**C++程序设计基础复习题（选择题）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **题号** | **题干** | **选项A** | **选项B** | **选项C** | **选项D** |
| 1 | 以下程序输出的结果是( ) #include <iostream> using namespace std; int main() {  int i,a[10];  for(i=9;i>=0;i--)  a[i]=10-i;  cout<<a[2]<<a[5]<<a[8];  return 0; } | 258 | 741 | 852 | 369 |
| 2 | 以下关于数组的描述正确的是( ) | 数组的大小是固定的，所有数组元素的类型必须相同 | 数组的大小是可变的，可以有不同的类型的数组元素 | 数组的大小是可变的，但所有数组元素的类型必须相同 | 数组的大小是固定的，但可以有不同的类型的数组元素 |
| 3 | 程序运行中需要从键盘上输入多于一个数据时，各数据之间应使用( )符号作为分隔符. | 逗号或分号 | 逗号或回车 | 空格或回车 | 空格或逗号 |
| 4 | 已知int i=0, x=1, y=0;在下列选项中，使i的值变成1的语句是( ) | if( x==y ) i++; | if( x&&y ) i++; | if( x||y ) i++; | if( !x ) i++; |
| 5 | 在Ｃ＋＋中，引用数组元素时，其数组下标的数据类型允许是( ) | 整型常量或整型表达式 | 浮点型常量 | 任何类型的表达式 | void |
| 6 | 执行语句for（i=0; i<3;i++）;后，变量i的值( ) | 2 | 3 | 5 | 4 |
| 7 | int a[10];给数组a的所有元素分别赋值为1、2、3、……10的语句是（ ） | for(i=1;i<11;i++)a[i-1]=i; | for(i=1;i<11;i++)a[i]=i; | for(i=1;i<11;i++)a[0]=1; | for(i=1;i<11;i++)a[i+1]=i; |
| 8 | 多重if~else语句嵌套使用时，寻找与else配对的if的方法是( )。 | 缩进位置相同的if | 其上最近的if | 其上最近未匹配的if | 下面最近的if |
| 9 | 为表示关系x>=y>=z，下面表达式中正确的是( )。 | (x>=y)&&(y>=z) | (x>=y)and(y>=z) | (x>=y)or(y>=z) | (x>=y>=z) |
| 10 | 以下表达式中，结果为false的是( )。 | 0 | '0' | 3.15 | -0.5 |
| 11 | 对使用关键字new所开辟的动态存储空间，释放时必须使用( )。 | free | create | delete | realse |
| 12 | 下面程序输出结果是（ ） #include<iostream> using namespace std; int main() {  int a=6,b=6;  if(a>5)  a-=1;  b+=1;   else   a+=1;  b-=1;   cout<<"a="<<a<<" ";  cout<<"b="<<b<<endl;  return 0; } | a=5 b=6 | a=5 b=7 | 5 7 | 编译出错 |
| 13 | 下面的MyClass类中声明了"+"运算符重载函数的原型，如果在主函数中有如下定义 MyClass a , b ; 那么执行语句 a+b ;时，编译器将a+b解释为( )。 | operator + (a,b) | b.operator + (b) | a.operator+b | a.operator + (b) |
| 14 | 下列说法中，错误的是( ) | 每个函数都有一个函数头和一个函数体，主函数也不例外 | 程序是由若干个函数组成的，但是必须有、而且只能有一个主函数 | 主函数只能调用用户函数或系统函数,用户函数可以相互调用 | 每个语句必须独占一行，语句的最后可以是一个分号，也可以是一个回车换行符号 |
| 15 | C++的合法注释是( ) | // This is a C program | "This is a C program" | /\*This is a C program/\* | /This is a C program/ |
| 16 | 当创建一个类对象时，系统自动调用\_\_\_\_\_并给予对象的数据成员进行初始化。 | 静态函数 | 析构函数 | 构造函数 | 友元函数 |
| 17 | 以下不正确的是( ) | for（；；）表示无限循环 | for（）表示无限循环 | 语句for（i=0；；i++）表示无限循环 | while（1）表示无限循环 |
| 18 | 在C和C++语言中，换行符是\_\_\_\_\_。 | '\n' | '\t' | '\v' | '\b' |
| 19 | 以下正确的函数原型(函数声明）是( ) | void f1(int ,int); | void f1(x,y); | void f1(int x,y); | f1(int x;int y); |
| 20 | 一个C++程序的执行是从（ ） | 本程序文件的第一个函数开始,到本程序main函数结束 | 本程序的main函数开始,到本程序文件的最后一个函数结束 | 本程序文件的第一个函数开始,到本程序文件的最后一个函数结束 | 本程序的main函数开始,到main函数结束 |
| 21 | 如果说明语句char c='\072';则变量c( ) | 包含1个字符 | 包含3个字符 | 说明不合法，c的值不确定 | 包含2个字符 |
| 22 | 在函数定义中的形参属于( ) | 静态变量 | 局部变量 | 全局变量 | 寄存器变量 |
| 23 | char x[]="abcdefg";  char y[]={'a','b','c','d','e','f','g'}; 对于以下叙述正确的是( ) | 数组x和数组y的长度相同 |  | 数组x和数组y等价 | 数组x的长度小于数组y的长度 |
| 24 | ( )能正确输出结果：C++ | char s[3]={'C','+','+','\0'};cout<<s<<endl; | char s[3]={'C','+','+'};cout<<s<<endl; | char s[]="C++";cout<<s<<endl; | char s[3]="C++";cout<<s<<endl; |
| 25 | 已知int a=1,b=2,c=3;以下语句执行后a,b,c的值是( ) if(a>b) c=a; a=b; b=c; | a=2, b=3, c=1 | a=1, b=2, c=3 | a=2, b=3, c=3 | a=2, b=3, c=2 |
| 26 | 以下选项中合法的用户标识符是( ) | \_2Test | long | A.dat | 3Dmax |
| 27 | 若有定义 int a=5;则执行完语句a+=a-=a\*a之后， a的值（ ） | -40 | 3 | -12 | 0 |
| 28 | 正确表示"x是大于5小于10的数"的表达式是（ ） | x>5|x<10 | x>5||x<10 | 5<x<10 | x>5&&x<10 |
| 29 | 执行下面的程序段后，变量ｋ中的值为( ) int k=3,s[2]; s[0]=k; k=s[1]\*10; | 不定值 | 33 | 30 | 10 |
| 30 | 在C++中,如果函数没有返回值，那么应该声明为（　　）类型 | char | double | void | int |
| 31 | 已知int x=50,y=80,z=90; 则执行语句if(x>y) z=x; x=y; y=z; 后，x, y, z的值是\_\_\_\_\_。 | 80,90,90 | 80,90,80 | 50,80,90 | 80,90,50 |
| 32 | 下面程序运行结果是( ) #include<iostream> using namespace std; int main() {  int a,b; for(a=1,b=1;a<=100;a++) { if(b>=20)break;  if(b%3!=1)  b-=5; else  b+=3;   } cout<<a<<endl;  return 0; } | 7 | 8 | 9 | 10 |
| 33 | 只能作用于整型数据的运算符是( )。 | - | % | \* | / |
| 34 | 下列关于类的权限的描述错误的是( )。 | 类的对象只能访问该类的公有成员 | 一个类可以将另一个类的对象作为成员 | 普通函数不能直接访问类的公有成员，必须通过对象访问 | 类本身的成员函数只能访问自身的私有成员 |
| 35 | 设int m=7，n=12；则表达式结果为3的是（ ） | n%=(m-m%5) | n%=m-m%5 | (n%=m)-(m%=5) | n%=(m%=5) |
| 36 | 对类中声明的变量，下列描述中正确的是( )。 | 任何情况下都可被该类所有实例共享 | 属于全局变量 | 只有类的公有成员可以使用 | 类中所有成员函数都可以使用 |
| 37 | 一个类可包含析构函数的个数是( )。 | 1个 | 0个 | 至少一个 | 0个或多个 |
| 38 | int n=1;  while（n--）  n++;  那么while循环执行次数是( ) | 2 | 1 | 0 | 无穷次 |
| 39 | 下列关于类和对象的叙述中，错误的是\_\_\_\_\_。 | 对象是类的抽象 | 对象是类的具体实例 | 类是对某一类对象的抽象 | 类和对象的关系是一种数据类型和变量的关系 |
| 40 | 设x和y均为bool变量，则表达式x&&y结果为真的条件是（ ） | 其中一个为假 | 它们均为假 | 其中一个为真 | 它们均为真 |
| 41 | 下面程序输出结果是（ ） #include<iostream> using namespace std; int main() {  int a=100,x=10,y=20,ok1=5,ok2=0;  if(x<y)  if(y!=10)  if(!ok1)   a=1;  else if(ok2)  a=10;   a=-1;  cout<<a<<endl;  return 0; } | 1 | 值不确定 | 0 | -1 |
| 42 | 若有说明:int a[10]; 则对a数组元素的正确引用是( ) | a[10] | a[3.5] | a(5) | a[10-10] |
| 43 | #include <iostream> using namespace std; int main() {  int a=1,b=2,c=2,t;   while(a<b<c)  { t=a;a=b;b=t;c--;}  cout<<a<<","<<b<<","<<c;  return 0; } 执行上述程序，其运行结果是( ) | 2，1，1 | 1，2，1 | 1，2，0 | 2，1，0 |
| 44 | 以下对一维数组a的正确说明是（　　） | int a(10); | int n; cin>>n; int a[n]; | int n=10,a[n]; | const int n=10; int a[n]; |
| 45 | 假设A是一个类，下面语句“A a1[2]; A &pa = a1;”执行时，系统会调用\_\_\_\_\_次A类的构造函数。 | 0 | 3 | 2 | 1 |
| 46 | 下面程序运行结果是( ) #include <iostream> using namespace std; int main() {  int x=0,y=0;   while(x<15)  {  y++;  x+=y;  }  cout<<y<<","<<x;  return 0; } 。 | 5,7 | 20,8 | 6,12 | 5,15 |
| 47 | 设存在函数int max(int，int)返回两参数中较大值，若求22，59，70三者中最大值，下列表达式不正确的是( )。 | int m = max(59，max(22，70))； | int m = max(max(22，59)，70)； | int m = max(22，59，70)； | int m = max(22，max(59，70))； |
| 48 | 若以下说明： int a[12]={1,2,3,4,5,6,7,8,9,10,11,12}; char c='a',d,g; 则数值为４的表达式是（ ）。 | a[g-c] | a[4] | a['d'-'c'] | a['d'-c] |
| 49 | int x=-1; do  {  x=x\*x; } while(!x); 下列说法正确的是( ) | 是死循环 | 有语法错误 | 循环执行一次 | 循环执行两次 |
| 50 | 下面程序运行结果是( ) #include<iostream>  using namespace std; int main() {  int x=3;  do  {  x-=2;  cout<<x<<＂#＂;  }while(!(--x));  return 0; } | 1#-2# | 死循环 | 3#0# | 1 |
| 51 | 对于有返回值的函数，要结束函数运行必须使用语句（ ） | return | main | break | continue |
| 52 | 当一个类对象生命期结束时，系统自动调用\_\_\_\_\_并销毁对象。 | 析构函数 | 静态函数 | 友元函数 | 构造函数 |
| 53 | 假定一个字符串的长度为n，则定义存储该字符串的字符数组的长度至少为( ) | n | n+1 | n-1 | n+2 |
| 54 | 已知类定义如下： class Test{ public:  void Set(double val); private:  double value; }; 则类体外对Set成员函数的定义正确的是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。 | void Set(double val) {value=val;} | void Test::Set(double val) {val=value;} | Test::void Set(double val) {value=val;} | void Test::Set(double val) {value=val;} |
| 55 | 设d为字符变量，下列表达式中不正确的是\_\_\_\_\_。 | d=50 | d='n' | d='\n' | d="\n" |
| 56 | 如果没有为一个类定义任何构造函数的情况下，下列描述正确的是( )。 | 这个类不需要构造函数 | 编译器总是自动创建一个不带参数的构造函数 | 这个类没有构造函数 | 该类不能通过编译 |
| 57 | 下列对构造函数的描述中，( )是错误的。 | 函数名与类名完全相同 | 构造函数允许重载 | 构造函数允许使用缺省参数 | 返回类型应该为void |
| 58 | class T { int a; public: void setValue(int aa) { a=aa;} }; int main() { T t,\*p=&t; ( ) return 0; } 在上段代码的空中填入的合法语句是( )。 | t.a=5; | p->setValue(5); | p.setValue(5); | t->setValue(5); |
| 59 | 下列关于析构函数的描述中，( )是正确的。 | 析构函数可以有1个或多个参数 | 一个类中可以重载多个版本析构函数 | 析构函数不能指定返回值类型 | 析构函数名与类名完全相同 |
