

计算机-C++程序设计（本科）

2017 年 10 月真题及答案解析

单项选择题：本大题共 20 小题。每小题 1 分，共 20 分。

1. 按照标识符的要求，不能组成标识符的符号是（ ）。

- A. 连接符
- B. 下划线
- C. 大小写字母
- D. 数字字符

答案：A

解析：组成标识符的符号可以是大小写字母、下划线和数字，且数字不能打头。

2. 下列输出语句中，正确的是（ ）。

- A. `cout<<("%c\n","student")`
- B. `cout<<("%s\n","hello")`
- C. `cout<<("%c\n","c")`
- D. `cout<<("%s\n",&a)`

答案：B

解析：`%c` 是输出单个字符的格式控制符；选项 C 中 "c" 是字符串，不能用 `c%` 格式输出。`%s` 是输出字符串的格式控制符。

3. 已知：`print()` 函数是一个类的常成员函数，无返回值，下列表示中正确的是（ ）。

- A. `void print() const`
- B. `void print(const)`
- C. `void const print()`
- D. `const void print()`

答案：A

解析：常成员函数的声明形式：类型标识符 函数名（参数列表）`const`；

4. if 与 else 在使用过程中为避免嵌套出现二义性，C++中规定与 else 子句配对的是（ ）。

- A. 其之前最近的 if 语句
- B. 其之前最近且尚未配对的 if 语句
- C. 缩排位置相同的 if 语句
- D. 其之后最近的 if 语句

答案：B

解析：C++中规定与 else 子句配对的是其之前最近且尚未配对的 if 语句。

5. 对使用关键字 new 所开辟的动态存储空间，释放时必须使用（ ）。

- A. free
- B. create
- C. delete
- D. release

答案：C

解析：C++为结构动态分配内存的一般格式为：指针名=new 结构名；当不再使用这个空间时，必须使用 delete 释放空间，格式为：delete 指针名；。

6. 逻辑运算符两侧运算对象的数据（ ）。

- A. 是逻辑型数据
- B. 只能是整型数据
- C. 只能是整型或字符型数据
- D. 可以是任何类型的数据

答案：D

解析：逻辑运算符两侧运算对象的数据可以是任何类型的数据。

7. 关于构造函数的说法，不正确的是（ ）。

- A. 没有定义构造函数时，系统将不会调用它
- B. 其名与类名完全相同
- C. 它在对象被创建时由系统自动调用

D. 没有返回值

答案：A

解析：当没有为一个类定义任何构造函数的情况下，C++编译器会自动建立一个不带参数的、函数体为空的构造函数，这就是默认构造函数。构造函数不能在程序中显式调用，它是由系统自动调用的。为了提高安全性和效率，构造函数的名字必须和类名相同，并在定义构造函数时不能指定返回类型，即不要返回值，即使是 void 类型也不可以。

8. 所谓数据封装就是将一组数据和与这组数据有关操作组装在一起，形成一个实体，这实体也就是（）。

A. 类

B. 对象

C. 函数体

D. 数据块

答案：A

解析：类是具有相同的属性和操作的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性（数据成员）和操作（成员函数）两个主要部分。

9. 下列关于类的继承描述中，错误的是（）。

A. 基类不一定具有派生类的全部属性和方法

B. 派生类可以访问基类的所有数据成员，也能调用基类的所有成员函数

C. 继承描述类的层次关系，派生类可以具有与基类相同的属性和方法

D. 一个基类可以有多个派生类，一个派生类可以有多个基类

答案：B

解析：派生类继承了基类所有的数据成员和成员函数。相对基类，派生类可以有以下变化：增加新的成员；重新定义已有的成员函数；改变基类成员的访问权限。一个基类可以有多个派生类，一个派生类可以有多个基类。

