

全国 2018 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. 若有以下类型标识符定义:char c = 'c'; int a = 4; float f = 3.14; double d = 1.212;则表达式 c + a/(int)d + f 的结果类型是
A. float B. char C. int D. double
2. 设 x 和 y 均为 bool 量,则 x&&y 为真的条件是
A. 它们均为真 B. 其中一个为真 C. 它们均为假 D. 其中一个为假
3. 拷贝构造函数应该是
A. 不带参数的构造函数 B. 带有一个参数的构造函数
C. 带有两个参数的构造函数 D. 缺省构造函数
4. 以下说法中正确的是
A. C++ 程序总是从第一个定义的函数开始执行
B. C++ 程序总是从 main 函数开始执行
C. C++ 中函数必须有返回值
D. C++ 中函数名必须唯一
5. 下列虚基类的声明中,正确的是
A. class virtual B:public A B. class B:virtual public A
C. class B: public A virtual D. virtual class B:public A
6. 下列哪个类型函数不适合声明为内联函数
A. 函数体语句较多 B. 函数体语句较少
C. 函数执行时间较短 D. 函数被频繁调用



7. 以下类模板定义正确的为
- A. `template < class T >` B. `template < class T, class int i >`
C. `template < class T, typename T >` D. `template < class T1, T2 >`
8. C++ 中要实现动态联编,调用虚函数时必须使用
- A. 基类指针 B. 类名 C. 派生类指针 D. 对象名
9. 下列对静态成员的描述中,不正确的是
- A. 静态成员不属于对象,是类的共享成员
B. 静态数据成员要在类外定义和初始化
C. 调用静态成员函数时要通过类或对象激活,所以静态成员函数拥有 `this` 指针
D. 非静态成员函数也可以操作静态数据成员
10. 下列对派生类的描述中,错误的是
- A. 一个派生类可以作为另一个派生类的基类
B. 派生类至少有一个基类
C. 派生类的成员除了它自己的成员外,还包含了它的基类的成员
D. 派生类中继承的基类成员的访问权限到派生类保持不变
11. 下列函数原型声明语句中,错误的是
- A. `int f(void);` B. `void f(int);` C. `int f(a);` D. `void f(double a);`
12. 如果有 `int` 型变量 `a`,则定义指向变量 `a` 的指针 `p` 正确的写法是
- A. `int p = &v` B. `int *p = &v` C. `int &p = *v` D. `int *p = v`
13. 假定指针变量 `p` 定义为“`int *p = new int(100);`”,要释放 `p` 所指向的动态内存,应使用语句
- A. `delete p;` B. `delete *p;` C. `delete &p;` D. `delete []p;`
14. 假定 `A` 为一个类,则执行“`A a[3], b(3);`”语句时调用该类构造函数的次数为
- A. 3 B. 4 C. 5 D. 9
15. C++ 中定义标准输入输出的库为
- A. `stdio` B. `math` C. `iostream` D. `stdlib`
16. 允许用户为类定义一种模式,使得类中的某些数据成员及某些成员函数的返回值能取任意类型,这是一个
- A. 类模板 B. 模板类 C. 函数模板 D. 模板函数
17. 下列关于运算符重载的叙述中,正确的是
- A. 通过运算符重载,可以定义新的运算符
B. 有的运算符只能作为成员函数重载
C. 若重载运算符 `+`,则相应的运算符函数名是 `+`
D. 重载一个二元运算符时,必须声明两个形参



18. 当使用 `ofstream` 流类定义一个流对象并打开一个磁盘文件时,文件的隐含打开方式为
- A. `ios::out | ios::binary` B. `ios::in | ios::binary`
C. `ios::out` D. `ios::in`
19. 有关函数模板和模板函数说法错误的是
- A. 函数模板只是对函数的描述,编译器不为其产生任何执行代码,所以它不是一个实实在在的函数
B. 模板函数是实实在在的函数,它由编译系统在遇到具体函数调用时所生成,并调用执行
C. 函数模板需要实例化为模板函数后才能执行
D. 当函数模板和一般函数同名时,系统先去匹配函数模板,将其实例化后进行调用
20. 对类的构造函数和析构造函数描述正确的是
- A. 构造函数可以重载,析构造函数不能重载
B. 构造函数不能重载,析构造函数可以重载
C. 构造函数可以重载,析构造函数也可以重载
D. 构造函数不能重载,析构造函数也不能重载