| 60 | 下列程序的运行结果为（ ） #include<iostream>  using namespace std; int main() {   int a=2;   int b=a+1;   cout<<a/b<<endl;  return 0; } | 0.666666666... | 0.66667 | 0 | 0.7 |
| 61 | 有关类性质说法错误的是\_\_\_\_\_。 | 析构函数和构造函数都不能有返回类型 | 一个类可以有多个构造函数，但只有一个析构函数 | 一个类中不可以声明具有类类型的数据成员 | 不能给析构函数指定参数 |
| 62 | 如下代码，对其描述正确的是( )。int \*p;p=new int(5); | 动态开辟5个int元素，初值不确定 | 存在语法错误 | 动态开辟5个int元素，初值为0 | 动态开辟初值为5的一个int元素 |
| 63 | 下列标识符中，( )是非法的。 | SIZE\_MB | 2002YEAR | YEAR2002 | \_SIZE\_MB |
| 64 | 下面不正确的字符串常量是( ) | ‘rst’ | “14’14” | “” | “0” |
| 65 | 对while 和do...while循环结构说法正确的是( ) | 只是表达形式不同 | 条件成立时，它们有可能一次也不执行 | do...while结构中的语句至少执行一次 | while结构中的语句至少执行一次 |
| 66 | 能用作C++变量名的是( )。 | 123abc | const | switch | \_1ab |
| 67 | 已知类声明如下“class A{ int a;};”，则类A的数据成员a的访问权限是\_\_\_\_\_。 | private | public | 不确定 | protected |
| 68 | class Fraction { int num,den; public: Fraction(int n=0.int d=1); ...... }; int main() { Fraction f1(3,5),f2(4),f3; f3=f1+f2; cout<<f3<<endl; return 0; } 为完成主函数的功能，类中还需添加( )函数声明。 | Fraction operator+(Fraction& f2); friend ostream& operator<<(ostream& out,const Fraction& f); | friend ostream& operator<<(ostream& out,const Fraction& f); | Fraction(); Fraction operator+(Fraction& f2); friend ostream& operator<<(ostream& out,const Fraction& f); | Fraction operator+(Fraction& f2); |
| 69 | 在C++语言中,非法的字符常量是( ) | “\n” | ‘\xaa’ | ‘\017’ | ‘\t’ |
| 70 | 判断char型变量n是否为小写字母的正确表达式为（ ） | ‘a’<=n<=‘z’ | (n>=‘a’)&&(n<=‘z’) | (‘a’>=n）| |(‘z’<=n) | （n>=a）&&（n<=z） |
| 71 | 若用数组名作为调用函数的实参，则传递给形参的是( ) | 数组元素的个数 | 数组的第一个元素值 | 数组的首地址 | 数组中全部的元素值 |
| 72 | 以下将一维数组ｍ正确初始化为全零的语句是（ ）。 | int m[10]=(0,0,0,0); | int m[10]={}; | int m[]={0}; | int m[10]={10\*2}; |
| 73 | 在循环语句的循环体中使用break语句的作用是( ) | 结束开关语句 | 结束程序执行 | 结束本次循环 | 结束该层的循环 |
| 74 | 求平方根函数的函数名为( ) | sqrt | pow | abs | cos |
| 75 | 若一个函数无返回值，则定义它时函数的类型应该为( ) | void | 任意 | 无 | int |
| 76 | C++中，系统自动为一个类生成默认构造函数的条件是该类没有自定义任何\_\_\_\_\_。 | 构造函数 | 有参构造函数 | 无参构造函数 | 成员函数 |
| 77 | 关键字( )定义允许在类外部使用的成员数据和成员函数。 | private | protected | public | virtual |
| 78 | 在C++语言中，换行符是（ ） | '\b' | '\t' | '\n' | '\f' |
| 79 | 已知类定义如下： class Test{ public:  Test(int a,int b,int c):x(c),z(a),y(b) { } private:  int x,y,z; }; 则Test类中3个数据成员的初始化顺序是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。 | z,x,y | x,y,z | x,z,y | z,y,x |
| 80 | main()函数的返回类型是( )。 | float | 任意类型 | 不返回值 | int |
| 81 | 下面有关for循环的正确描述是( ) | 在for循环中，不能用break语句跳出循环体 | for循环的循环体语句中，可以包含多条语句，但必须 用大括号括起来 | for循环只能用于循环次数已经确定的情况 | for循环是先执行循环体语句，后判断表达式 |
| 82 | 假定MyClass为一个类，执行"MyClass a[3], \*p;"语句时会自动调用该类型构造函数( )次。 | 4 | 5 | 2 | 3 |
| 83 | 有以下程序 #include<iostream> using namespace std; int fun(int x,int y) { return(x+y); } int main()  {  int a=2,b=5,c=8;  cout<<(fun((int)fun(a+c,b),a-c))<<endl;  return 0;  } 程序运行后的输出结果是（ ） | 9 | 编译出错 | 21 | 9.0 |
| 84 | 下列for循环的次数为（ ） for（i=0, x=0; !x&&i<=5; i++）; | 6 | 5 | 1 | 无限 |
| 85 | 下面正确的字符常量是( ) | 'W' | '\\'' | "C" | "\0" |
| 86 | 如没有使用private关键字定义类的数据成员，则默认为( )。 | private | protected | public | friend |
| 87 | 如下代码，对其描述正确的是( )。int \*p;p=new int[5]; | 动态创建了5个int数据所需连续内存空间，初值不确定 | 动态创建5个int数据所需的连续内存空间，初值为0 | 动态创建一个int所需内存空间，且初值为5 | 存在语法错误 |
| 88 | 现有定义int a;double b;floatc;char k;,则表达式a/b+c-k值的类型为:. | float | double | int | char |
| 89 | 以下数据中,不正确的数值或字符常量是(). | o13 | 9861 | 5L | 0 |
| 90 | 已知char a;int b;float c;double d;则表达式a-b+c-d结果为()型. | float | double | int | char |
| 91 | 以下语句中,:当字符串s1和s2相等时,能够输出"OK"的是() | if(s1==s2)printf("OK"); | if(s1[0]==s2[0])printf("OK"); | if(strcmp(s1,s2))puts("OK"); | if(!strcmp(s1,s2))puts("OK"); |
| 92 | 下列选项中不符合良好程序设计风格的是(). | 数据说明的次序要规范化 | 对所有输入的数据都要进行检查,确保数据的合法性 | 程序的效率第一,清晰第二 | 程序中要有必要的注释 |
| 93 | 关于字符常量,以下叙述正确的是(). | 所有的字符常量都可以作为整型量来处理 | 单引号中的大写字母和小写字母代表的是相同的字符常量 | 字符常量能包含大于一个的字符 | 空格不是一个字符常量 |
| 94 | 以下选项中,非法的字符常量是(). | 0' | \xAA' | \\' | \0201' |
| 95 | 下列关于类,对象,属性和方法的叙述中错误的是(). | 方法用于表示对象的行为 | 属性用于描述对象的状态 | 基于同一个类产生的两个对象不可以分别设置自己的属性值 | 类是对一类具有相同的属性和方法对象的描述 |
| 96 | 可在C++程序中,用作用户标识符的一组标识符是(). | 2c DO SIG | void define WORD | as\_b3 \_123 If | for -abc case |
| 97 | 下述错误的C++语言常量是(). | 5. | .25 | 0xf | 090 |
| 98 | 下列字符序列中是C++语言保留字的是(). | scanf | sizeof | sqrt | include |
| 99 | 以下不能正确进行字符串赋初值的语句是(). | char \*str="good!"; | char str[5]={'g','o','o','d',0}; | char str[5]="good!"; | char str[]="good!"; |
| 100 | 字符串"\\\'efg\'\\"的长度是:(). | 7 | 3 | 11 | 5 |
| 101 | 以下能正确进行字符串赋值的语句是 | char s[5];s="good!"; | char s[5]={'a','e','i','o','u'}; | char s[5]="good!"; | char \*s;s="good!"; |
| 102 | 可在C++程序中,用做用户标识符的一组标识符是(). | and \_2007 | case Bigl | Date y-m-d | Hi Dr.Tom |
| 103 | C++语言中的标识符只能由字母,数字和下划线三种字符组成,且第一个字符(). | 可以是字母,数字和下划线中,任一字符 | 必须为字母 | 必须为下划线 | 必须为字母或下划线 |
| 104 | 下列语句中符合C++语言语法的赋值语句是() | x=7+y,y++,x++ | x=7+y++=x+7; | x=7+y+z=x+z; | x=7+y,z=x+7; |
| 105 | 以下不符合C语言语法的赋值语句是(). | ++j; | y=(a=3,6\*5); | a=1,b=2 | a=b=5; |
| 106 | 字符串"\nabcdef\\"的长度是(). | 8 | 6 | 10 | 4 |
| 107 | 以下字符中,是转义字符的是(). | \b' | \a' | '\\' | \c' |
| 108 | 下列语句中符合C++语言语法的赋值语句是:(). | a=7+b,c=a+7; | a=7+b,b++,a+7 | a=7+b+c=a+7; | a=7+b++=a+7; |
| 109 | C++语言中,以下几种运算符的优先次序()的排列是正确的. | 由高到低为:算术运算符,关系运算符,!,&&,||,赋值运算符 | 由高到低为:算术运算符,关系运算符,赋值运算符,!,&&,|| | 由高到低为:!,算术运算符,关系运算符,&&,||,赋值运算符 | 由高到低为:!,&&,||,算术运算符,赋值运算符 |
| 110 | 在C++语言中,运算对象必须是整型数的运算符是(). | %和\ | \ | \*\* | % |
| 111 | 以下语句中,符合C++语言语法的赋值语句是(). | m+n=10; | a=10+b,b+=a+10; | a=7+b,c=a+7 | m=n |
| 112 | 设a和b均为 double型常量,且a=5.5,b=2.5,则表达式(int)a+b/b的值是(). | 6 | 5.500000 | 6.000000 | 6.500000 |
| 113 | 下列语句中符合C++语言语法的赋值语句是(). | i=7+1,i++,1+4 | i=1+j+k=i+2; | 1=1+i++=i+5; | i=1+2,k=j+2; |
| 114 | 对象实现了数据和操作的结合,是指对数据和数据的操作进行. | 隐藏 | 集成 | 封装 | 组合 |
| 115 | 字母A的ASCⅡ码为十进制数65,且C2为字符型,则执行语句C2='A'+'6'-'3';后,C2为()字符. | C | D | 不确定的值 | 69 |
| 116 | 下列语句中符合C++语言语法的语句是( ). | x=7+y,y++,z++; | x=y+2=x+y+z; | x=3+y++=x+3; | x=(3+b,z)=x+3; |
| 117 | 以下字符中,不是转义字符的是(). | '\b' | '\\' | '\c' | '\a' |
| 118 | 下列可用于C++语言用户标识符的一组是(). | a33,l12,Car | for,-abc,IF case | void,define,WORD | 2a,DO,size of |
| 119 | 判断两个字符串是否相等,正确的表达方式是(). | while(strcmp(s1,s2)=0) | while(strcmp(s1,s2)==0) | while(s1==s2) | while(s1=s2) |
| 120 | 下列选项中可作为C++语言合法常量的是(). | -80.0e | -80 | -8e1.0 | -80a |
| 121 | 以下符合C++语言语法的实型常量是(). | 8E-2 | 2.2E0.5 | 3.14.159E | E14 |
| 122 | 设x和y都是int类型,且x=1,y=2,则printf("%d%d",x,y,(x,y))的输出结果是(). | 1,2 | 1 2 2 | 输出值不确定 | 12 |
| 123 | 下面正确的字符常量是(). | '\9' | 'W' | "c" | '\' |
| 124 | 以下选项中,合法的一组C++语言数值常量是(). | 028.5e-3 -0xf | 0x8A 10,000 3.e5 | 12 OXa23 4.5e0 | .177 4c1.5 Oabc |
| 125 | 已知字母A的ASCII码为十进制数65,且c为字符型,则执行语句c='A'+'6'-'3';后,c中的值为:(). | 不确定的值 | D | d | c |
| 126 | C++语言中,运算对象必须是整型的运算符是(). | 〈= | %= | = | / |
| 127 | 以下数据中,属于字符串常量的是(). | 'ask' | 'm' | "ASK" | ASK |
| 128 | 以下选项中,可以作为C++语言中,合法整数的是(). | afbc | 110110B | 0Xbbc | 01988 |
| 129 | 下列字符序列中可用作C++标识符的一组字符序列是(). | S.b,sum,average,\_above | D56,r\_1\_2,name,\_st\_1 | class,day,lotus\_1,2day | #md,&12x,month,student\_n! |
| 130 | 以下数值中,不正确的整型常量是(). | 0xaaaa | 0x16 | 078 | 16543 |
| 131 | 判断字符串s1是否大于字符串s2,应该使用(). | if(strcmp(s1,s2)<0) | if(strcmp(s2,s1)<0) | if(s1>s2) | if(strcmp(s1,s2)) |
| 132 | 下列标识符中不合法的C++语言用户自定义标识符是(). | sin | \_ | enum | printf |