10. 在 C++ 中，函数原型不能标识（）。

A. 函数的返回类型

B. 函数参数的个数

C. 函数参数类型

D. 函数的功能

答案：D

解析：函数原型可以标识函数的返回类型、函数参数的个数和函数参数类型，但不能标识函数的功能。

11. 若二维数组 y 有 m 列，则位于 $y[i][j]$ 之前的元素数量是（ ）。

A. $j * m + i$

B. $i * m + j$

C. $i * m + j - 1$

D. $i * m + j + 1$

答案：B

解析： $y[i][j]$ 前面的元素有前 i 行 $i * m$ 个元素再加上第 i 行第 j 列前面的 j 个元素，一共 $i * m + j$ 个元素。

12. 下列关于类的权限的描述错误的是（ ）。

A. 类本身的成员函数只能访问自身的私有成员

B. 类的对象只能访问该类的公有成员

C. 普通函数不能直接访问类的公有成员，必须通过对象访问

D. 一个类可以将另一个类的对象作为成员

答案：A

解析：类本身的成员函数既可以访问自身的私有成员，更可以访问自身的公有成员。

13. 下面不能够判断字符串 S 是空串的是（ ）。

A. `if(S[0]==0)`

B. `if(strlen(S)==0)`

C. `if(strcmp(S, "")==0)`

D. `if(S=='\0')`

答案：D

解析： S 是字符串， S 本身代表字符串本身及其首地址，也可以理解为是存放字符串的数组的数组名，选项 A 是正确的，因为如果字符串的第一个字符的 ASCII 码值是 0（字符串的结束标

记的\0 的 ASCII 码值就是 0)，说明就是空串。选项 B，if(strlen(S)==0)是利用系统函数判断字符串 S 的长度是否等于 0，当然能够判断字符串 S 是空串。选项 C，if(strcmp(S, "")==0) 是利用系统函数比较字符串 S 和空串是否相等，当然能够判断字符串 S 是否是空串。选项 D，if(S=='\0')中的 S=='\0'这个表达式左边是字符串，右边是字符，显然是非法的，如果改写成 if(S=="\0")就可以判断字符串 S 是否是空串。

14. 下列输出字符'd'的方法中，错误的是（）。

- A. cout<<put('d')
- B. cout<<'d'
- C. cout. put('d')
- D. char a='d'; cout<<a;

答案：A

解析：cout.put（）是一个输出字符的函数，cout.put('d')相当于cout<<'d'。选项 A 是非法语句。

15. 关于引用，下列的说法中错误的是（）。

- A. 引用是给被引用的变量取一个别名
- B. 引用主要是用来作函数的形参和函数的返回值
- C. 在声明引用时，要给它另开辟内存单元
- D. 在声明引用时，必须同时使它初始化

答案：C

解析：引用就是为现有的对象起个别名，别名的地址与引用对象的地址是一样的，使用引用时没有分配新的存储区域。引用主要是用来作函数的形参和函数的返回值；对象在引用前必须先初始化。

16. 下面关于 C++字符数组的叙述中，错误的是（）。

- A. 字符数组可以放字符串
- B. 字符数组的字符可以整体输入、输出
- C. 可以在赋值语句中通过赋值运算符“=”对字符数组整体赋值
- D. 可以用关系运算符对字符数组比较大小

答案：C

解析：以下列这个程序为例：`#include <iostream>#include <string>using namespace std;void main(){char s1[]="ABCD",s2[6];cin>>s2;if(s1!=s2) cout<<s1<<endl;else cout<<s2<<endl;}程序运`

```
EDCBA
ABCD
Press any key to continue
```

行结果：可见字符数组可以放字符串，字符数组的字符可以整体输入、输出。C++字符数组可以用关系运算符对字符数组比较大小。把上面程序中的 `cin>>s2;`改为 `s2="EDCBA";`编译出错，错误信息为：

17. 下列说法不正确的是（ ）。

- A. 主函数 main 中定义的变量在整个文件或程序中有效
- B. 不同函数中，可以使用相同名字的变量
- C. 形式参数是局部变量
- D. 在一个函数内部，可以在复合语句中定义变量，这些变量只在复合语句中有效

