

课程代码:04737

- ## 选择题部分

- 给出函数声明: `void func(int, int y=1)`; 下列函数调用中, 错误的是
A. `func()`; B. `func(0)`;
C. `func(1)`; D. `func(2, 3)`;
- 下列关于 C++ 的程序结构的叙述中, 错误的是
A. 一个程序中必须有一个主函数
B. 一个程序的所有代码必须写到一个文件中
C. 执行到主函数中的 `return` 语句, 程序结束
D. 执行到主函数最后面的括号 `}`, 程序结束
- 下列关于类的叙述中, 错误的是
A. 类体可以为空, 没有成员变量和成员函数
B. 类成员的默认访问权限是私有
C. 只有类的成员函数才能访问类中的私有成员
D. 类是一种自定义数据类型
- `void f()` 是类 A 的非静态成员函数, `a` 是类 A 的对象, 下列说法中正确的是
A. `a.f()` 也可以写成 `A::f()`
B. `a.f()` 也可以写成 `a.A::f()`
C. 编译器为对象 `a` 分配的存储空间中包含了成员函数 `f()` 的存储空间
D. 编译器会为类 A 和对象 `a` 分别分配存储空间
- 在一个类中, 不能有多的是
A. 构造函数 B. 析构函数
C. 成员函数 D. 静态成员函数
- 下列关于运算符重载的叙述中, 错误的是
A. 只能重载一元或二元运算符
B. 运算符可以被重载为类的成员函数或全局函数
C. 运算符重载时, 可以对运算符优先级、结合性等进行重新定义
D. 不能创建新的运算符

7. 对于类 A 和类 B, 如果 A(B()); 能够通过编译, 那么最有可能是下面哪一种情况
- A. 类 A 中定义了构造函数 A(const B *)
 - B. 类 B 是类 A 的派生类, 且定义了无参构造函数
 - C. 类 B 是类 A 的友元类
 - D. 类 A 中存在类型转换函数 B()
8. 如果类 A 的构造函数都是 protected 的, 类 B 以公有方式继承类 A, 那么
- A. 在 main()函数中, A a; 语句正确
 - B. 在 main()函数中, B b; 语句正确
 - C. 在类 B 中可以实例化类 A 的对象
 - D. 在类 A 中可以实例化类 B 的对象
9. 类 A 没有基类, 也没有任何数据成员, 下列叙述中正确的是
- A. 类 A 一定是抽象类, 不能实例化
 - B. 类 A 可以实例化, 但其对象的存储空间大小为 0
 - C. 如果类 A 中只有虚函数, 则其对象的存储空间大小也不会为 0
 - D. 如果类 A 中没有虚函数, 则其对象的存储空间大小取决于成员函数的多少
10. 派生类可以从基类继承
- A. 私有成员变量
 - B. 拷贝构造函数
 - C. 析构函数
 - D. 友元函数
11. 类 A 派生了类 B, A a; A *pa; B b; B *pb; 下列语句错误的是
- A. pa=&a;
 - B. pa=&b;
 - C. pb=&a;
 - D. pb=&b;
12. 下列关于纯虚函数和空函数的叙述, 错误的是
- A. 纯虚函数和空函数都没有函数体
 - B. 只包含纯虚函数的类, 不能实例化
 - C. 只包含空函数的类, 可以实例化
 - D. 包含纯虚函数和空函数的类, 可以派生出新类
13. 下列可以定义为虚函数的是
- A. 构造函数
 - B. 析构函数
 - C. 静态函数
 - D. 友元函数
14. 下列关于 cout 的叙述, 正确的是
- A. cout 不能向文件输出数据
 - B. cout 是流类 ostream 的对象
 - C. cout 不支持重定向
 - D. cout 只能输出字符类型数据
15. 使用 cin 从键盘连续读入数据时, 可以中断输入流的组合键是
- A. Ctrl+Z
 - B. Ctrl+C
 - C. Ctrl+U
 - D. Ctrl+X
16. 打开文件时需要明确文件的使用方式, 下列哪一个不是有效的文件使用方式
- A. 只读
 - B. 追加
 - C. 只写
 - D. 覆盖



17. 如果一个文件只能顺序读取，其原因可能是
 - A. 文件的格式决定的
 - B. 文件的存储介质决定的
 - C. 文件的大小决定的
 - D. 文件的内容决定的
18. 模板的作用主要是
 - A. 提高代码运行效率
 - B. 实现多态
 - C. 实现封装
 - D. 降低编码工作量
19. 通过类模板，可以实例化多个不同的类，这些类之间的关系是
 - A. 相互独立
 - B. 继承关系
 - C. 组合关系
 - D. 包含关系
20. 非抽象类一般被称为具体类。若具体类 A 定义如下：

```
class A { Sub subs[100]; }
```