| 133 | 以下程序段中不能正确赋字符串(编译时系统会提示错误)的是(). | char t[]="abcdefg",\*s=t; | char s[10];s="abcdefg"; | char s[10];strcpy(s,"abcdefg"); | char s[10]="abcdefg"; |
| 134 | 下面各选项中均是C++语言合法标识符的选项组是(). | 33 we auto | \_43 3e\_else | 23\_me\_3ew | ER -DF 32 |
| 135 | 若已定义x和y为 double 类型,则表达式x=1,y=x+3/2 的值是(). | 1 | 2 | 2.0 | 2.5 |
| 136 | C++语言中,char 类型数据占(). | 2个字节 | 4个字节 | 8个字节 | 1个字节 |
| 137 | 以下不能将s所指字符串正确复制到t所指存储空间的是( ) | while(\*t=\*s){t＋＋；s＋＋} | for(i=0；t[i]=s[i]；i＋＋)； | do{\*t＋＋=\*s＋＋；}while(\*s)； | for(i=0,j=0；t[i＋＋]=s[j＋＋]；)； |
| 138 | 下述正确的C++语言常量是(). | E2 | 3e-3 | 1.5E2.5 | 5.0E |
| 139 | 下面各语句行中能正确进行赋值字符串操作的是(). | char s[5]={'a','b','c','d','e'}; | char s[5];scanf("%s",&s); | char \*s;s="ABCDEF"; | char \*s;gets(s); |
| 140 | 下述语句中在字符串s1和s2相等时显示"they are Equal"的是(). | if(s1==s2)puts("they are Equal"); | if(\*s1==\*s2)puts("they are Equal"); | if(!strcmp(s1,s2))puts("they are Equal"); | if(strcmp(s1,s2))puts("they are Equal"); |
| 141 | 以下数据中,不正确的数值或字符常量是(). | 8.9e1.2 | 82.5 | 10 | 0xff00 |
| 142 | 下列能正确定义字符串的语句是(). | char str="kx43"; | char str[]={'\064'}; | char str="; | char str[]="\0"; |
| 143 | 以下选项中,不能作为合法常量的是(). | 3.14e0.5 | 3.14e05 | 3.14e0 | 3.14e+5 |
| 144 | 有以下程序: #include<iostream> using namespace std; int main(){ int x; cin>>x; if(x<=3); else   if(x!=10)cout<<x; } 程序运行时,输入的值在哪个范围才会有输出结果. | 大于3且不等10的整数 | 小于3的整数 | 大于3或等于10的整数 | 不等于10的整数 |
| 145 | 当把4个表达式用作if语句的控制表达式时,有一个选项与其他3个选项含义不同,这个选项是(). | !k%2==1 | (k%2)!=0 | k%2==1 | k%2 |
| 146 | 两次运行下面的程序,如果从键盘上分别输入3和2,则输出结果是().    #include<iostream>  using namespace std; main() { int x;  cin>>x  if(x++>2)cout<<x;  else cout<<x--;  } | 4和3 | 4和2 | 3和2 | 4和1 |
| 147 | 当a=6,b=14,c=8,d=4时,执行下面程序段后,x的值是().    if(a<b) if(c<d)x=4  else  if(a<c) if(b<d)x=6;  else x=8;  else x=14;  else x=27; | 6 | 27 | 4 | 8 |
| 148 | 以下选项中,能用作用户标识符的是(). | unsigned | void | o | 8\_8 |
| 149 | 设有条件表达式:(EXP)?i++:j--,则以下表达式中(EXP)完全等价的是(). | (EXP!=0) | (EXP!=1) | (EXP==1) | (EXP==0) |
| 150 | 以下程序的执行结果是(). main() {int x=2,y=2,z=0; if(z<0) if(y>0)x=4; else x=5; printf("%d\t",x); if(z=y>0)x=7; else if(y=0)x=3; else x=6; printf("%d\t",x); printf("%d\t",z); } | 520 | 271 | 345 | 205 |
| 151 | 若有条件表达式exp?a++:b--,则以下表达式中能完全等价于表达式exp的是(). | (exp==1) | (exp!=0) | (exp==0) | (exp!=1) |
| 152 | 有如下程序: #include<iostream> using namespace std; main() { int x=1,a=0,b=0; switch(x) { case 0:b++; case 1:a++; case 2:a++;b++; } cout<<"a="<<a<<",b="<<b; } 该程序的输出结果是 | a=1,b=1 | a=2,b=2 | a=2,b=1 | a=1,b=0 |
| 153 | 在C++语言中,if语句后的一对原括号中,用以决定分支的流程的表达式(). | 只能用逻辑表达式或关系表达式 | 只能用逻辑表达式 | 可用任意表达式 | 只能用关系表达式 |
| 154 | 为表示关系x≥y≥z,应使用的C++语言表达式是(). | (x>=y)or(y>=z) | (x>=y>=z) | (x>=y)AND(y>=z) | (x>=y)&&(y>=z) |
| 155 | 下面标识符中合法的用户标识符为() | return | \_float | -int | (int)(1.6) |
| 156 | int a=1,b=2,c=3; if(a>b)a=b; if(a>c)a=c; ,则a的值为(). | 1 | 3 | 不一定 | 2 |
| 157 | 若有int x,y,z;,则执行以下程序段后的输出结果为: x=1;y=5;z=3; if(x>y)x=y; y=z; cout<<"x="<<x<<",y="<<y<<",z="<<z; | x=1,y=5,z=3 | x=1,y=3,z=3 | x=5,y=3,z=5 | x=5,y=3,z=3 |
| 158 | 若有定义:int a=9;,则执行语句a+=a -=a+a;后,a的结果是(). | 18 | 9 | -18 | -9 |
| 159 | 设有定义:int m=0;.以下选项的四个表达式中与其他三个表达式的值不相同的是(). | m++ | ++m | m+1 | m+=1 |
| 160 | 能正确表示逻辑关系"n≥5或n≤2"的C++语言表达式是() | n>=5 or n<=2 | n>=5||n<=2 | n>=5&&n<=2 | n>=2&n<=5 |
| 161 | 若k是整型,则以下程序段的执行结果是:() k=-3; if(k<=0)cout<<"####"; else cout<<"&&&&"; | &&&& | 有语法错误,无结果 | ####&&&& | #### |
| 162 | 正确的标识符是(). | x=8 | !at | z.2 | y2 |
| 163 | 若有int a=1,b=2,c=3;,则执行下列语句后a的值为(). if(a>b)a=b; if(a>c)a=c; | 2 | 1 | 不一定 | 3 |
| 164 | 下列程序的运行结果是(). main() { int a=2,b=3,c=4; if(a<b) if(b<0)c=0; else c+=1; cout<<c; } | 2 | 5 | 4 | 3 |
| 165 | 当a=4,b=5,c=7,d=6时,执行下面一段程序:   if(a<b) if(c<d)x=1;  else  if(a<c) if(b<c)x=2;  else x=3;  else x=4;  else x=5;    程序执行后,x的值为(). | 3 | 4 | 1 | 2 |
| 166 | 以下程序的运行结果是().   main() { int a=-5,b=1,c=1;  int x=0,y=2,z=0;  if(c>0)x=x+y;  if(a<=0) { if(b>0) if(c<=0)y=x - y;  } else if(c>0)y=x - y;  else z=y;  cout<<x<<","<<y<<","<<z;  } | 2,2,2 | 2,0,2 | 2,2,0 | 0,2,0 |
| 167 | C++语言中,switch后的括号内表达式的值可以是 | 只能为整型,字符型,枚举型 | 只能为整型 | 任何类型 | 只能为整型和字符型 |
| 168 | 下列不合法的用户标识符是(). | 4d | \_8\_ | Double | j2\_KEY |
| 169 | 当a=1,b=2,c=3,d=4时,执行下面程序段后,x的值是().  if(a<b) if(c<d)x=1;  else  if(a<c) if(b<d)x=2;  else x=3;  else x=6;  else x=7; | 3 | 1 | 6 | 2 |
| 170 | 以下标识符不是关键字的是(). | return | break | char | Switch |
| 171 | 有以下程序: #include<iostream> using namespace std; int main() { int x=1,y=0;  if(!x)y++;  else if(x==0)  if(x)y+=2;  else y+=3;  cout<<y; } 程序运行后的输出结果是(). | 1 | 3 | 0 | 2 |
| 172 | 已知int k=67;,则表达式k/10的值为; | 6.7 | 6.70 | 6 | 7 |
| 173 | 若有a=45,b=27,c=0;,则执行c=a>b?a:b后c的结果是(). | 72 | 27 | 18 | 45 |
| 174 | 有以下程序 main() {int a=0,b=0,c=0,d=0;  if(a=1)  b=1;  c=2;  else   d=3; cout<<a<<","<<b<<","<<c<<","<<d; } 程序输出结果是(). | 0,1,2,0 | 0,0,0,3 | 1,1,2,0 | 编译有错 |
| 175 | 下列运算符优先级最高的是(). | && | || | = | + |
| 176 | 以下标识符中不能作为合法的C++用户定义标识符的是(). | to | \_if | answer | signed |
| 177 | 下列运算符中优先级别最低的运算符是 | <= | != | + | || |
| 178 | 以下程序的运行结果是 main() { int a=2,b=-1,c=2 ;  if(a<b)  if(b<0)c=0;  else c+=1;  cout<<c; } | 1 | 0 | 2 | 3 |
| 179 | 以下是if语句的基本形式: if (表达式) 语句 其中,"表达式". | 可以是任意合法的表达式 | 必须是逻辑表达式或关系表达式 | 必须是逻辑表达式 | 必须是关系表达式 |
| 180 | 有以下程序:  int main() { int x=1,y=0,a=0,b=0;  switch(x)  { case 1:  switch(y)  { case 0:a++;break;  case 1:b++;break;  }  case 2:a++;b++;break;  case 3:a++;b++;  }  cout<<"a="<<a<<",b="<<b; } 程序的运行结果是(). | a=2,b=1 | a=2,b=2 | a=1,b=1 | a=1,b=0 |
| 181 | 以下程序的输出结果是(). int main() { int a=8,b=6,m=1;  switch(a%4)  { case 0:m++;break;  case 1:m++;  switch(b%3)  { default:m++;  case 0:m++;break;  }  }  cout<<m; } | 1 | 2 | 4 | 3 |
| 182 | 设有如下程序段: int k=10; while(k==0) k--; ,则下述说明中正确的是(). | 循环体执行一次 | 死循环 | 循环体一次也不执行 | 循环体执行10次 |
| 183 | 以下不构成无限循环的语句或语句组是(). | n=0；  do{＋＋n；}while(n<=0)； | for(n=0,i=1； ； i＋＋)n＋=i； | n=10； while(n)；{n--；} | n=0； while(1){n＋＋；} |
| 184 | 若有int y=10;,则执行下列语句后的输出为(). while(y--); printf("y=%d\n",y); | y=-1 | while 构成无限循环 | y=1 | y=0 |
| 185 | 下列程序执行输出的结果是().   f(int a) { int b=0;  static c=3;  a=c++;b++;  return(a);  } main() { int a=2,i,k;  for(i=0;i<2;i++)  k=f(a++);   cout<<k;  } | 3 | 4 | 5 | 6 |
| 186 | 以下程序的执行结果是(). main() {int num=0; while(num<=2){num++;cout<<num<<",";}} | 1,2,3, | 0,1,2 | 1,2, | 1,2,3,4, |
| 187 | 针对for 循环语句,for(表达式1;表达式2;表达式3),下列说法正确的是(). | break语句只能跳出最内层循环 | break语句可以一次性跳出多层嵌套循环 | continue语句不能跳出最外层循环 | continue语句可以一次性跳出多层嵌套循环 |
| 188 | 下面程序的运行结果是().    #include<iostream> using namespace std; main()   { int i;  for(i=1;i<=5;i++) { if(i%2)  cout<<"$";  else   cout<<"@";  }   } | @$@$@ | $@$@ | @$@$ | $@$@$ |
| 189 | 对 for(表达式1;;表达式3)可理解为(). | for(表达式1;表达式3;表达式3) | for(表达式1;0;表达式3) | for(表达式1;表达式1;表达式3) | for(表达式1;1;表达式3) |
| 190 | 下述循环的循环次数是(). int k=2; while(k=0)  ;k--; | 1次 | 0次 | 无限次 | 2次 |
| 191 | 下面程序是从键盘输入4位正整数,输入0或负数时结束循环.A处填写内容是().  main() {  int num;  do{  cin>>num;  }while(\_\_\_A\_\_\_); } | num>0 | !num | num==0 | !num !=0 |
| 192 | 若有int i=0;,则以下程序段中,while 循环的循环次数是() while(i<10) { if(i<1)continue; if(i==5)break; i++; } | 6 | 死循环,不能确定次数 | 1 | 4 |
