

# 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)

在每小题列出的四个备选项中只有一个是符合题目要求的,请将其选出并将“答题纸”的相应代码涂黑。错涂、多涂或未涂均无分。

1. 在 C++ 语言中用于注释的标识符是  
A. &                      B. //                      C. \*                      D. /
2. 编写 C++ 程序一般需经过的几个步骤依次是  
A. 编辑、调试、编译、连接                      B. 编译、调试、编辑、连接  
C. 编译、编辑、连接、运行                      D. 编辑、编译、连接、运行
3. 要求通过函数来实现一种不太复杂的功能,并且要求加快执行速度,选用  
A. 内联函数              B. 重载函数              C. 内部函数              D. 函数模板
4. 设  $x=8$ , 执行语句  $y=x++$ , 则  $y$  等于  
A. 6                      B. 7                      C. 9                      D. 8
5. 以下运算符中,优先级最高的是  
A. \*                      B. ->                      C. &&                      D. >=
6. 关于 new 运算符的下列描述中,错误的是  
A. 它可以用来动态创建对象和对象数组  
B. 使用它创建的对象和对象数组可以使用运算符 delete 删除  
C. 使用它创建对象时要调用构造函数  
D. 使用它创建对象数组时必须指定初始值
7. 存储以下数据,占用存储字节最多的是  
A. 0                      B. '0'                      C. "0"                      D. 0.0
8. 关于类和对象不正确的说法是  
A. 类是一种类型,它封装了数据和操作  
B. 对象是类的实例  
C. 一个类的对象只有一个  
D. 一个对象必属于某个类
9. 下面有关构造函数的描述中,正确的是  
A. 构造函数可以带有返回值  
B. 构造函数的名字与类名完全相同  
C. 构造函数必须带有参数  
D. 构造函数必须定义,不能缺省

10. 假定 a 为一个整型数组名,则元素 a[4] 的字节地址为  
A. a + 4                  B. a + 8                  C. a + 16                  D. a + 32
11. 类中能访问静态成员的函数是  
A. 虚函数                  B. 静态成员函数          C. 构造函数                  D. 析构函数
12. 下列说明语句中正确的是  
A. int a, &ra = a, &&ref = ra                  B. int &refa[10]  
C. int a, &ra = a, & \* refp = &ra                  D. int \* pi, \* &pref = pi
13. 关于对象概念的描述中,说法错误的是  
A. 对象就是 C 语言中的结构体  
B. 对象代表着正在创建的系统中的—个实体  
C. 对象是类的一个变量  
D. 对象之间的信息传递是通过消息进行的
14. 有关多态性说法不正确的是  
A. C++ 语言的多态性分为编译时的多态性和运行时的多态性  
B. 编译时的多态性可通过函数重载实现  
C. 运行时的多态性可通过模板和虚函数实现  
D. 实现运行时多态性的机制称为动态多态性
15. 关于友元函数的描述中,错误的是  
A. 友元函数不是成员函数  
B. 友元函数只能访问类中私有成员  
C. 友元函数破坏隐藏性,尽量少用  
D. 友元函数说明在类体内,使用关键字 friend
16. 类中定义的成员默认访问属性为  
A. public                  B. private                  C. protected                  D. friend
17. 拷贝构造函数的参数是  
A. 某个对象名                  B. 某个对象的成员名  
C. 某个对象的引用名                  D. 某个对象的指针名
18. 关于类模板的说法正确的是  
A. 类模板的主要作用是生成抽象类  
B. 类模板实例化时,编译器将根据给出的模板实参生成一个类  
C. 在类模板中的数据成员具有同样类型  
D. 类模板中的成员函数没有返回值
19. 下列有关重载函数的说法中正确的是  
A. 重载函数必须具有不同的返回值类型  
B. 重载函数参数个数必须相同  
C. 重载函数必须有不同的形参列表  
D. 重载函数名可以不同
20. 当一个类的某个函数被说明为 virtual 时,在该类的所有派生类中的同原型函数  
A. 都是虚函数                  B. 只有被重新说明时才是虚函数  
C. 都不是虚函数                  D. 只有被重新说明为 virtual 时才是虚函数