下列叙述错误的是
 - A. 类 A 的实例化对象大小不为 0
 - B. Sub 一定是具体类
 - C. 类 A 的子类一定是具体类
 - D. 类 A 的拷贝构造函数无需自定义

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 15 小题，每空 1 分，共 15 分。

21. 在面向对象程序设计中，类是现实世界中客观事物的_____。
22. 在一个由多个文件组成的 C++ 项目中，类 A 的定义放在头文件 a.h 中，如果有多个 .cpp 文件都通过 #include "a.h" 用到类 A，那么在链接时会发生_____错误。
23. 使用 private 修饰类成员，有利于隐藏程序的_____。
24. 在同一个作用域内，不能声明_____的标识符。
25. 使用 new 实例化对象时，系统将自动调用该对象的_____。
26. 静态局部变量具有局部作用域和全局_____。
27. 定义在所有类和函数之外的变量具有_____作用域。
28. 一个类的常量成员函数，不能_____该类的成员变量。
29. 通过_____机制，可以扩充和完善已有的类以适应新的需求。
30. 类的静态成员函数中不能使用_____指针。
31. 类 A 不能多次成为类 B 的直接基类，但可以多次成为类 B 的_____。
32. 要实现动态多态，除了继承之外，必须使用_____。
33. 多重继承指的是一个派生类同时有一个以上的_____。
34. “张三是一个厨师”体现了一种_____关系。
35. 进行函数重载时，新函数和已有函数的参数个数和类型完全相同，那么新函数必需用_____修饰。



三、程序填空题：本大题共 5 小题，每小题 4 分，共 20 分。请按试题顺序和空格顺序在答题卡（纸）指定位置上填写答案，错填、不填均无分。

36. 程序完成后的执行结果为：90

```
#include <iostream>
#include<string>
using namespace std;

class Student
{
private:
    string name;
    int score;
public:
    Student(string n)
    {
        name = n;
    }
    _____ (1) _____
    {
        score = s;
    }
    int getScore()
    {
        return score;
    }
    int averageScore(Student stu)
    {
        _____ (2) _____
    }
};

int main()
{
    Student stu1("tom"), stu2("jerry");
    stu1.setScore(100);
    stu2.setScore(80);
    cout<<stu1.averageScore(stu2);
    return 0;
}
```



37. 程序完成后的执行结果为: x=6,y=8,z=10

```
#include <iostream>
using namespace std;

class TwoCord
{
private:
    int x, y;
public:
    TwoCord(int x, int yy)
    {
        (1)
        y = yy;
    }
    void display()
    {
        cout<<"x="<<x<<"",y="<<y;
    }
};

class ThreeCord
{
    int z;
public:
    ThreeCord(int x, int yy, int zz):TwoCord(x, yy)
    {
        z = zz;
    }
    void display()
    {
        (2)
        cout<<"",z="<<z;
    }
};

int main()
{
    ThreeCord c(6, 8, 10);
    c.display();
    return 0;
}
```



38. 程序完成后的运行结果为: 100

```
#include <iostream>
using namespace std;
class Singleton
{
public:
    int gId;
    static Singleton* getInstance()
    {
        if(instance == NULL) instance = new Singleton();
        return instance;
    }
private:
    Singleton() { }
    static Singleton* instance;
};
_____ (1) _____
int main()
{
    _____ (2) _____
    p->gId = 100;
    cout<<p->gId;
    return 0;
}
```

39. 程序完成后的运行结果为: 10 30 50

```
#include <iostream>
using namespace std;
class Container
{
public:
    Container(int defVal=10)
    {
        for(int i=0; i<3; i++) _____ (1) _____
    }
    _____ (2) _____ { return item[index]; }
    int Item(int index) const { return item[index]; }
private:
    int item[3];
};
```



```

int main()
{
    Container vc;
    const Container cc(20);
    int i;
    for(i=0; i<3; i++) vc.Item(i) += i * cc.Item(i);
    for(i=0; i<3; i++) cout<<vc.Item(i)<<" ";
    return 0;
}

```

40. 程序完成后的运行结果为: (4,6)

```

#include <iostream>
using namespace std;

```

```

class TwoCord
{

```

```

    int x, y;

```

```

public:

```

```

    TwoCord(int xx, int yy)
    {

```

```

        x=xx;