| 193 | 阅读下列程序,则运行结果为(). #include<iostream> using namespace std; fun() { static int x=3;  x++;  return x; } int main() { int i,x;  for(i=0;i<3;i++)  x=fun();  cout<<x; } | 5 | 3 | 6 | 4 |
| 194 | 若有int mm;,以下程序段执行后输出结果是(). for(mm=3;;mm=mm-1); cout<<mm; | 3 | 死循环 | 2 | 1 |
| 195 | 以下程序的功能是计算:s=1+12+123+1234+12345 int main(void) { int t=0,s=0,i; for(i=1;i<=5;i++)  { t=i+\_\_\_\_\_\_;  s=s+t;  } cout<<"s="<<s; } | t | t\*100 | t+10 | t\*10 |
| 196 | 在下列选项中,没有构成死循环的是(). | for(;;); | int i=100; while(1){i=i+1; } | int s=36; while(s)--s; | int k=10000; do{k++;}while(k>10000); |
| 197 | 以下程序的输出结果是(). double f(int n) { int i;double s;  s=1.0;  for(i=1;i<=n;i++)  s+=1.0/i;  return s; } int main() { int i,m=3;float a=0.0;  for(i=0;i<m;i++)  a+=f(i); cout<<a; } | 5.500000 | 8.25 | 4.000000 | 3.000000 |
| 198 | t为int类型,进人下面的循环之前,t的值为0 while(t=1) {……} ,则以下叙述中,正确的是(). | 循环控制表达式不合法 | 循环控制表达式的值为1 | 其他说法都不对 | 循环控制表达式的值为0 |
| 199 | 若有int w=12;,对以下程序段,描述正确的是(). while(w=0)w=w-1; | 循环是无限循环 | 循环体语句一次也不执行 | 循环体执行了10次 | 循环体语句执行了一次 |
| 200 | 有以下程序: int main() { int i,j,m=55; for(i=1;i<=3;i++) for(j=3;j<=i;j++)m=m%j; cout<<m; } 程序的运行结果是(). | 3 | 2 | 1 | 0 |
| 201 | 有以下程序 int main() { int y=9;  for(;y>0;y--)  if(y%3==0)  cout<<--y; } 程序的运行结果是(). | 963 | 741 | 875421 | 852 |
| 202 | 有以下程序  main() { int y=9;  for(; y>0;y--) if(y%3==0)cout<<--y;  }   程序的运行结果是(). | 875421 | 741 | 963 | 852 |
| 203 | 以下语句的执行结果是(). int m=0,s=1; do{ s+=m++; }while(m<6); cout<<s; | 17 | 18 | 15 | 16 |
| 204 | 从键盘输入的整数中,找出最小值min,输入0时结束.请在A处填写正确语句(). main() { int min=0,a; do{ cin>>a; if(min>a)min=a; }while(\_\_\_A\_\_\_); cout<<min; } | a !=0 | a==0 | a>0&&a<0 | a=0 |
| 205 | 以下语句中,循环次数不为10次的语句是(). | i=1;m:if(i<=10){i++;goto m;} | i=1;do{i++;}while(i<=10); | i=10;while(i>0){--i;} | for(i=1;i<10;i++); |
| 206 | 若有说明int a[3][4];则a数组元素的非法引用是(). | a[4-2][0] | a[1][3] | a[0][4] | a[0][2\*1] |
| 207 | 下述对C++语言字符数组的描述中错误的是(). | 字符数组中的字符串可以整体输入,输出. | 不可以用关系运算符对字符数组中的字符串进行比较. | 字符数组中可以存放字符串. | 可以通过赋值运算符"="对字符数组整体赋值. |
| 208 | 若有说明:int array[3][3]={0};,则下面正确的叙述是() | 只有元素array[0][0]可得到初值 | 此说明语句不正确 | 数组array中每个元素均可得到初值0 | 数组array中各元素都可得到初值,但其值不一定为0 |
| 209 | 若二维数组a有m列,则在a[i][j]前的元素个数为(). | j\*m+i | i\*m+j+1 | j\*m+i+1 | i\*m+j |
| 210 | 已知有声明"char s[20]="Hello";",在程序运行过程中若要想使数组s中的内容修改为"Good",则以下语句能够实现此功能的是(). | strcpy(s,"Good"); | strcat(s,"Good"); | s[20]="Good"; | s="Good"; |
| 211 | 若有以下说明: int a[12]={1,2,3,4,5,6,7,8,9,10,11,12}; char c='a',d,g; ,则数值为4的表达式是() | a['d'-c] | a[4] | a['d'-'c'] | a[g-c] |
| 212 | 请阅读以下程序:   void fun(int b[]) { static int i=0;  do  { b[i]+=b[i+1];  }while(++i<2); }  int main() { int k,a[5]={1,3,5,4,9};  fun(a);  for(k=0;k<5;k++)cout<<a[k]; }   上面程序的输出是(). | 48579 | 48549 | 48999 | 13579 |
| 213 | 有以下程序:  main() { int aa[5][5]={{5,6,1,8},{1,2,3,4},{1,2,5,6},{5,9,10,2}};  int i,s=0;  for(i=0;i<4;i++) s+=aa[i][2];  cout<s;  } 程序运行后的输出结果是(). | 19 | 26 | 20 | 10 |
| 214 | 若有说明:int a[3][4];,则对a数组元素的正确引用是(). | a[2][4] | a[1+1][0] | a[1,3] | a(2)(1) |
| 215 | 以下对一维整型数组a的正确说明是(). | int n=10,a[n]; | int n; cin>>n; int a[n]; | #define SIZE 10(换行)int a[SIZE]; | int a(10); |
| 216 | char a1[]="abc",a2[80]="1234";将a1串连接到a2串后面的语句是(). | strcat(a2,a1); | strcpy(a2,a1); | strcpy(a1,a2); | strcat(a1,a2); |
| 217 | 对以下说明语句 int a[10]={6,7,8,9,10};的正确理解是(). | 将5个初值依次赋给a[6]至a[10] | 将5个初值依次赋给a[0]至a[4] | 因为数组长度与初值的个数不相同,所以此语句不正确 | 将5个初值依次赋给a[1]至a[5] |
| 218 | 若有说明:int a[][3]={1,2,3,4,5,6,7};,则a数组第一维的大小是(). | 2 | 无确定值 | 4 | 3 |
| 219 | 若已有定义: int a[]={0,1,2,3,4,5,6,7,8,9},\*p=a,i; 其中,0≤i≤9,则对a数组元素的引用不正确的是(). | p[i] | a[p - a] | \*(\*(a+i)) | \*(&a[i]) |
| 220 | 若有定义:int m[]={5,4,3,2,1},i=4;,则下面对m数组元素的引用中,错误的是(). | m[m[i]] | m[2\*2] | m[m[0]] | m[--i] |
| 221 | 定义int a[]={0,1,2,3,4,5,6,7,8,9},\*p=a,i;,其中,0≤i≤9,则对a数组元素的引用不正确的是(). | a[p-a] | \*(&a[i]) | p[i] | \*(\*(a+i)) |
| 222 | 若有以下说明: int a[12]={8,9,10,11,12}; 则值为9的表达式是(). | a[2] | a[3] | a[1] | a[4] |
| 223 | 以下程序的输出结果是 main() {char ch[2][5]={"6937","8254"},\*p[2]; int i,j,s=0; for(i=0;i<2;i++) p[i]=ch[i]; for(i=0;i<2;i++) for(j=0;p[i][j]>'\0' && p[i][j]<='9';j+=2) s=10 \* s+p[i][j]- '0'; pritnf("%d\n",s); } | 6385 | 69825 | 63825 | 693825 |
| 224 | 以下程序段的输出结果为(). for(i=4;i>1;i--) for(j=1;j<i;j++) putchar('#'); | # | ### | 无 | ###### |
| 225 | 若有说明int n[4][4],则n数组元素的非法引用是() | n[0][4] | n[5-2][0] | n[0][3\*1] | n[1][3] |
| 226 | int a[10]; 给数组a的所有元素分别赋值为1,2,3,……的语句是(). | for(i=1;i<11;i++)a[i-1]=i; | for(i=1;i<11;i++)a[i]=i; | for(i=1;i<11;i++)a[i+1]=i; | for(i=1;i<11;i++)a[0]=1; |
| 227 | 函数定义时的参数为形参,调用函数时所用的参数为实参,则下列描述正确的是(). | 实参与形参是双向传递 | 形参和实参可以同名 | 实参类型一定要在调用时指定 | 形参可以是表达式 |
| 228 | 存在多层嵌套循环时,下列说法正确的是(). | continue语句能够中,止包含该语句的最内层循环 | break语句可以中,止所有循环 | break语句能够中,止包含该语句的最内层循环 | continue语句可以中,止所有循环 |
| 229 | 以下函数调用语句中,实参的个数是(). func((e1,e2),(e3,e4,e5)); | 3 | 5 | 语法错误 | 2 |
| 230 | 若变量已正确定义,要将a和b中的数进行交换,下面不正确的语句是( ) | t=a;a=b;b=t; | t=b;b=a;a=t; | a=a+b,b=a-b,a=a-b; | a=t;t=b;b=a; |
| 231 | 下面对函数的叙述,正确的是(). | 函数中无论是否有 return语句,都能返回正确值 | 函数中没有 return语句,说明不带回值 | 函数不能嵌套定义 | 函数的返回值是通过 return语句获得的 |
| 232 | C++语言的if语句嵌套时,if 与else 的配对关系是(). | 每个else 总是与它上面的最近的if 配对 | 每个else 总是与最外层的if 配对 | 每个else 与if 的配对是任意的 | 每个else 总是与它上面的if 配对 |
| 233 | C++语言关于实参和形参说法正确的是(). | 实参和与之对应的形参占用同一存储单元 | 形参是虚拟的,不占用存储单元 | 实参和与之对应的形参各占用独立的存储单元 | 只有当实参与形参同名时,才共占用同一存储单元 |
| 234 | 在调用函数时,如果实参是简单变量,它与对应形参之间的数据传递方式是(). | 传递方式由用户指定 | 由实参传给形参,再由形参传回实参 | 地址传递 | 单向值传递 |
| 235 | 求平方根函数的函数名为(). | abs | cos | pow | sqrt |
| 236 | C++语言规定,函数返回值的类型是(). | 由return语句中的表达式类型所决定 | 由调用该函数时的主调函数类型所决定 | 由调用该函数时系统临时决定 | 由定义该函数时所指定的数值类型决定 |
| 237 | 若函数调用时的实参为变量,以下关于函数形参和实参的叙述中,正确的是(). | 函数的形参和实参分别占用不同的存储单元 | 函数的实参和其对应的形参共占同一存储单元 | 形参只是形式上的存在,不占用具体存储单元 | 同名的实参和形参占同一存储单元 |
| 238 | 在定义全局变量和局部静态变量的同时赋值,变量的初始化在()时确定的. | 编辑 | 调试 | 编译 | 运行 |
| 239 | 以下对C++语言函数的有关描述中正确的是(). | 在C++中调用函数时,只能把实参值传给形参,形参值不能返回给实参 | C++程序中有调用关系的所有函数必须放在同一个源程序文件中 | 没有返回值的函数不能被使用 | C++函数可以递归调用也可以嵌套调用 |
| 240 | 假设所有变量均为整型,则表达式(a=1,b=9,b++,a+b)的值是() | 9 | 10 | 11 | 12 |
| 241 | 用数组名作为函数调用时的实参,实际上传递给形参的是(). | 数组元素的个数 | 数组全部元素的值 | 数组第一个元素的值 | 数组首地址 |
| 242 | 对auto存储类型的变量,若不赋值,则其值是(). | 系统自动赋值0 | ' \0 ' | 根据情况赋值 | 随机的 |
| 243 | 关于建立函数的目的,以下正确的说法是(). | 减少程序文件所占内存 | 提高程序的执行效率 | 提高程序的可读性 | 减少程序的篇幅 |
| 244 | 若调用一个函数,且此函数中,没有 return语句,则正确的说法是:该函数(). | 没有返回值 | 返回一个不确定的值 | 返回若干个系统默认值 | 能返回一个用户所希望的值 |
| 245 | 以下对静态局部变量的叙述哪一个是不正确的(). | 静态局部变量在整个程序运行期间都不释放 | 在一个函数中定义的静态局部变量可以被另一函数调用 | 数值型静态局部变量的初值默认为0 | 静态局部变量是在编译时赋初值的,故它只被赋值一次 |
| 246 | 以下叙述中,不正确的是(). | 在一个函数内定义的变量只在本函数范围内有效 | 在一个函数内的复合语句中,定义的变量在本函数范围内有效 | 函数中的形式参数是局部变量 | 在不同的函数中,可以使用相同的名字的变量 |

**C++程序设计基础复习题（选择题答案）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **题号** | **题干** | **选项A** | **选项B** | **选项C** | **选项D** | **答案** |
| 1 | 以下程序输出的结果是( ) #include <iostream> using namespace std; int main() {  int i,a[10];  for(i=9;i>=0;i--)  a[i]=10-i;  cout<<a[2]<<a[5]<<a[8];  return 0; } | 258 | 741 | 852 | 369 | C |
| 2 | 以下关于数组的描述正确的是( ) | 数组的大小是固定的，所有数组元素的类型必须相同 | 数组的大小是可变的，可以有不同的类型的数组元素 | 数组的大小是可变的，但所有数组元素的类型必须相同 | 数组的大小是固定的，但可以有不同的类型的数组元素 | A |