答案：A

解析：主函数 main 中定义的变量只在主函数中有效。

18. 非数组指针或引用型变量做实参时，它和对应虚参之间的数据传递方式是（ ）。

- A. 地址传递
- B. 单向值传递
- C. 双向值传递
- D. 由用户指定传递方式

答案：B

解析：非数组指针做实参时和对应虚参之间的数据传递方式是单向地址值的传递。引用型变量做实参时，它和对应虚参之间的数据传递方式是单向数值传递。引用型变量做形参时，它和对应实参之间的数据传递方式是地址传递。下面是引用型变量做实参时的程序实例：`#include <iostream>#include <string>using namespace std;void swap(string s11, string s22){ string temp=s11;s11=s22;s22=temp; cout<<"交换为 str1 =<<s11<<"str2="<<s22<<endl;}void main(){ string str1("现在"),str2("过去"); string &s1=str1, &s2=str2; cout<<"引用:s1="<<s1<<"s2="<<s2<<endl; swap(s1, s2); cout<<"返回`

```
引用:s1=现在s2=过去
交换为str1 =过去str2=现在
返回后:str1=现在str2=过去
Press any key to continue
```

后:str1="<<str1<<"str2="<<str2<<endl;}运行结果：

19. 下面叙述中错误的是（ ）。

- A. 预处理命令都必须以“#”开始
- B. 在程序中凡是以“#”开始的语句行都是预处理命令行
- C. C++程序在程序执行过程中对预处理命令进行处理
- D. 一行只能写一条预处理命令

答案：C

解析：预处理命令在源程序编译之前处理，预处理命令都以“#”开头，后面没有“;”，一行只能写一条预处理命令，预处理命令不是 C++ 的一部分。

20. 下列关于友元函数的描述，正确的是（ ）。

- A. 友元函数可以存取私有成员、公有成员和保护成员
- B. 友元函数不可以是一个类
- C. 友元函数的作用之一是实现数据的隐藏性
- D. 在类中说明的友元函数，函数的定义不可在类体之外

答案：A

解析：可以在类 A 中通过关键字 `friend` 声明或定义某个独立函数或另一个类 B 的某个成员函数或另一个类 B 为类 A 的友元函数，友元函数可以无限制地存取类 A 的成员（包括私有、公有和保护成员）。友元函数可在类中的私有或公有部分通过关键字 `friend` 说明或定义，但如在类中声明，而在类外定义，就不能再在类外使用 `friend` 关键字。类友元函数应被看作类的接口的一部分，使用它的主要目的是提高效率，因为它可以直接访问对象的私有成员，从而省去调用类的相应成员函数的开销。

填空题：本大题共 20 空，每空 1 分，共 20 分。

21. `#include <iostream. h>`命令中，`include` 的意义是：_____。

答案：包含头文件

22. `this` 指针始终指向调用成员函数的_____。

答案：对象

23. 假定 AB 为一个类，则执行语句 `AB a[10];` 时，系统自动调用该类的构造函数的次数为_____。

答案：10

24. 一个抽象类的派生类可以实例化的必要条件是实现了所有的_____。

答案：纯虚函数

25. cin 是输入流 istream 的一个对象，处理标准输入；_____是输出流 ostream 的一个对象，处理标准输出。

答案：COUT

26. 定义虚函数所用的关键字是_____。

答案：VIRTUAL

27. 设在程序中使用如下语句申请了一个对象数组：Point *ptr=new Point[2]；当要释放 ptr 指向的动态数组对象时，所使用的语句是_____。

答案：DELETE[]PTR;

28. 书写程序语句时，适当增加空行和程序注释以增加程序的_____。

答案：可读性

29. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取_____方式。

答案：传地址或引用

30. 如果要把 A 类成员函数 f()且返回值为 void 声明为类 B 的友元函数，则应在类 B 的定义中加入语句_____。

答案：FRIEND VOID A::F();

31. C++语言的_____提供了与要操作的元素类型无关的算法。

答案：标准模板库(STL)

32. 为了在对象生存期结束时释放其指针成员所指向的动态存储空间，通常为该类定义_____。

答案：析构函数

33. 在保护派生中，基类权限为 public 的成员在派生类中为_____。

答案：保护类型(PROTECT)