```

```

        y = yy;
    }

```

```

    _____ (1) _____ (const TwoCord &c1, const TwoCord &c2)
    {

```

```

        _____ (2) _____
    }

```

```

    void display()
    {

```

```

        cout<<"("<<x<<" "<<y<<"")";
    }

```

```

};

```

```

int main()
{

```

```

    TwoCord c1(1,2), c2(3,4);

```

```

    c1 = c1 + c2;

```

```

    c1.display();

```

```

    return 0;
}

```



四、程序分析题：本大题共 5 小题，每小题 6 分，共 30 分。阅读程序后，填写程序的正确运行结果。

41. #include <iostream>
using namespace std;

```
class Triangle
{
public:
    void setRows(int rows) { _rows = rows; }
    void create()
    {
        for(int i=0; i<_rows; i++)
        {
            for(int j=0; j<=i; j++)
                _array[i][j] = _array[i-1][j-1]+_array[i-1][j];
            _array[i][0] = _array[i][i] = 1;
        }
    }
    void display()
    {
        for(int i=0; i<_rows; i++)
        {
            for(int j=0; j<=i; j++) cout<<_array[i][j]<<" "
            cout<<endl;
        }
    }
private:
    int _rows;
    int _array[100][100];
};

int main()
{
    Triangle ta;
    ta.setRows(5);
    ta.create();
    ta.display();
    return 0;
}
```




```

42. #include <iostream>
    using namespace std;
    class A
    {
    public:
        A(int n): val(n) { }
    protected:
        int val;
    };
    class B : public A
    {
        B *pb;
    public:
        B(int n):A(n) { pb = (n>0? new B(n-1):0); }
        ~B() { delete pb; }
        void display()
        {
            cout<<val<<endl;
            if(pb) pb->display();
        }
    };
    int main()
    {
        B b(4);
        b.display();
        return 0;
    }

```

```

43. #include <iostream>
    using namespace std;
    class A
    {
    public:
        virtual ~A() { }
        void f() { g(); h(); }
        virtual void h() { cout<<"A::h()"<<endl; }
    protected:
        void g() {cout<<"A::g()"<<endl; }
    };

```



```

class B
{
public:
    B(int n):val(n) {}
protected:
    int val;
};
class C : public A, private B
{
public:
    C(int n1, int n2):B(n2),num(n1) {}
    virtual ~C() {}
    virtual void h()
    {
        cout<<"num="<<num<<endl;
        cout<<"val="<<val<<endl;
    }
protected:
    void g() { cout<<"C::g()"<<endl; }
private:
    int num;
};
int main()
{
    C c(1,2);
    c.f();
    return 0;
}

```

44. #include <iostream>
using namespace std;

```

class A
{
public:
    virtual void f() { cout<<"A::f()"<<endl; }
    void g() { cout<<"A::g()"<<endl; };
};

```



```

class B : public A
{
public:
    void f() { cout<<"B::f()"<<endl; }
    virtual void g() { cout<<"B::g()"<<endl; };
};
class C : public B
{
public:
    void f() { cout<<"C::f()"<<endl; }
    void g() { cout<<"C::g()"<<endl; };
};
int main()
{
    C c;
    A *pa = &c;
    B *pb = &c
    pa->f();
    pa->g();
    pb->f();
    pb->g();
    return 0;
}

```

45. #include <iostream>
using namespace std;

```

int main()
{
    cout.width(5);
    cout.fill('#');
    cout<<135<<endl;
    cout.fill('0');
    cout<<135<<endl;
    cout.width(10);
    cout.setf(ios::left);
    cout<<135.79<<endl;
    return 0;
}

```



五、程序设计题：本大题共 2 小题，第 46 小题 5 分，第 47 小题 10 分，共 15 分。

46. 某程序需要定义一个用户类，要求如下：

- (1) 用户类的成员变量包括 id 和密码，其中 id 为常量，在用户对象生存期间不能修改，密码不能通过用户对象直接访问。
- (2) 用户对象可以修改自己密码，但需要先验证旧密码是否正确。
- (3) 保证如下 main() 函数的执行结果为：false true

```
int main()
{
    User user(1001, "123456");
    cout<<boolalpha<<user.changePwd("123", "863973")<<" ";
    cout<<boolalpha<<user.changePwd("123456", "863973");
    return 0;
}
```

47. 定义一个日志管理类，实现如下功能：

- (1) 将单条日志信息追加到二进制日志文件 e:\log.dat 中。日志信息包括：日志序号（整型）、日志类型（整型）和日志内容（字符串，长度不大于 100 字节）。
- (2) 计算日志文件中日志的数量。
- (3) 从日志文件中读取最近的 10 条日志，显示到屏幕上。如果不足 10 条，则全部显示。