| 3 | 程序运行中需要从键盘上输入多于一个数据时，各数据之间应使用( )符号作为分隔符. | 逗号或分号 | 逗号或回车 | 空格或回车 | 空格或逗号 | C |
| 4 | 已知int i=0, x=1, y=0;在下列选项中，使i的值变成1的语句是( ) | if( x==y ) i++; | if( x&&y ) i++; | if( x||y ) i++; | if( !x ) i++; | C |
| 5 | 在Ｃ＋＋中，引用数组元素时，其数组下标的数据类型允许是( ) | 整型常量或整型表达式 | 浮点型常量 | 任何类型的表达式 | void | A |
| 6 | 执行语句for（i=0; i<3;i++）;后，变量i的值( ) | 2 | 3 | 5 | 4 | B |
| 7 | int a[10];给数组a的所有元素分别赋值为1、2、3、……10的语句是（ ） | for(i=1;i<11;i++)a[i-1]=i; | for(i=1;i<11;i++)a[i]=i; | for(i=1;i<11;i++)a[0]=1; | for(i=1;i<11;i++)a[i+1]=i; | A |
| 8 | 多重if~else语句嵌套使用时，寻找与else配对的if的方法是( )。 | 缩进位置相同的if | 其上最近的if | 其上最近未匹配的if | 下面最近的if | C |
| 9 | 为表示关系x>=y>=z，下面表达式中正确的是( )。 | (x>=y)&&(y>=z) | (x>=y)and(y>=z) | (x>=y)or(y>=z) | (x>=y>=z) | A |
| 10 | 以下表达式中，结果为false的是( )。 | 0 | '0' | 3.15 | -0.5 | A |
| 11 | 对使用关键字new所开辟的动态存储空间，释放时必须使用( )。 | free | create | delete | realse | C |
| 12 | 下面程序输出结果是（ ） #include<iostream> using namespace std; int main() {  int a=6,b=6;  if(a>5)  a-=1;  b+=1;   else   a+=1;  b-=1;   cout<<"a="<<a<<" ";  cout<<"b="<<b<<endl;  return 0; } | a=5 b=6 | a=5 b=7 | 5 7 | 编译出错 | D |
| 13 | 下面的MyClass类中声明了"+"运算符重载函数的原型，如果在主函数中有如下定义 MyClass a , b ; 那么执行语句 a+b ;时，编译器将a+b解释为( )。 | operator + (a,b) | b.operator + (b) | a.operator+b | a.operator + (b) | D |
| 14 | 下列说法中，错误的是( ) | 每个函数都有一个函数头和一个函数体，主函数也不例外 | 程序是由若干个函数组成的，但是必须有、而且只能有一个主函数 | 主函数只能调用用户函数或系统函数,用户函数可以相互调用 | 每个语句必须独占一行，语句的最后可以是一个分号，也可以是一个回车换行符号 | D |
| 15 | C++的合法注释是( ) | // This is a C program | "This is a C program" | /\*This is a C program/\* | /This is a C program/ | A |
| 16 | 当创建一个类对象时，系统自动调用\_\_\_\_\_并给予对象的数据成员进行初始化。 | 静态函数 | 析构函数 | 构造函数 | 友元函数 | C |
| 17 | 以下不正确的是( ) | for（；；）表示无限循环 | for（）表示无限循环 | 语句for（i=0；；i++）表示无限循环 | while（1）表示无限循环 | B |
| 18 | 在C和C++语言中，换行符是\_\_\_\_\_。 | '\n' | '\t' | '\v' | '\b' | A |
| 19 | 以下正确的函数原型(函数声明）是( ) | void f1(int ,int); | void f1(x,y); | void f1(int x,y); | f1(int x;int y); | A |
| 20 | 一个C++程序的执行是从（ ） | 本程序文件的第一个函数开始,到本程序main函数结束 | 本程序的main函数开始,到本程序文件的最后一个函数结束 | 本程序文件的第一个函数开始,到本程序文件的最后一个函数结束 | 本程序的main函数开始,到main函数结束 | D |
| 21 | 如果说明语句char c='\072';则变量c( ) | 包含1个字符 | 包含3个字符 | 说明不合法，c的值不确定 | 包含2个字符 | A |
| 22 | 在函数定义中的形参属于( ) | 静态变量 | 局部变量 | 全局变量 | 寄存器变量 | B |
| 23 | char x[]="abcdefg";  char y[]={'a','b','c','d','e','f','g'}; 对于以下叙述正确的是( ) | 数组x和数组y的长度相同 |  | 数组x和数组y等价 | 数组x的长度小于数组y的长度 | 数组x的长度大于数组y的长度 |
| 24 | ( )能正确输出结果：C++ | char s[3]={'C','+','+','\0'};cout<<s<<endl; | char s[3]={'C','+','+'};cout<<s<<endl; | char s[]="C++";cout<<s<<endl; | char s[3]="C++";cout<<s<<endl; | C |
| 25 | 已知int a=1,b=2,c=3;以下语句执行后a,b,c的值是( ) if(a>b) c=a; a=b; b=c; | a=2, b=3, c=1 | a=1, b=2, c=3 | a=2, b=3, c=3 | a=2, b=3, c=2 | C |
| 26 | 以下选项中合法的用户标识符是( ) | \_2Test | long | A.dat | 3Dmax | A |
| 27 | 若有定义 int a=5;则执行完语句a+=a-=a\*a之后， a的值（ ） | -40 | 3 | -12 | 0 | A |
| 28 | 正确表示"x是大于5小于10的数"的表达式是（ ） | x>5|x<10 | x>5||x<10 | 5<x<10 | x>5&&x<10 | D |
| 29 | 执行下面的程序段后，变量ｋ中的值为( ) int k=3,s[2]; s[0]=k; k=s[1]\*10; | 不定值 | 33 | 30 | 10 | A |
| 30 | 在C++中,如果函数没有返回值，那么应该声明为（　　）类型 | char | double | void | int | C |
| 31 | 已知int x=50,y=80,z=90; 则执行语句if(x>y) z=x; x=y; y=z; 后，x, y, z的值是\_\_\_\_\_。 | 80,90,90 | 80,90,80 | 50,80,90 | 80,90,50 | A |
| 32 | 下面程序运行结果是( ) #include<iostream> using namespace std; int main() {  int a,b; for(a=1,b=1;a<=100;a++) { if(b>=20)break;  if(b%3!=1)  b-=5; else  b+=3;   } cout<<a<<endl;  return 0; } | 7 | 8 | 9 | 10 | B |
| 33 | 只能作用于整型数据的运算符是( )。 | - | % | \* | / | B |
| 34 | 下列关于类的权限的描述错误的是( )。 | 类的对象只能访问该类的公有成员 | 一个类可以将另一个类的对象作为成员 | 普通函数不能直接访问类的公有成员，必须通过对象访问 | 类本身的成员函数只能访问自身的私有成员 | D |
| 35 | 设int m=7，n=12；则表达式结果为3的是（ ） | n%=(m-m%5) | n%=m-m%5 | (n%=m)-(m%=5) | n%=(m%=5) | C |
| 36 | 对类中声明的变量，下列描述中正确的是( )。 | 任何情况下都可被该类所有实例共享 | 属于全局变量 | 只有类的公有成员可以使用 | 类中所有成员函数都可以使用 | D |
| 37 | 一个类可包含析构函数的个数是( )。 | 1个 | 0个 | 至少一个 | 0个或多个 | A |
| 38 | int n=1;  while（n--）  n++;  那么while循环执行次数是( ) | 2 | 1 | 0 | 无穷次 | D |
| 39 | 下列关于类和对象的叙述中，错误的是\_\_\_\_\_。 | 对象是类的抽象 | 对象是类的具体实例 | 类是对某一类对象的抽象 | 类和对象的关系是一种数据类型和变量的关系 | A |
| 40 | 设x和y均为bool变量，则表达式x&&y结果为真的条件是（ ） | 其中一个为假 | 它们均为假 | 其中一个为真 | 它们均为真 | D |
| 41 | 下面程序输出结果是（ ） #include<iostream> using namespace std; int main() {  int a=100,x=10,y=20,ok1=5,ok2=0;  if(x<y)  if(y!=10)  if(!ok1)   a=1;  else if(ok2)  a=10;   a=-1;  cout<<a<<endl;  return 0; } | 1 | 值不确定 | 0 | -1 | D |
| 42 | 若有说明:int a[10]; 则对a数组元素的正确引用是( ) | a[10] | a[3.5] | a(5) | a[10-10] | D |
| 43 | #include <iostream> using namespace std; int main() {  int a=1,b=2,c=2,t;   while(a<b<c)  { t=a;a=b;b=t;c--;}  cout<<a<<","<<b<<","<<c;  return 0; } 执行上述程序，其运行结果是( ) | 2，1，1 | 1，2，1 | 1，2，0 | 2，1，0 | C |
| 44 | 以下对一维数组a的正确说明是（　　） | int a(10); | int n; cin>>n; int a[n]; | int n=10,a[n]; | const int n=10; int a[n]; | D |
| 45 | 假设A是一个类，下面语句“A a1[2]; A &pa = a1;”执行时，系统会调用\_\_\_\_\_次A类的构造函数。 | 0 | 3 | 2 | 1 | C |
| 46 | 下面程序运行结果是( ) #include <iostream> using namespace std; int main() {  int x=0,y=0;   while(x<15)  {  y++;  x+=y;  }  cout<<y<<","<<x;  return 0; } 。 | 5,7 | 20,8 | 6,12 | 5,15 | D |
| 47 | 设存在函数int max(int，int)返回两参数中较大值，若求22，59，70三者中最大值，下列表达式不正确的是( )。 | int m = max(59，max(22，70))； | int m = max(max(22，59)，70)； | int m = max(22，59，70)； | int m = max(22，max(59，70))； | C |
| 48 | 若以下说明： int a[12]={1,2,3,4,5,6,7,8,9,10,11,12}; char c='a',d,g; 则数值为４的表达式是（ ）。 | a[g-c] | a[4] | a['d'-'c'] | a['d'-c] | D |
| 49 | int x=-1; do  {  x=x\*x; } while(!x); 下列说法正确的是( ) | 是死循环 | 有语法错误 | 循环执行一次 | 循环执行两次 | C |
| 50 | 下面程序运行结果是( ) #include<iostream>  using namespace std; int main() {  int x=3;  do  {  x-=2;  cout<<x<<＂#＂;  }while(!(--x));  return 0; } | 1#-2# | 死循环 | 3#0# | 1 | A |
| 51 | 对于有返回值的函数，要结束函数运行必须使用语句（ ） | return | main | break | continue | A |
| 52 | 当一个类对象生命期结束时，系统自动调用\_\_\_\_\_并销毁对象。 | 析构函数 | 静态函数 | 友元函数 | 构造函数 | A |
| 53 | 假定一个字符串的长度为n，则定义存储该字符串的字符数组的长度至少为( ) | n | n+1 | n-1 | n+2 | B |
| 54 | 已知类定义如下： class Test{ public:  void Set(double val); private:  double value; }; 则类体外对Set成员函数的定义正确的是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。 | void Set(double val) {value=val;} | void Test::Set(double val) {val=value;} | Test::void Set(double val) {value=val;} | void Test::Set(double val) {value=val;} | D |
| 55 | 设d为字符变量，下列表达式中不正确的是\_\_\_\_\_。 | d=50 | d='n' | d='\n' | d="\n" | D |
| 56 | 如果没有为一个类定义任何构造函数的情况下，下列描述正确的是( )。 | 这个类不需要构造函数 | 编译器总是自动创建一个不带参数的构造函数 | 这个类没有构造函数 | 该类不能通过编译 | B |
| 57 | 下列对构造函数的描述中，( )是错误的。 | 函数名与类名完全相同 | 构造函数允许重载 | 构造函数允许使用缺省参数 | 返回类型应该为void | D |
| 58 | class T { int a; public: void setValue(int aa) { a=aa;} }; int main() { T t,\*p=&t; ( ) return 0; } 在上段代码的空中填入的合法语句是( )。 | t.a=5; | p->setValue(5); | p.setValue(5); | t->setValue(5); | B |
| 59 | 下列关于析构函数的描述中，( )是正确的。 | 析构函数可以有1个或多个参数 | 一个类中可以重载多个版本析构函数 | 析构函数不能指定返回值类型 | 析构函数名与类名完全相同 | C |
| 60 | 下列程序的运行结果为（ ） #include<iostream>  using namespace std; int main() {   int a=2;   int b=a+1;   cout<<a/b<<endl;  return 0; } | 0.666666666... | 0.66667 | 0 | 0.7 | C |
| 61 | 有关类性质说法错误的是\_\_\_\_\_。 | 析构函数和构造函数都不能有返回类型 | 一个类可以有多个构造函数，但只有一个析构函数 | 一个类中不可以声明具有类类型的数据成员 | 不能给析构函数指定参数 | C |