## 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 小题,每小题 1 分,共 20 分)

21. C 语言中 scanf 命令在 C++ 中改用:\_\_\_\_\_。
22. 假定类 AB 中有一个公用属性的静态数据成员 static int bb;在类外不通过对象名给该成员 bb 赋值为 10 的写法:\_\_\_\_\_。
23. 在类内部定义的公有(public)数据可以在类外进行存取,但\_\_\_\_\_和私有(private)数据不能在类外直接存取。
24. 在 C++ 中,虚函数帮助实现了类的\_\_\_\_\_。
25. 如果要把类 B 的成员函数 void fun()说明为类 A 的友元函数,则应在类 A 中加入语句\_\_\_\_\_。
26. 执行下列程序 double a = 3.1415926, b = 3.14; cout << setprecision(5) << a << ", " << setprecision(5) << b << endl; 程序的输出结果是\_\_\_\_\_。
27. 在 C++ 中,访问一个指针所指向的对象的成员所用的指向运算符是\_\_\_\_\_。
28. 基类的公有成员在派生类中的访问权限由\_\_\_\_\_决定。
29. 由 const 修饰的对象称为\_\_\_\_\_。
30. 如果通过同一个基类派生一系列的类,则将这些类总称为\_\_\_\_\_。
31. 为了实现运行时的多态性,派生类需重新定义基类中的\_\_\_\_\_。
32. 在 C++ 中有两种参数传递方式:传值和传\_\_\_\_\_。
33. C++ 中 ostream 的直接基类\_\_\_\_\_。
34. 已知 int \*p = NULL,使用 new 为指针 p 申请一个存储大小为 10 的存放 int 型的空间,代码为\_\_\_\_\_。
35. 在编译指令中,宏定义使用\_\_\_\_\_指令。
36. 每个对象都是所属类的一个\_\_\_\_\_。
37. 函数重载时,编译系统会根据\_\_\_\_\_或形参的个数来区分。
38. 静态成员函数、友元函数、构造函数和析构函数中,不属于成员函数的是\_\_\_\_\_。
39. 如果要把类 B 的成员函数 void fun()说明为类 A 的友元函数,则应在类 A 中加入语句\_\_\_\_\_。
40. 局部对象和全局对象中,在同一程序中\_\_\_\_\_生存期最长。

### 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,请用下划线标出错误所在行并指出错误原因。

41. #include <iostream>

using namespace std;

class base{

int \*p;

public:

base (int a){

p = &a;

}

int get(){

return p;

}

};

int main()

{

base b(3);

cout << b.get();

}

42.

#include <iostream>

using namespace std;

class base{

int a;

int get(){

return a;

}

void setValue(int b){

a = b;

}

int geta(){

return a;

}

};

int main()

{

base b;

b.setValue(7);

return 0;

}

43. #include <iostream>  
using namespace std;  
class A {  
 static int x;  
 int y;  
public:  
 A(int a,int b) {  
 x = a;  
 y = b;  
 }  
 int get() {  
 return x + y;  
 }  
};  
x = 5;  
int main() {  
 A a(1,2);  
 cout << a.get() << endl;  
 return 0;  
}

44.  
#include <iostream>  
using namespace std;  
class A {  
 int x;  
public:  
 A(int a) {  
 x = a;  
 }  
 void set(int a) {  
 x = a;  
 }  
 void get() {  
 cout << x << endl;;  
 }  
};  
int main() {  
 const A a(4);  
 a.set(6);  
 a.get();  
 a.set(10);  
 a.get();  
 return 0;  
}

```

45. #include <iostream>
    using namespace std;
    class base{
        int a,b;
    public:
        void setzero() {
            x = 0; y = 0;
        }
        void show() {
            cout << x << " " << y << endl;
        }
    };
    int main() {
        base b;
        b.setzero(0,0);
        return 0;
    }