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. 将 `int` 类型指针 `p` 转换为 `char` 类型指针,则强制转换语句为_____。
22. 当一个成员函数被调用时,该成员函数的_____指向调用它的对象。
23. 在_____继承的情况下,基类数据成员在派生类中的访问权限保持不变。
24. Windows 环境下,由 C++ 源程序文件编译而成的目标文件的扩展名是_____。
25. 具有至少一个纯虚函数的类是_____。
26. C++ 中字符串是通过字符数组来表示的,每一个字符串都有一个结尾字符_____。
27. C++ 中定义重载函数时,应至少使重载函数的参数个数或_____不同。
28. 假定 $x = 15$,则表达式 $x < = 10 ? 20 : 30$ 的值为_____。
29. 不同对象可以调用相同名称的函数,但执行完全不同行为的现象称为_____。
30. 假设类 `F` 的对象 `f` 是类 `A` 的成员对象,则“`A a`”语句执行时,先调用类_____的构造函数。



31. 设“int a=3,b=4,c=5;”,表达式“(a+b)>c&&b==c”的值是_____。
32. 描述命题“A小于B或小于C”的表达式为_____。
33. C++中字符串“a+b=12\n\t”的长度为_____。
34. C++类中构造函数的个数最多是无限个,析构函数的个数最多是_____个。
35. C++的静态数据成员需要在_____进行初始化,可以被该类的所有对象共享。
36. C++中一般的程序都要有两条语句,包含头文件“#include <iostream>”语句和使用命名空间“_____”语句。
37. 用new申请某一个类的动态对象数组时,在该类中必须能够匹配到没有形参的或缺省参数的_____,否则应用程序会产生一个编译错误。
38. C++中解决命名冲突的机制是_____。
39. 拷贝构造函数使用_____作为参数初始化创建中的对象。
40. 假如一个类的名称为F,使用这个类的一个对象初始化该类的另一个对象时,可以调用_____构造函数来完成此功能。

三、改错题:本大题共5小题,每小题4分,共20分。以下程序中各有一处错误,请抄写有错误的语句并改正。

41. #include <iostream.h>
using namespace std;
void setzero(int &a) {
 a=0;}
int main() {
 int x1=10;
 setzero(&x1);
 cout << x1 << endl;
 return 0;
}
42. 改正后程序输出结果为2 5 8 11 14
#include <iostream.h>
using namespace std;
void main()
{
 int i=1;
 while(i<=15){
 i++;
 if(i%3==2)continue;
 else cout << i << " ";
 }
}



43. #include <iostream.h>

using namespace std;

class Point

{

public:

void init() {}

static void output() {}

};

void main()

{

Point P;

Point::init();

P.output();

}

44. #include <iostream.h>

using namespace std;

class A {

int x,y;

public:

void set(int a,int b)

{ x = a;y = b;}

int getx()

{ return x;}

int gety()

{ return y;}

int sum()

{ return x + y;}

};

int main() {

A a;

a.set(3,4);

cout << a.x << a.y << a.sum() << endl;

return 0;

}



45. 申请一个长度为 10 的 int 型空间,之后释放该空间。

```
#include <iostream. h >
using namespace std;
int main() {
    int *p = new int[10];
    delete p;
    return 0;
}
```

四、完成程序题:本大题共 5 小题,每小题 4 分,共 20 分。请在答题纸上按试题顺序和空格顺序写出答案。

46. 把从键盘上输入的一批整数(以 -1 作为终止输入的标志)保存到文本文件“a:xxk1.dat”中。

```
#include <iostream >
#include <fstream. h >
#include <stdlib. h >
void main()
{
    ofstream fout("a:xxk1. dat");
    if (_____) {
        cout << "文件没有打开!" << endl;
        exit(1);}
    int x;
    cin >> x;
    while (x! = -1) {
        fout << x << ' ';
        cin >> x;
    }
    _____; //关闭文件
}
```