| 62 | 如下代码，对其描述正确的是( )。int \*p;p=new int(5); | 动态开辟5个int元素，初值不确定 | 存在语法错误 | 动态开辟5个int元素，初值为0 | 动态开辟初值为5的一个int元素 | D |
| 63 | 下列标识符中，( )是非法的。 | SIZE\_MB | 2002YEAR | YEAR2002 | \_SIZE\_MB | B |
| 64 | 下面不正确的字符串常量是( ) | ‘rst’ | “14’14” | “” | “0” | A |
| 65 | 对while 和do...while循环结构说法正确的是( ) | 只是表达形式不同 | 条件成立时，它们有可能一次也不执行 | do...while结构中的语句至少执行一次 | while结构中的语句至少执行一次 | C |
| 66 | 能用作C++变量名的是( )。 | 123abc | const | switch | \_1ab | D |
| 67 | 已知类声明如下“class A{ int a;};”，则类A的数据成员a的访问权限是\_\_\_\_\_。 | private | public | 不确定 | protected | A |
| 68 | class Fraction { int num,den; public: Fraction(int n=0.int d=1); ...... }; int main() { Fraction f1(3,5),f2(4),f3; f3=f1+f2; cout<<f3<<endl; return 0; } 为完成主函数的功能，类中还需添加( )函数声明。 | Fraction operator+(Fraction& f2); friend ostream& operator<<(ostream& out,const Fraction& f); | friend ostream& operator<<(ostream& out,const Fraction& f); | Fraction(); Fraction operator+(Fraction& f2); friend ostream& operator<<(ostream& out,const Fraction& f); | Fraction operator+(Fraction& f2); | A |
| 69 | 在C++语言中,非法的字符常量是( ) | “\n” | ‘\xaa’ | ‘\017’ | ‘\t’ | A |
| 70 | 判断char型变量n是否为小写字母的正确表达式为（ ） | ‘a’<=n<=‘z’ | (n>=‘a’)&&(n<=‘z’) | (‘a’>=n）| |(‘z’<=n) | （n>=a）&&（n<=z） | B |
| 71 | 若用数组名作为调用函数的实参，则传递给形参的是( ) | 数组元素的个数 | 数组的第一个元素值 | 数组的首地址 | 数组中全部的元素值 | C |
| 72 | 以下将一维数组ｍ正确初始化为全零的语句是（ ）。 | int m[10]=(0,0,0,0); | int m[10]={}; | int m[]={0}; | int m[10]={10\*2}; | C |
| 73 | 在循环语句的循环体中使用break语句的作用是( ) | 结束开关语句 | 结束程序执行 | 结束本次循环 | 结束该层的循环 | D |
| 74 | 求平方根函数的函数名为( ) | sqrt | pow | abs | cos | A |
| 75 | 若一个函数无返回值，则定义它时函数的类型应该为( ) | void | 任意 | 无 | int | A |
| 76 | C++中，系统自动为一个类生成默认构造函数的条件是该类没有自定义任何\_\_\_\_\_。 | 构造函数 | 有参构造函数 | 无参构造函数 | 成员函数 | A |
| 77 | 关键字( )定义允许在类外部使用的成员数据和成员函数。 | private | protected | public | virtual | C |
| 78 | 在C++语言中，换行符是（ ） | '\b' | '\t' | '\n' | '\f' | C |
| 79 | 已知类定义如下： class Test{ public:  Test(int a,int b,int c):x(c),z(a),y(b) { } private:  int x,y,z; }; 则Test类中3个数据成员的初始化顺序是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。 | z,x,y | x,y,z | x,z,y | z,y,x | B |
| 80 | main()函数的返回类型是( )。 | float | 任意类型 | 不返回值 | int | D |
| 81 | 下面有关for循环的正确描述是( ) | 在for循环中，不能用break语句跳出循环体 | for循环的循环体语句中，可以包含多条语句，但必须 用大括号括起来 | for循环只能用于循环次数已经确定的情况 | for循环是先执行循环体语句，后判断表达式 | B |
| 82 | 假定MyClass为一个类，执行"MyClass a[3], \*p;"语句时会自动调用该类型构造函数( )次。 | 4 | 5 | 2 | 3 | D |
| 83 | 有以下程序 #include<iostream> using namespace std; int fun(int x,int y) { return(x+y); } int main()  {  int a=2,b=5,c=8;  cout<<(fun((int)fun(a+c,b),a-c))<<endl;  return 0;  } 程序运行后的输出结果是（ ） | 9 | 编译出错 | 21 | 9.0 | A |
| 84 | 下列for循环的次数为（ ） for（i=0, x=0; !x&&i<=5; i++）; | 6 | 5 | 1 | 无限 | A |
| 85 | 下面正确的字符常量是( ) | 'W' | '\\'' | "C" | "\0" | A |
| 86 | 如没有使用private关键字定义类的数据成员，则默认为( )。 | private | protected | public | friend | A |
| 87 | 如下代码，对其描述正确的是( )。int \*p;p=new int[5]; | 动态创建了5个int数据所需连续内存空间，初值不确定 | 动态创建5个int数据所需的连续内存空间，初值为0 | 动态创建一个int所需内存空间，且初值为5 | 存在语法错误 | A |
| 88 | 现有定义int a;double b;floatc;char k;,则表达式a/b+c-k值的类型为:. | float | double | int | char | B |
| 89 | 以下数据中,不正确的数值或字符常量是(). | o13 | 9861 | 5L | 0 | A |
| 90 | 已知char a;int b;float c;double d;则表达式a-b+c-d结果为()型. | float | double | int | char | B |
| 91 | 以下语句中,:当字符串s1和s2相等时,能够输出"OK"的是() | if(s1==s2)printf("OK"); | if(s1[0]==s2[0])printf("OK"); | if(strcmp(s1,s2))puts("OK"); | if(!strcmp(s1,s2))puts("OK"); | D |
| 92 | 下列选项中不符合良好程序设计风格的是(). | 数据说明的次序要规范化 | 对所有输入的数据都要进行检查,确保数据的合法性 | 程序的效率第一,清晰第二 | 程序中要有必要的注释 | C |
| 93 | 关于字符常量,以下叙述正确的是(). | 所有的字符常量都可以作为整型量来处理 | 单引号中的大写字母和小写字母代表的是相同的字符常量 | 字符常量能包含大于一个的字符 | 空格不是一个字符常量 | A |
| 94 | 以下选项中,非法的字符常量是(). | 0' | \xAA' | \\' | \0201' | D |
| 95 | 下列关于类,对象,属性和方法的叙述中错误的是(). | 方法用于表示对象的行为 | 属性用于描述对象的状态 | 基于同一个类产生的两个对象不可以分别设置自己的属性值 | 类是对一类具有相同的属性和方法对象的描述 | C |
| 96 | 可在C++程序中,用作用户标识符的一组标识符是(). | 2c DO SIG | void define WORD | as\_b3 \_123 If | for -abc case | C |
| 97 | 下述错误的C++语言常量是(). | 5. | .25 | 0xf | 090 | D |
| 98 | 下列字符序列中是C++语言保留字的是(). | scanf | sizeof | sqrt | include | B |
| 99 | 以下不能正确进行字符串赋初值的语句是(). | char \*str="good!"; | char str[5]={'g','o','o','d',0}; | char str[5]="good!"; | char str[]="good!"; | C |
| 100 | 字符串"\\\'efg\'\\"的长度是:(). | 7 | 3 | 11 | 5 | A |
| 101 | 以下能正确进行字符串赋值的语句是 | char s[5];s="good!"; | char s[5]={'a','e','i','o','u'}; | char s[5]="good!"; | char \*s;s="good!"; | D |
| 102 | 可在C++程序中,用做用户标识符的一组标识符是(). | and \_2007 | case Bigl | Date y-m-d | Hi Dr.Tom | A |
| 103 | C++语言中的标识符只能由字母,数字和下划线三种字符组成,且第一个字符(). | 可以是字母,数字和下划线中,任一字符 | 必须为字母 | 必须为下划线 | 必须为字母或下划线 | D |
| 104 | 下列语句中符合C++语言语法的赋值语句是() | x=7+y,y++,x++ | x=7+y++=x+7; | x=7+y+z=x+z; | x=7+y,z=x+7; | D |
| 105 | 以下不符合C语言语法的赋值语句是(). | ++j; | y=(a=3,6\*5); | a=1,b=2 | a=b=5; | C |
| 106 | 字符串"\nabcdef\\"的长度是(). | 8 | 6 | 10 | 4 | A |
| 107 | 以下字符中,是转义字符的是(). | \b' | \a' | '\\' | \c' | C |
| 108 | 下列语句中符合C++语言语法的赋值语句是:(). | a=7+b,c=a+7; | a=7+b,b++,a+7 | a=7+b+c=a+7; | a=7+b++=a+7; | A |
| 109 | C++语言中,以下几种运算符的优先次序()的排列是正确的. | 由高到低为:算术运算符,关系运算符,!,&&,||,赋值运算符 | 由高到低为:算术运算符,关系运算符,赋值运算符,!,&&,|| | 由高到低为:!,算术运算符,关系运算符,&&,||,赋值运算符 | 由高到低为:!,&&,||,算术运算符,赋值运算符 | C |
| 110 | 在C++语言中,运算对象必须是整型数的运算符是(). | %和\ | \ | \*\* | % | D |
| 111 | 以下语句中,符合C++语言语法的赋值语句是(). | m+n=10; | a=10+b,b+=a+10; | a=7+b,c=a+7 | m=n | B |
| 112 | 设a和b均为 double型常量,且a=5.5,b=2.5,则表达式(int)a+b/b的值是(). | 6 | 5.500000 | 6.000000 | 6.500000 | C |
| 113 | 下列语句中符合C++语言语法的赋值语句是(). | i=7+1,i++,1+4 | i=1+j+k=i+2; | 1=1+i++=i+5; | i=1+2,k=j+2; | D |
| 114 | 对象实现了数据和操作的结合,是指对数据和数据的操作进行. | 隐藏 | 集成 | 封装 | 组合 | C |
| 115 | 字母A的ASCⅡ码为十进制数65,且C2为字符型,则执行语句C2='A'+'6'-'3';后,C2为()字符. | C | D | 不确定的值 | 69 | B |
| 116 | 下列语句中符合C++语言语法的语句是( ). | x=7+y,y++,z++; | x=y+2=x+y+z; | x=3+y++=x+3; | x=(3+b,z)=x+3; | A |
| 117 | 以下字符中,不是转义字符的是(). | '\b' | '\\' | '\c' | '\a' | C |
| 118 | 下列可用于C++语言用户标识符的一组是(). | a33,l12,Car | for,-abc,IF case | void,define,WORD | 2a,DO,size of | A |
| 119 | 判断两个字符串是否相等,正确的表达方式是(). | while(strcmp(s1,s2)=0) | while(strcmp(s1,s2)==0) | while(s1==s2) | while(s1=s2) | B |
| 120 | 下列选项中可作为C++语言合法常量的是(). | -80.0e | -80 | -8e1.0 | -80a | B |
| 121 | 以下符合C++语言语法的实型常量是(). | 8E-2 | 2.2E0.5 | 3.14.159E | E14 | A |
| 122 | 设x和y都是int类型,且x=1,y=2,则printf("%d%d",x,y,(x,y))的输出结果是(). | 1,2 | 1 2 2 | 输出值不确定 | 12 | D |
| 123 | 下面正确的字符常量是(). | '\9' | 'W' | "c" | '\' | B |
| 124 | 以下选项中,合法的一组C++语言数值常量是(). | 028.5e-3 -0xf | 0x8A 10,000 3.e5 | 12 OXa23 4.5e0 | .177 4c1.5 Oabc | C |
| 125 | 已知字母A的ASCII码为十进制数65,且c为字符型,则执行语句c='A'+'6'-'3';后,c中的值为:(). | 不确定的值 | D | d | c | B |
| 126 | C++语言中,运算对象必须是整型的运算符是(). | 〈= | %= | = | / | B |
| 127 | 以下数据中,属于字符串常量的是(). | 'ask' | 'm' | "ASK" | ASK | C |
| 128 | 以下选项中,可以作为C++语言中,合法整数的是(). | afbc | 110110B | 0Xbbc | 01988 | C |
| 129 | 下列字符序列中可用作C++标识符的一组字符序列是(). | S.b,sum,average,\_above | D56,r\_1\_2,name,\_st\_1 | class,day,lotus\_1,2day | #md,&12x,month,student\_n! | B |
| 130 | 以下数值中,不正确的整型常量是(). | 0xaaaa | 0x16 | 078 | 16543 | C |
| 131 | 判断字符串s1是否大于字符串s2,应该使用(). | if(strcmp(s1,s2)<0) | if(strcmp(s2,s1)<0) | if(s1>s2) | if(strcmp(s1,s2)) | B |
| 132 | 下列标识符中不合法的C++语言用户自定义标识符是(). | sin | \_ | enum | printf | C |
