B {int a; B *next};
C. struct C {int a; A *next}; D. struct D {T1 a; T2 *next};
2. 若有说明语句:char c= ‘\65’ ;则变量 c _____。
A. 包含 1 个字符 B. 包含 2 个字符
C. 包含 3 个字符 D. 说明不合法，c 的值不确定
3. 若已定义 x 和 y 为 double 类型,则表达式:x=2, y=x+5/2 的值是： _____。
A. 2 B. 3 C. 4.0 D 4.5
4. _____是 C 语言提供的合法的关键词。
A. smitch B. shor C. break D. default
5. 下列语句_____正确定义了数组。
A. int s1[10],s2[2,3]; B. int n,a[n];
C. char s [5]="ABC"; D. int x(5);
6. 对于拷贝构造函数，正确的描述是_____。
A. 在 C++ 语言中，如果不自定义类的拷贝构造函数，则每个类都有默认的拷贝构造函数
B. 必须为每个类定义拷贝构造函数
C. 如果要使用拷贝构造函数，则必须在类中先定义
D. 当定义了类的构造函数时，如果要使用拷贝构造函数，则必须定义拷贝构造函数
7. 在 C++ 语言中，对函数参数默认值描述正确的是____。
A. 每个函数只能有一个参数设定默认值
B. 一个函数的参数若有多个，则参数默认值的设定可以不连续
C. 函数参数必须设定默认值
D. 在设定了参数的默认值后，该参数后面定义的所有参数都必须设定默认值
8. 下列各种函数中， _____不是类的成员函数。
A. 构造函数 B. 析构函数 C. 友元函数 D. 拷贝构造函数

9. 对于结构中定义的成员，其隐含访问权限为_____。
A. public B. protected C. private D. static
10. 关键字 ____ 可以用来说明常量。
A. malloc B. new C. const D. class

二、阅读理解题(每题 10 分， 共 50 分)阅读下面的程序，写出每个程序的运行结果。

1. #include <iostream>
using namespace std;
int Do (int x)
{

 if (x == 0)
 return 1;
 int count = 0;
 while (x > 0) {
 x /= 10 ;
 count++;
 }
 return count;
}
int main ()
{

 int A [] = {3, 1234, 654321, 0, 12345};

 for (int i = 0; i < sizeof (A) / sizeof (int); i++)
 cout << Do (A [i]) << endl;
 return 0;
}

2. #include <iostream>
using namespace std;
#include <iostream>
using namespace std;
int Fun (char *str)
{

 int count = 0;
 int i = 0;
 while (str [i] != '\0') {
 if (str [i] >= '0' && str [i] <= '9')

<pre> count += str [i] - '0' ; i++; } return count; } int main() { cout << Fun ("HangZhou") << endl; cout << Fun ("12345") << endl; cout << Fun ("HangZhou12345") << endl; cout << Fun ("12345+54321") << endl; return 0; } 3. #include <iostream.h> class sample { public: sample(); sample(int); ~sample(); void display(); protected: int x; }; sample::sample() : x(0) { cout<<" constructing normally\n" ; } sample::sample(int m) : x (m) { cout<<" constructing with a number:" <<x<<endl; } sample::~sample() { cout<<" destructing\n" ; } void sample::display()</pre>	<pre>{ cout<<" display a number:" <<x<<endl; } void main() { sample *p = new sample; sample obj2(18); p->display(); obj2.display(); delete p; } 4. #include <iostream.h> template <class T> class sample { public: sample() : n (0) {} ; sample(T x) : n (x) {} sample<T> operator +(const sample<T>&); void disp() {cout<<" n=" <<n<<endl;} protected: T n; }; template <class T> sample<T> sample<T>::operator +(const sample<T> &s) { sample<T> temp; temp.n=n+s.n; return temp; } void main() { sample<int> s1(10),s2(20),s3; s3=s1+s2; s1.disp (); s2.disp (); s3.disp(); }</pre>
---	--

<pre>5. class Student{ protected: string coursename; //课程名 int classhour; //学时 int credit; //学分 public: Student() {coursename="#";classhour=0;credit=0;} virtual void Calculate() {credit=classhour/16;} void SetCourse(string str,int hour){ coursename=str; classhour=hour; } int GetHour() {return classhour;} void Print() { cout<<coursename<<'\\t'<<classhour<<"学时" <<'\\t'<<credit<<"学分"<<endl; } }; class GradeStudent:public Student{ public: GradeStudent() {} ; void Calculate() {credit = classhour /20;} }; int main() { Student s,*ps; GradeStudent g; s.SetCourse("数学",80); g.SetCourse("英语",60); ps=&s; ps->Calculate(); ps->Print(); ps=&g; ps->Calculate(); ps->Print(); return 0; }</pre>	<p>三、程序设计(每题 10 分， 共 30 分)</p> <p>1. 设计一函数 fcount, 统计一个正整数数组中有几个数组元素是三位偶数。写一主调函数，用键盘输入一个整型数组 arr 的 20 个元素值，调用 fcount 函数之后, 在主调函数中输出显示统计结果。</p> <p>2. 编写一个函数 CountLetter，统计出字符串中有多少个不同的英文字母；大小写字母认为是同一个。CountLetter (“Hello”) 的结果为 4， CountLetter (“ A Sample string”) 的结果为 11。</p> <p>3. 下面是使用某矩阵类 CMatrix 的例子和运行结果，请设计完成该矩阵类 CMatrix。</p> <pre>int main () { int S1[] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}; CMatrix A (3,5), B (3,5); A.Assign (S1); //数组元素赋给矩阵元素 A.Display (); //矩阵显示 int S2[] = {2,0,3,4,5,1,7,8,9,10,11,2,3,14,10}; B.Assign (S2); //数组元素赋给矩阵元素 B.Display (); //矩阵显示 CMatrix C; C = A+ B; //矩阵相加 C.Display (); return 0; }</pre> <p>运行结果：</p> <pre>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 2 0 3 4 5 1 7 8 9 10 11 2 3 14 10 3 0 6 8 10 7 14 16 18 20 22 14 16 28 25</pre>
---	--