47. #include <iostream. h >

```
using namespace std;
_____
_____
{ T tmp = x * x + y * y + x * y;
return tmp;
}
```



```
int main() {
    int x1 = 1, y1 = 4;
    float x2 = 1.1, y2 = 2.2;
    double x3 = 2.0, y3 = 3.1;
    cout << fun(x1, y1) << endl;
    cout << fun(x2, y2) << endl;
    cout << fun(x3, y3) << endl;
    return 0;
}
```

48. 完成程序,使其输出结果为 79

```
#include <iostream. h >
using namespace std;
int main() {
    int a[6] = {23,15,64,33,40,58};

    _____
    s1 = s2 = a[0];
    for( int *p = a + 1; p < a + 6; p++) {
        if( s1 > *p) s1 = *p;
        if _____;
    }
    cout << s1 + s2 << endl;
    return 0;
}
```

49. #include <iostream. h >

```
class Point
{
    int X, Y;
public:
    Point( int x = 0, int y = 0)
        { X = x; Y = y; Countp++; }
    Point( Point &p)
        { X = p. X; Y = p. Y; Countp++; }
    ~ Point() { Countp--; }
    _____;
    static int Countp;
    void display() { cout << X << " , " << Y << " , " ; }
};
```



```

Point myfun( Point p1 , Point p2 , Point p3)
{ Point tmp(p1.X + p2.X + p3.X, p1.Y + p2.Y + p3.Y);
  return tmp;
}

int Point::Countp = 0;

void main()
{ Point pp0, pp1(1,2), pp2(1);
  Point p = myfun(pp0, pp1, pp2);
  p.display();
  _____// 输出 Countp 的值
}

```

50. 完成程序,使其结果为

x = 6, y = 10

x = 11, y = 10

```
#include <iostream.h>
```

```
using namespace std;
```

```
class Sample{
```

```
private:
```

```
    int x;
```

```
    _____;
```

```
public:
```

```
    Sample(int a);
```

```
    void print();
```

```
};
```

```
Sample::Sample(_____)
```

```
{ x = a; y = x++; }
```

```
void Sample::print()
```

```
{ cout << "x = " << x << " , y = " << y << endl; }
```

```
int Sample::y = 25;
```

```
void main()
```

```
{
```

```
    Sample s1(5);
```

```
    Sample s2(10);
```

```
    s1.print();
```

```
    s2.print();
```

```
}
```



五、程序分析题:本大题共2小题,每小题5分,共10分。阅读程序后,写出程序的正确运行结果。

51.

```
#include <iostream>
using namespace std;
void Result(char ch) {
    switch(ch) {
        case 'A': case 'a': cout << "well!"; break;
        case 'B': case 'b': cout << "good!"; break;
        case 'C': case 'c': cout << "pass!"; break;
        default: cout << "bad!"; break;
    }
}
void main()
{
    char a1 = 'b', a2 = 'C', a3 = 'f';
    cout << "Results:" << endl;
    Result(a1); Result(a2); Result(a3); Result('A');
}
```

52. #include <iostream.h>

```
using namespace std;
class Sample {
protected:
    int x;
public:
    Sample() { x = 0; }
    Sample(int val) { x = val; }
    void operator++() { x++; }
};
class Derived: public Sample {
    int y;
public:
    Derived(): Sample() { y = 0; }
    Derived(int val1, int val2): Sample(val1) { y = val2; }
    void operator--() { x--; y--; }
```



```

void disp() {
    cout << "x = " << x << ", y = " << y << endl; }
};

void main() {
    Derived d(3,5);
    D. disp();
    d++;
    D. disp();
    d--;
    d--;
    D. disp();
}

```

六、程序设计题:本大题共 1 小题,每小题 10 分,共 10 分。

53. 设计一个圆类 circle 和一个桌子类 table。circle 类包含私有数据成员 radius 和求圆面积的成员函数 getarea(); table 类包含私有数据成员 height 和返回高度的成员函数 getheight()。roundtable 类继承所有上述类的数据成员和成员函数,添加了私有数据成员 color 和相应的成员函数。其中,main 函数已给出。请完成程序的其他部分。

```

void main()
{
    roundtable rt(0.8,1.2,"黑色");
    cout << "圆桌属性数据" << endl;
    cout << "高度:" << rt. getheight() << "米" < < endl;
    cout << "面积:" << tr. getarea() << "平方米" < < endl;
    cout << "颜色" << rt. getcolor() << endl;
}

```