```

#### 四、完成程序题(本大题共 5 小题,每小题 4 分,共 20 分)

46. 用简单选择法对 10 个数排序,使其从小到大排列。

```

#include"stdafx.h"
#include <iostream>
using namespace std;
int main( void)
{
    int i;
    int num[ 10 ],t,j;
    for(_____)
        cin >> num[ i ];
    for(i = 0; i < 9; i ++ )
        for(j = i + 1; j < 10; j ++ )
            if( num[ i ] > num[ j ])
            {
                t = num[ i ];
                num[ i ] = num[ j ];
                _____;
            }
    for(i = 0; i < 10; i ++ )
        cout << num[ i ] << " ";
}

```

47. 打印以下图案:

```
* * * * *
  * * * * *
    * * * *
      * * *
        *
```

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main( void)
{
    int j,n;
    for(_____)//i 控制行号
    {
        for(j = 1;j < i;j ++)//j 控制空格
            cout << " ";
        for(_____)//n 控制星号
            cout << " * ";
        cout << endl;
    }
}
```

48. 任意输入 10 个同学的成绩,计算其平均成绩。要求用函数 average() 计算平均成绩,主函数输入数据并输出结果。

```
#include "stdafx.h"
#include <iostream>
using namespace std;
void main( void)
{
    float average( float a[ ] );
    float score[ 10 ];
    for(_____)
    {
        cin >> score[ i ];
    }
    cout << "average:" << average( score ) << endl;
}

float average( float a[ ] )
{
    float sum = 0;
    for( int i = 0; i < 10; i ++ ) {
        _____;
    }
    return ( sum/10 );
}
```

49. 编一个函数 to\_lower(), 实现将字符串中的大写字母转换成相应小写字母。主函数输入数据并输出结果。

```
#include "stdafx.h"
#include <iostream>
_____;
void main( void)
{
    void to_lower( char a[ ] );
    char str[ 10 ];
    cin >> str;
    to_lower( str );
    cout << str << endl;
}
void to_lower( char a[ ] )
{
    for( int i=0; i<10&& a[i] != '\0'; i++)
        if( _____ )
            a[i] += 32;
}
```

50. 下面程序用来求直角三角形斜边长度。

```
#include <iostream.h>
#include <math.h>
class Line;
class Point
{ private:
    double x,y;
    _____
public:
    Point( double i=0, double j=0)
    { x=i; y=j; }
    Point( Point &p)
    { x=p.x; y=p.y; }
};
class Line
{ private:
    Point p1,p2;
public:
    Line( Point &p1, Point &p2): _____ {}
    double GetLength( );
};
double Line::GetLength( )
{ double dx=p2.x-p1.x;
  double dy=p2.y-p1.y;
  return sqrt(dx*dx+dy*dy);
}
void main()
{ Point p1,p2(6,8);
  Line L1(p1,p2);
  cout << L1.GetLength( ) << endl;
}
```



五、程序分析题(本大题共2小题,每小题5分,共10分)

51. 给出下面程序输出结果。

```
#include <iostream.h>

class Base
{ private:
int Y;
public:
Base(int y=0) { Y = y; cout << "Base(" << y << ") \n"; }
~ Base() { cout << " ~ Base() \n"; }
void print() { cout << Y << " "; }
};

class Derived:public Base
{ private:
int Z;
public:
Derived (int y, int z):Base(y)
{ Z = z;
cout << "Derived(" << y << ", " << z << ") \n";
}
~ Derived() { cout << " ~ Derived() \n"; }
void print()
{ Base::print();
cout << Z << endl;
}
};

void main()
{ Derived d(10,20);
d. print();
}
```

52. 给出下面程序输出结果。

```
#include "iostream.h"

void main()
{
int x,y,z;
x = 3;
y = x ++ ;
z = ++ x;
cout << "x =" << x << " ; y =" << y << " ; z =" << z << endl;
}
```

## 六、程序设计题(本大题共 10 分)

53. 写一个程序,定义一个抽象类 Shape,由它派生 3 个类:Square(正方形)、Trapezoid(梯形)和 Triangle(三角形)。用虚函数分别计算几种图形面积、并求它们的和。要求用基类指针数组,使它每一个元素指向一个派生类对象。

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

```
class Shape
```

```
{ public:
```

```
virtual double area() const = 0;
```

```
};
```