34. 在函数前面用_____保留字修饰时，则表示该函数为内联函数。

答案：INLINE

35. 面向对象的四个基本特性是多态性、继承性、封装性、_____。

答案：抽象

36. 派生类的主要用途是可以定义其基类中_____。

答案：不具有的成员

37. 若 `int a=8; int b=(++a)++;` 则 `b=_____`。

答案：9

38. 在 C++ 中，利用向量类模板定义一个具有 10 个 `int` 的向量 A，其元素均被置为 1，实现此操作的语句是_____。

答案：VECTOR<INT>A(10,1)

39. 将指向对象的引用作为函数的形参，形参是对象的引用，实参是_____。

答案：对象名

40. 重载的运算符保持其原有的_____、优先级和结合性不变。

答案：操作数

改错题：本大题共 5 小题，共 20 分。

41.



```
#include "stdafx.h"
#include <iostream>
using namespace std;
class Test
{ private:
int x,y = 20;
public:
Test(int i,int j) { x = i,y = j;}
int getx() { return x;}
int gety() { return y;}
};
void main()
{ Test mt(10,20);
cout << mt.getx() << endl;
cout << mt.gety() << endl;
}
```

答案：INT X,Y=20; 错。在类内部不能对数据成员直接赋值。[修改] INT X,Y;

42.



```
#include <iostream.h>
#include <string.h>
class Base
{ public:
Base(char *s = "\0") { strcpy(name,s); }
void show();
protected:
char name[20];
};
Base b;
void show()
{ cout << "name:" << b.name << endl; }
void main()
{ Base d2("hello");
show();
}
```

答案: VOID SHOW(); 错。它是普通函数不是成员函数，但是要访问类成员，需要定义为友元函数。[修改] FRIEND VOID SHOW();

43.

```
#include <iostream >
using namespace std;
int main() {
const int num = 20;
int scores[ num ];
for( int i = 1; i < = num; i ++ )
scores[ i ] = i;
return 0; }
```

答案: SCORES[I]=I;或 I<=MUM;错。数组访问越界。[修改] SCORES[I-1]=I; 或 I<MUM;

44.

```
#include <iostream.h >
class Aton
{ int X,Y;
protected:
int zx,zy;
public:
void init( int i,int j) { zx = i;zy = j;}
Aton( int i,int j,int n =0,int m =0)
{ X = i,Y = j,zx = m,zy = n;
}
};
void main( )
{ Aton A(25,20,3,5);
A. init(5,9);
cout << A. X << endl;
cout << A. Y << endl;
}
```

答案：INT X,Y;错。因为 X, Y 都是私有的，在类外无法直接访问。[修改] PUBLIC: INT X,Y;

45.

```
#include <iostream >
using namespace std;
int main( )
{ int a = 10,b = 18,c = 77;
const int *p = c;
cout << *p << endl;
return 0;
}
```

答案：CONST INT * P=C;错。P 是指针不能用变量赋值。[修改] CONST INT *P=&C;

完成程序题：本大题共 5 小题，每小题 4 分，共 20 分。

46. 下面程序中用来求数组和。请在下面程序的横线处填上适当内容，以使程序完整，并使程序的输出为：s=150。

```
#include <iostream. h >
class Arr
{ int * a , n ;
public:
Arr() : a ( 0 ) , n ( 0 ) {}
Arr( int * aa , int nn )
{ n = nn ;
a = new int [ n ] ;
for( int i = 0 ; i < nn ; i ++ )
* ( a + i ) = * ( aa + i ) ;
}
~ Arr() { delete a ; }
_____ ;
{ return * ( a + i ) ; }
};
void main()
{ int b [ 5 ] = { 10 , 20 , 30 , 40 , 50 } ;
Arr a1 ( b , 5 ) ;
int i = 0 , s = 0 ;
_____
s += a1 . GetValue ( i ) ;
cout << " s = " << s << endl ;
}
```

答案：INT GETVALUE(INT I)，FOR(;I<5;I++)

47. 在下面程序的横线处填上适当字句，完成类的定义。

```
class line;
class box
{ private:
int color;
int upx, upy;
int lowx, lowy;
public:
_____
void set_color ( int c ) { color = c; }
void define_box ( int x1 , int y1 , int x2 , int y2 )
{ upx = x1 ; upy = y1 ; lowx = x2 ; lowy = y2 ; }
};
class line
{ private:
int color;
int startx, starty;
int endx, endy;
public:
friend int same_color( line l, box b );
void set_color ( int c ) { color = c; }
void define_line ( _____ )
{ startx = x1 ; starty = y1 ; endx = x2 ; endy = y2 ; }
};
int same_color( line l, box b )
{ if ( l. color == b. color ) return 1 ;
return 0 ;
}
```