| 133 | 以下程序段中不能正确赋字符串(编译时系统会提示错误)的是(). | char t[]="abcdefg",\*s=t; | char s[10];s="abcdefg"; | char s[10];strcpy(s,"abcdefg"); | char s[10]="abcdefg"; | B |
| 134 | 下面各选项中均是C++语言合法标识符的选项组是(). | 33 we auto | \_43 3e\_else | 23\_me\_3ew | ER -DF 32 | C |
| 135 | 若已定义x和y为 double 类型,则表达式x=1,y=x+3/2 的值是(). | 1 | 2 | 2.0 | 2.5 | C |
| 136 | C++语言中,char 类型数据占(). | 2个字节 | 4个字节 | 8个字节 | 1个字节 | D |
| 137 | 以下不能将s所指字符串正确复制到t所指存储空间的是( ) | while(\*t=\*s){t＋＋；s＋＋} | for(i=0；t[i]=s[i]；i＋＋)； | do{\*t＋＋=\*s＋＋；}while(\*s)； | for(i=0,j=0；t[i＋＋]=s[j＋＋]；)； | C |
| 138 | 下述正确的C++语言常量是(). | E2 | 3e-3 | 1.5E2.5 | 5.0E | B |
| 139 | 下面各语句行中能正确进行赋值字符串操作的是(). | char s[5]={'a','b','c','d','e'}; | char s[5];scanf("%s",&s); | char \*s;s="ABCDEF"; | char \*s;gets(s); | D |
| 140 | 下述语句中在字符串s1和s2相等时显示"they are Equal"的是(). | if(s1==s2)puts("they are Equal"); | if(\*s1==\*s2)puts("they are Equal"); | if(!strcmp(s1,s2))puts("they are Equal"); | if(strcmp(s1,s2))puts("they are Equal"); | C |
| 141 | 以下数据中,不正确的数值或字符常量是(). | 8.9e1.2 | 82.5 | 10 | 0xff00 | A |
| 142 | 下列能正确定义字符串的语句是(). | char str="kx43"; | char str[]={'\064'}; | char str="; | char str[]="\0"; | D |
| 143 | 以下选项中,不能作为合法常量的是(). | 3.14e0.5 | 3.14e05 | 3.14e0 | 3.14e+5 | A |
| 144 | 有以下程序: #include<iostream> using namespace std; int main(){ int x; cin>>x; if(x<=3); else   if(x!=10)cout<<x; } 程序运行时,输入的值在哪个范围才会有输出结果. | 大于3且不等10的整数 | 小于3的整数 | 大于3或等于10的整数 | 不等于10的整数 | A |
| 145 | 当把4个表达式用作if语句的控制表达式时,有一个选项与其他3个选项含义不同,这个选项是(). | !k%2==1 | (k%2)!=0 | k%2==1 | k%2 | A |
| 146 | 两次运行下面的程序,如果从键盘上分别输入3和2,则输出结果是().    #include<iostream>  using namespace std; main() { int x;  cin>>x  if(x++>2)cout<<x;  else cout<<x--;  } | 4和3 | 4和2 | 3和2 | 4和1 | A |
| 147 | 当a=6,b=14,c=8,d=4时,执行下面程序段后,x的值是().    if(a<b) if(c<d)x=4  else  if(a<c) if(b<d)x=6;  else x=8;  else x=14;  else x=27; | 6 | 27 | 4 | 8 | D |
| 148 | 以下选项中,能用作用户标识符的是(). | unsigned | void | o | 8\_8 | C |
| 149 | 设有条件表达式:(EXP)?i++:j--,则以下表达式中(EXP)完全等价的是(). | (EXP!=0) | (EXP!=1) | (EXP==1) | (EXP==0) | A |
| 150 | 以下程序的执行结果是(). main() {int x=2,y=2,z=0; if(z<0) if(y>0)x=4; else x=5; printf("%d\t",x); if(z=y>0)x=7; else if(y=0)x=3; else x=6; printf("%d\t",x); printf("%d\t",z); } | 520 | 271 | 345 | 205 | B |
| 151 | 若有条件表达式exp?a++:b--,则以下表达式中能完全等价于表达式exp的是(). | (exp==1) | (exp!=0) | (exp==0) | (exp!=1) | B |
| 152 | 有如下程序: #include<iostream> using namespace std; main() { int x=1,a=0,b=0; switch(x) { case 0:b++; case 1:a++; case 2:a++;b++; } cout<<"a="<<a<<",b="<<b; } 该程序的输出结果是 | a=1,b=1 | a=2,b=2 | a=2,b=1 | a=1,b=0 | C |
| 153 | 在C++语言中,if语句后的一对原括号中,用以决定分支的流程的表达式(). | 只能用逻辑表达式或关系表达式 | 只能用逻辑表达式 | 可用任意表达式 | 只能用关系表达式 | C |
| 154 | 为表示关系x≥y≥z,应使用的C++语言表达式是(). | (x>=y)or(y>=z) | (x>=y>=z) | (x>=y)AND(y>=z) | (x>=y)&&(y>=z) | D |
| 155 | 下面标识符中合法的用户标识符为() | return | \_float | -int | (int)(1.6) | B |
| 156 | int a=1,b=2,c=3; if(a>b)a=b; if(a>c)a=c; ,则a的值为(). | 1 | 3 | 不一定 | 2 | A |
| 157 | 若有int x,y,z;,则执行以下程序段后的输出结果为: x=1;y=5;z=3; if(x>y)x=y; y=z; cout<<"x="<<x<<",y="<<y<<",z="<<z; | x=1,y=5,z=3 | x=1,y=3,z=3 | x=5,y=3,z=5 | x=5,y=3,z=3 | B |
| 158 | 若有定义:int a=9;,则执行语句a+=a -=a+a;后,a的结果是(). | 18 | 9 | -18 | -9 | C |
| 159 | 设有定义:int m=0;.以下选项的四个表达式中与其他三个表达式的值不相同的是(). | m++ | ++m | m+1 | m+=1 | A |
| 160 | 能正确表示逻辑关系"n≥5或n≤2"的C++语言表达式是() | n>=5 or n<=2 | n>=5||n<=2 | n>=5&&n<=2 | n>=2&n<=5 | B |
| 161 | 若k是整型,则以下程序段的执行结果是:() k=-3; if(k<=0)cout<<"####"; else cout<<"&&&&"; | &&&& | 有语法错误,无结果 | ####&&&& | #### | B |
| 162 | 正确的标识符是(). | x=8 | !at | z.2 | y2 | D |
| 163 | 若有int a=1,b=2,c=3;,则执行下列语句后a的值为(). if(a>b)a=b; if(a>c)a=c; | 2 | 1 | 不一定 | 3 | B |
| 164 | 下列程序的运行结果是(). main() { int a=2,b=3,c=4; if(a<b) if(b<0)c=0; else c+=1; cout<<c; } | 2 | 5 | 4 | 3 | B |
| 165 | 当a=4,b=5,c=7,d=6时,执行下面一段程序:   if(a<b) if(c<d)x=1;  else  if(a<c) if(b<c)x=2;  else x=3;  else x=4;  else x=5;    程序执行后,x的值为(). | 3 | 4 | 1 | 2 | D |
| 166 | 以下程序的运行结果是().   main() { int a=-5,b=1,c=1;  int x=0,y=2,z=0;  if(c>0)x=x+y;  if(a<=0) { if(b>0) if(c<=0)y=x - y;  } else if(c>0)y=x - y;  else z=y;  cout<<x<<","<<y<<","<<z;  } | 2,2,2 | 2,0,2 | 2,2,0 | 0,2,0 | C |
| 167 | C++语言中,switch后的括号内表达式的值可以是 | 只能为整型,字符型,枚举型 | 只能为整型 | 任何类型 | 只能为整型和字符型 | A |
| 168 | 下列不合法的用户标识符是(). | 4d | \_8\_ | Double | j2\_KEY | A |
| 169 | 当a=1,b=2,c=3,d=4时,执行下面程序段后,x的值是().  if(a<b) if(c<d)x=1;  else  if(a<c) if(b<d)x=2;  else x=3;  else x=6;  else x=7; | 3 | 1 | 6 | 2 | B |
| 170 | 以下标识符不是关键字的是(). | return | break | char | Switch | D |
| 171 | 有以下程序: #include<iostream> using namespace std; int main() { int x=1,y=0;  if(!x)y++;  else if(x==0)  if(x)y+=2;  else y+=3;  cout<<y; } 程序运行后的输出结果是(). | 1 | 3 | 0 | 2 | C |
| 172 | 已知int k=67;,则表达式k/10的值为; | 6.7 | 6.70 | 6 | 7 | C |
| 173 | 若有a=45,b=27,c=0;,则执行c=a>b?a:b后c的结果是(). | 72 | 27 | 18 | 45 | D |
| 174 | 有以下程序 main() {int a=0,b=0,c=0,d=0;  if(a=1)  b=1;  c=2;  else   d=3; cout<<a<<","<<b<<","<<c<<","<<d; } 程序输出结果是(). | 0,1,2,0 | 0,0,0,3 | 1,1,2,0 | 编译有错 | C |
| 175 | 下列运算符优先级最高的是(). | && | || | = | + | D |
| 176 | 以下标识符中不能作为合法的C++用户定义标识符的是(). | to | \_if | answer | signed | D |
| 177 | 下列运算符中优先级别最低的运算符是 | <= | != | + | || | D |
| 178 | 以下程序的运行结果是 main() { int a=2,b=-1,c=2 ;  if(a<b)  if(b<0)c=0;  else c+=1;  cout<<c; } | 1 | 0 | 2 | 3 | C |
| 179 | 以下是if语句的基本形式: if (表达式) 语句 其中,"表达式". | 可以是任意合法的表达式 | 必须是逻辑表达式或关系表达式 | 必须是逻辑表达式 | 必须是关系表达式 | A |
| 180 | 有以下程序:  int main() { int x=1,y=0,a=0,b=0;  switch(x)  { case 1:  switch(y)  { case 0:a++;break;  case 1:b++;break;  }  case 2:a++;b++;break;  case 3:a++;b++;  }  cout<<"a="<<a<<",b="<<b; } 程序的运行结果是(). | a=2,b=1 | a=2,b=2 | a=1,b=1 | a=1,b=0 | A |
| 181 | 以下程序的输出结果是(). int main() { int a=8,b=6,m=1;  switch(a%4)  { case 0:m++;break;  case 1:m++;  switch(b%3)  { default:m++;  case 0:m++;break;  }  }  cout<<m; } | 1 | 2 | 4 | 3 | B |
| 182 | 设有如下程序段: int k=10; while(k==0) k--; ,则下述说明中正确的是(). | 循环体执行一次 | 死循环 | 循环体一次也不执行 | 循环体执行10次 | C |
| 183 | 以下不构成无限循环的语句或语句组是(). | n=0；  do{＋＋n；}while(n<=0)； | for(n=0,i=1； ； i＋＋)n＋=i； | n=10； while(n)；{n--；} | n=0； while(1){n＋＋；} | A |
| 184 | 若有int y=10;,则执行下列语句后的输出为(). while(y--); printf("y=%d\n",y); | y=-1 | while 构成无限循环 | y=1 | y=0 | A |
| 185 | 下列程序执行输出的结果是().   f(int a) { int b=0;  static c=3;  a=c++;b++;  return(a);  } main() { int a=2,i,k;  for(i=0;i<2;i++)  k=f(a++);   cout<<k;  } | 3 | 4 | 5 | 6 | B |
| 186 | 以下程序的执行结果是(). main() {int num=0; while(num<=2){num++;cout<<num<<",";}} | 1,2,3, | 0,1,2 | 1,2, | 1,2,3,4, | A |
| 187 | 针对for 循环语句,for(表达式1;表达式2;表达式3),下列说法正确的是(). | break语句只能跳出最内层循环 | break语句可以一次性跳出多层嵌套循环 | continue语句不能跳出最外层循环 | continue语句可以一次性跳出多层嵌套循环 | A |
| 188 | 下面程序的运行结果是().    #include<iostream> using namespace std; main()   { int i;  for(i=1;i<=5;i++) { if(i%2)  cout<<"$";  else   cout<<"@";  }   } | @$@$@ | $@$@ | @$@$ | $@$@$ | D |
| 189 | 对 for(表达式1;;表达式3)可理解为(). | for(表达式1;表达式3;表达式3) | for(表达式1;0;表达式3) | for(表达式1;表达式1;表达式3) | for(表达式1;1;表达式3) | D |
| 190 | 下述循环的循环次数是(). int k=2; while(k=0)  ;k--; | 1次 | 0次 | 无限次 | 2次 | B |
