

全国 2017 年 10 月高等教育自学考试

C++程序设计试题

课程代码:04737

请考生按规定用笔将所有试题的答案涂、写在答题纸上。

选择题部分

注意事项:

1. 答题前,考生务必将自己的考试课程名称、姓名、准考证号用黑色字迹的签字笔或钢笔填写在答题纸规定的位置上。

2. 每小题选出答案后,用 2B 铅笔把答题纸上对应题目的答案标号涂黑。如需改动,用橡皮擦干净后,再选涂其他答案标号。不能答在试题卷上。

一、单项选择题:本大题共 20 小题,每小题 1 分,共 20 分。在每小题列出的备选项中只有一项是最符合题目要求的,请将其选出。

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

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)`

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()`

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

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`

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

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
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
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
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
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. $i * m + j + 1$
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
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')`
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;`
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

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
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
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
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
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

非选择题部分

注意事项:

用黑色字迹的签字笔或钢笔将答案写在答题纸上,不能答在试题卷上。

二、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: _____。

22. this 指针始终指向调用成员函数的 _____。

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 _____。

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 _____。

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。
26. 定义虚函数所用的关键字是 _____。
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 _____。
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 _____。
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。
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 _____。
31. C++ 语言的_____提供了与要操作的元素类型无关的算法。
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 _____。
33. 在保护派生中,基类权限为 public 的成员在派生类中为_____。
34. 在函数前面用_____保留字修饰时,则表示该函数为内联函数。
35. 面向对象的四个基本特性是多态性、继承性、封装性、_____。
36. 派生类的主要用途是可以定义其基类中_____。
37. 若 int a = 8; int b = (++ a) ++; 则 b = _____。
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 _____。
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 _____。
40. 重载的运算符保持其原有的 _____、优先级和结合性不变。

三、改错题:本大题共 5 小题,每小题 4 分,共 20 分。

下面的程序段中有一处错误,请在“答题纸”上写出错误行语句并指出错误原因。

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;
}
```

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();
}
```

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; }
```

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;
}
```

45.

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

四、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;
}
```

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 ;
}

```

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 ;
}

```

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;
}

```

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; }

```

五、程序分析题:本大题共 2 小题,每小题 5 分,共 10 分。

51. #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();
}
```

52. #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;
}
```

六、程序设计题:本大题共 1 小题,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;
}
double getImage()
{
return image;
}
void print() {
if(image > 0)
cout << "复数:" << real << " + " << image << "i" << endl;
if(image < 0)
cout << "复数:" << real << " - " << image << "i" << endl;
}
friend Complex add(Complex, Complex); //声明友元函数
};
void main()
{
Complex c1(19, 0.864), c2, c3;
c2.setRI(90, 125.012);
c3 = add(c1, c2);
cout << "复数一:"; c1.print();
cout << "复数二:"; c2.print();
cout << "相加后:"; c3.print();
}
```