

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

第二部分 非选择题

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;
    }
};

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 1 小题 5 分，第 2 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。