答案： FRIEND INT SAME_COLOR(LINE l, BOX b);, INT X1, INT Y1, INT X2, INT Y2

48. 在下面程序的横线处填上适当的字句，使该程序执行结果为 40。

```
#include <iostream. h >
class Test
{ public:
    _____;
    Test (int i =0)
    { x = i + x; }
    int Getnum()
    { return Test: :x + 10; }
};

_____ ;
void main()
{ Test test;
  cout << test. Getnum() << endl;
}
```

答案: STATIC INT X;, INT TEST: :X=30;

49. 在下面程序的横线处填上适当的字句，使输出为: 0,2,10。



```
#include "stdafx.h"
#include <iostream>
#include "math.h"
using namespace std;
class Magic
{ double x;
public:
Magic( double d = 0.00 ) : x( fabs( d ) )
{ }
Magic operator + ( _____ )
{
return Magic( sqrt( x * x + c. x * c. x ) );
}
_____ &operator << ( ostream & stream, Magic & c )
{ stream << c. x;
return stream;
}
};
void main( )
{ Magic ma;
cout << ma << " , " << Magic( 2 ) << " , " << ma + Magic( - 6 ) + Magic( - 8 ) << endl;
}
```

答案: OPERATOR+(MAGIC & C), FRIEND OSTREAM&OPERATOR

50. 下面是一个输入半径，输出其面积和周长的 C++ 程序，在横线处填上正确的语句。（圆周率为 3.14159）


```
#include "stdafx.h"
#include <iostream>
#include "math.h"
using namespace std;
_____ //宏定义
void main()
{ double rad;
cout << "rad = ";
cin >> rad;
double l = 2.0 * pi * rad;
_____ ;
cout << "\n The long is:" << l << endl;
cout << "The area is:" << s << endl; }
```

答案：# DEFINE PI3.14159, DOUBLE S= PI * RAD * RAD;

程序分析题：本大题共 2 小题，每小题 5 分，共 10 分。

51. 给出下面程序的输出结果



```
#include <iostream. h >

class example
{ int a;
public:
example( int b =5) { a = b ++ ;}
void print() { a = a + 1 ;cout << a << " " ;}
void print() const
{ cout << a << endl ;}
};

void main()
{ example x;
const example y(2) ;
x. print() ;
y. print() ;
}
```

答案： 6 2

52. 给出下面程序的输出结果



```
#include <iostream. h >
class Based
{ public:
Based()
{ cout << "Based 构造函数\n";fun();}
virtual void fun()
{ cout << "Based::fun() 函数\n";}
};
class Derived:public Based
{ public:
Derived()
{ cout << "Derived 构造函数\n";fun();}
void fun() { cout << "Derived::fun() 函数\n";}
};
void main()
{ Derived d;
}
```

答案：BASED 构造函数 BASED::FUN()函数 DERIVED 构造函数 DERIVED::FUN()函数

程序设计题：本大题共 1 题，每小题 10 分，共 10 分。

53. 声明复数类 Complex，该类中有两个私有变量 real，image 分别表示一个复数的实部和虚部。为 Complex 类添加适当的构造函数，并使用友元函数 add 实现复数加法。

```
#include <iostream >
using namespace std;
class Complex
{
private:
double real, image;
public :
void setRI( double a, double b)
{
real = a;
image = b;
}
double getReal()
{
return real;
}
}
```

答案：在类内添加：COMPLEX(){ }COMPLEX(DOUBLE A , DOUBLE B){REAL = A;IMAGE = B;}类外添加：COMPLEX ADD(COMPLEX C1, COMPLEX C2)//定义友元函数{COMPLEX C3;C3.REAL=C1.REAL + C2.REAL;//访问 COMPLEX 类中的私有成员 C3.IMAGE= CL.IMAGE+C2.IMAGE;RETURN C3;}