| 191 | 下面程序是从键盘输入4位正整数,输入0或负数时结束循环.A处填写内容是().  main() {  int num;  do{  cin>>num;  }while(\_\_\_A\_\_\_); } | num>0 | !num | num==0 | !num !=0 | A |
| 192 | 若有int i=0;,则以下程序段中,while 循环的循环次数是() while(i<10) { if(i<1)continue; if(i==5)break; i++; } | 6 | 死循环,不能确定次数 | 1 | 4 | B |
| 193 | 阅读下列程序,则运行结果为(). #include<iostream> using namespace std; fun() { static int x=3;  x++;  return x; } int main() { int i,x;  for(i=0;i<3;i++)  x=fun();  cout<<x; } | 5 | 3 | 6 | 4 | C |
| 194 | 若有int mm;,以下程序段执行后输出结果是(). for(mm=3;;mm=mm-1); cout<<mm; | 3 | 死循环 | 2 | 1 | B |
| 195 | 以下程序的功能是计算:s=1+12+123+1234+12345 int main(void) { int t=0,s=0,i; for(i=1;i<=5;i++)  { t=i+\_\_\_\_\_\_;  s=s+t;  } cout<<"s="<<s; } | t | t\*100 | t+10 | t\*10 | D |
| 196 | 在下列选项中,没有构成死循环的是(). | for(;;); | int i=100; while(1){i=i+1; } | int s=36; while(s)--s; | int k=10000; do{k++;}while(k>10000); | C |
| 197 | 以下程序的输出结果是(). double f(int n) { int i;double s;  s=1.0;  for(i=1;i<=n;i++)  s+=1.0/i;  return s; } int main() { int i,m=3;float a=0.0;  for(i=0;i<m;i++)  a+=f(i); cout<<a; } | 5.500000 | 8.25 | 4.000000 | 3.000000 | A |
| 198 | t为int类型,进人下面的循环之前,t的值为0 while(t=1) {……} ,则以下叙述中,正确的是(). | 循环控制表达式不合法 | 循环控制表达式的值为1 | 其他说法都不对 | 循环控制表达式的值为0 | B |
| 199 | 若有int w=12;,对以下程序段,描述正确的是(). while(w=0)w=w-1; | 循环是无限循环 | 循环体语句一次也不执行 | 循环体执行了10次 | 循环体语句执行了一次 | B |
| 200 | 有以下程序: int main() { int i,j,m=55; for(i=1;i<=3;i++) for(j=3;j<=i;j++)m=m%j; cout<<m; } 程序的运行结果是(). | 3 | 2 | 1 | 0 | C |
| 201 | 有以下程序 int main() { int y=9;  for(;y>0;y--)  if(y%3==0)  cout<<--y; } 程序的运行结果是(). | 963 | 741 | 875421 | 852 | D |
| 202 | 有以下程序  main() { int y=9;  for(; y>0;y--) if(y%3==0)cout<<--y;  }   程序的运行结果是(). | 875421 | 741 | 963 | 852 | D |
| 203 | 以下语句的执行结果是(). int m=0,s=1; do{ s+=m++; }while(m<6); cout<<s; | 17 | 18 | 15 | 16 | D |
| 204 | 从键盘输入的整数中,找出最小值min,输入0时结束.请在A处填写正确语句(). main() { int min=0,a; do{ cin>>a; if(min>a)min=a; }while(\_\_\_A\_\_\_); cout<<min; } | a !=0 | a==0 | a>0&&a<0 | a=0 | A |
| 205 | 以下语句中,循环次数不为10次的语句是(). | i=1;m:if(i<=10){i++;goto m;} | i=1;do{i++;}while(i<=10); | i=10;while(i>0){--i;} | for(i=1;i<10;i++); | D |
| 206 | 若有说明int a[3][4];则a数组元素的非法引用是(). | a[4-2][0] | a[1][3] | a[0][4] | a[0][2\*1] | C |
| 207 | 下述对C++语言字符数组的描述中错误的是(). | 字符数组中的字符串可以整体输入,输出. | 不可以用关系运算符对字符数组中的字符串进行比较. | 字符数组中可以存放字符串. | 可以通过赋值运算符"="对字符数组整体赋值. | D |
| 208 | 若有说明:int array[3][3]={0};,则下面正确的叙述是() | 只有元素array[0][0]可得到初值 | 此说明语句不正确 | 数组array中每个元素均可得到初值0 | 数组array中各元素都可得到初值,但其值不一定为0 | C |
| 209 | 若二维数组a有m列,则在a[i][j]前的元素个数为(). | j\*m+i | i\*m+j+1 | j\*m+i+1 | i\*m+j | D |
| 210 | 已知有声明"char s[20]="Hello";",在程序运行过程中若要想使数组s中的内容修改为"Good",则以下语句能够实现此功能的是(). | strcpy(s,"Good"); | strcat(s,"Good"); | s[20]="Good"; | s="Good"; | A |
| 211 | 若有以下说明: int a[12]={1,2,3,4,5,6,7,8,9,10,11,12}; char c='a',d,g; ,则数值为4的表达式是() | a['d'-c] | a[4] | a['d'-'c'] | a[g-c] | A |
| 212 | 请阅读以下程序:   void fun(int b[]) { static int i=0;  do  { b[i]+=b[i+1];  }while(++i<2); }  int main() { int k,a[5]={1,3,5,4,9};  fun(a);  for(k=0;k<5;k++)cout<<a[k]; }   上面程序的输出是(). | 48579 | 48549 | 48999 | 13579 | B |
| 213 | 有以下程序:  main() { int aa[5][5]={{5,6,1,8},{1,2,3,4},{1,2,5,6},{5,9,10,2}};  int i,s=0;  for(i=0;i<4;i++) s+=aa[i][2];  cout<s;  } 程序运行后的输出结果是(). | 19 | 26 | 20 | 10 | A |
| 214 | 若有说明:int a[3][4];,则对a数组元素的正确引用是(). | a[2][4] | a[1+1][0] | a[1,3] | a(2)(1) | B |
| 215 | 以下对一维整型数组a的正确说明是(). | int n=10,a[n]; | int n; cin>>n; int a[n]; | #define SIZE 10(换行)int a[SIZE]; | int a(10); | C |
| 216 | char a1[]="abc",a2[80]="1234";将a1串连接到a2串后面的语句是(). | strcat(a2,a1); | strcpy(a2,a1); | strcpy(a1,a2); | strcat(a1,a2); | A |
| 217 | 对以下说明语句 int a[10]={6,7,8,9,10};的正确理解是(). | 将5个初值依次赋给a[6]至a[10] | 将5个初值依次赋给a[0]至a[4] | 因为数组长度与初值的个数不相同,所以此语句不正确 | 将5个初值依次赋给a[1]至a[5] | B |
| 218 | 若有说明:int a[][3]={1,2,3,4,5,6,7};,则a数组第一维的大小是(). | 2 | 无确定值 | 4 | 3 | D |
| 219 | 若已有定义: int a[]={0,1,2,3,4,5,6,7,8,9},\*p=a,i; 其中,0≤i≤9,则对a数组元素的引用不正确的是(). | p[i] | a[p - a] | \*(\*(a+i)) | \*(&a[i]) | C |
| 220 | 若有定义:int m[]={5,4,3,2,1},i=4;,则下面对m数组元素的引用中,错误的是(). | m[m[i]] | m[2\*2] | m[m[0]] | m[--i] | C |
| 221 | 定义int a[]={0,1,2,3,4,5,6,7,8,9},\*p=a,i;,其中,0≤i≤9,则对a数组元素的引用不正确的是(). | a[p-a] | \*(&a[i]) | p[i] | \*(\*(a+i)) | D |
| 222 | 若有以下说明: int a[12]={8,9,10,11,12}; 则值为9的表达式是(). | a[2] | a[3] | a[1] | a[4] | C |
| 223 | 以下程序的输出结果是 main() {char ch[2][5]={"6937","8254"},\*p[2]; int i,j,s=0; for(i=0;i<2;i++) p[i]=ch[i]; for(i=0;i<2;i++) for(j=0;p[i][j]>'\0' && p[i][j]<='9';j+=2) s=10 \* s+p[i][j]- '0'; pritnf("%d\n",s); } | 6385 | 69825 | 63825 | 693825 | A |
| 224 | 以下程序段的输出结果为(). for(i=4;i>1;i--) for(j=1;j<i;j++) putchar('#'); | # | ### | 无 | ###### | D |
| 225 | 若有说明int n[4][4],则n数组元素的非法引用是() | n[0][4] | n[5-2][0] | n[0][3\*1] | n[1][3] | A |
| 226 | int a[10]; 给数组a的所有元素分别赋值为1,2,3,……的语句是(). | for(i=1;i<11;i++)a[i-1]=i; | for(i=1;i<11;i++)a[i]=i; | for(i=1;i<11;i++)a[i+1]=i; | for(i=1;i<11;i++)a[0]=1; | A |
| 227 | 函数定义时的参数为形参,调用函数时所用的参数为实参,则下列描述正确的是(). | 实参与形参是双向传递 | 形参和实参可以同名 | 实参类型一定要在调用时指定 | 形参可以是表达式 | B |
| 228 | 存在多层嵌套循环时,下列说法正确的是(). | continue语句能够中,止包含该语句的最内层循环 | break语句可以中,止所有循环 | break语句能够中,止包含该语句的最内层循环 | continue语句可以中,止所有循环 | C |
| 229 | 以下函数调用语句中,实参的个数是(). func((e1,e2),(e3,e4,e5)); | 3 | 5 | 语法错误 | 2 | D |
| 230 | 若变量已正确定义,要将a和b中的数进行交换,下面不正确的语句是( ) | t=a;a=b;b=t; | t=b;b=a;a=t; | a=a+b,b=a-b,a=a-b; | a=t;t=b;b=a; | D |
| 231 | 下面对函数的叙述,正确的是(). | 函数中无论是否有 return语句,都能返回正确值 | 函数中没有 return语句,说明不带回值 | 函数不能嵌套定义 | 函数的返回值是通过 return语句获得的 | C |
| 232 | C++语言的if语句嵌套时,if 与else 的配对关系是(). | 每个else 总是与它上面的最近的if 配对 | 每个else 总是与最外层的if 配对 | 每个else 与if 的配对是任意的 | 每个else 总是与它上面的if 配对 | A |
| 233 | C++语言关于实参和形参说法正确的是(). | 实参和与之对应的形参占用同一存储单元 | 形参是虚拟的,不占用存储单元 | 实参和与之对应的形参各占用独立的存储单元 | 只有当实参与形参同名时,才共占用同一存储单元 | C |
| 234 | 在调用函数时,如果实参是简单变量,它与对应形参之间的数据传递方式是(). | 传递方式由用户指定 | 由实参传给形参,再由形参传回实参 | 地址传递 | 单向值传递 | D |
| 235 | 求平方根函数的函数名为(). | abs | cos | pow | sqrt | D |
| 236 | C++语言规定,函数返回值的类型是(). | 由return语句中的表达式类型所决定 | 由调用该函数时的主调函数类型所决定 | 由调用该函数时系统临时决定 | 由定义该函数时所指定的数值类型决定 | D |
| 237 | 若函数调用时的实参为变量,以下关于函数形参和实参的叙述中,正确的是(). | 函数的形参和实参分别占用不同的存储单元 | 函数的实参和其对应的形参共占同一存储单元 | 形参只是形式上的存在,不占用具体存储单元 | 同名的实参和形参占同一存储单元 | A |
| 238 | 在定义全局变量和局部静态变量的同时赋值,变量的初始化在()时确定的. | 编辑 | 调试 | 编译 | 运行 | C |
| 239 | 以下对C++语言函数的有关描述中正确的是(). | 在C++中调用函数时,只能把实参值传给形参,形参值不能返回给实参 | C++程序中有调用关系的所有函数必须放在同一个源程序文件中 | 没有返回值的函数不能被使用 | C++函数可以递归调用也可以嵌套调用 | D |
| 240 | 假设所有变量均为整型,则表达式(a=1,b=9,b++,a+b)的值是() | 9 | 10 | 11 | 12 | C |
| 241 | 用数组名作为函数调用时的实参,实际上传递给形参的是(). | 数组元素的个数 | 数组全部元素的值 | 数组第一个元素的值 | 数组首地址 | D |
| 242 | 对auto存储类型的变量,若不赋值,则其值是(). | 系统自动赋值0 | ' \0 ' | 根据情况赋值 | 随机的 | D |
| 243 | 关于建立函数的目的,以下正确的说法是(). | 减少程序文件所占内存 | 提高程序的执行效率 | 提高程序的可读性 | 减少程序的篇幅 | C |
| 244 | 若调用一个函数,且此函数中,没有 return语句,则正确的说法是:该函数(). | 没有返回值 | 返回一个不确定的值 | 返回若干个系统默认值 | 能返回一个用户所希望的值 | B |
| 245 | 以下对静态局部变量的叙述哪一个是不正确的(). | 静态局部变量在整个程序运行期间都不释放 | 在一个函数中定义的静态局部变量可以被另一函数调用 | 数值型静态局部变量的初值默认为0 | 静态局部变量是在编译时赋初值的,故它只被赋值一次 | B |
| 246 | 以下叙述中,不正确的是(). | 在一个函数内定义的变量只在本函数范围内有效 | 在一个函数内的复合语句中,定义的变量在本函数范围内有效 | 函数中的形式参数是局部变量 | 在不同的函数中,可以使用相同的名字的变量 | B |