

绝密★启用前

2022 年 4 月高等教育自学考试全国统一命题考试

## C++程序设计

(课程代码 04737)

注意事项:

1. 本试卷分为两部分, 第一部分为选择题, 第二部分为非选择题。
2. 应考者必须按试题顺序在答题卡(纸)指定位置上作答, 答在试卷上无效。
3. 涂写部分、画图部分必须使用 2B 铅笔, 书写部分必须使用黑色字迹签字笔。

## 第一部分 选择题

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. 下列对 C++ 程序的描述中, 不正确的是  
A. 继承于 C 语言  
B. 必有且仅有一个主函数  
C. 以 .cpp 作为文件扩展名  
D. 程序从 #include 指令开始执行
2. 在下面循环语句中循环体执行的次数为  
for(int i=1; i<n; i++)  
if(i>n/2) break;  
A. n/2  
B. n/2+1  
C. n-1  
D. n/2-1
3. 有函数原型 "int fun(int,float);", 以下语句正确的是  
A. int a=fun();  
B. cout<<fun(1,3.14)  
C. double a=fun();  
D. cout<<fun(1+3.13)
4. 关于封装, 下列说法中不正确的是  
A. 经过封装, 对象的属性都是不可见的  
B. 经过封装, 每个对象都成为相对独立的实体  
C. 经过封装, 对象的全部属性和操作结合在一起, 形成一个整体  
D. 经过封装, 一个对象的实现细节被尽可能地隐藏起来(不可见)
5. 假定 A 为一个类, abc 为该类的一个成员函数, 则参数表中隐含的第一个参数为  
A. abc  
B. \*this  
C. this  
D. this&
6. 将一个类 A 或函数 f() 说明为另一个类 B 的友元, 则 A 或 f() 可直接访问类 B 的  
A. 只能是公有成员  
B. 具有任何权限的成员  
C. 只能是保护成员  
D. 除私有成员之外的任何成员

7. 从一个基类派生出的各个类的对象之间  
A. 共享部分数据成员和函数成员  
B. 共享所有数据成员, 每个对象还包含基类的所有属性  
C. 共享部分数据成员, 每个对象还包含基类的所有属性  
D. 不共享任何数据成员, 但每个对象还包含基类的所有属性
8. 表示式 a+b 中的 "+" 是作为成员函数重载的运算符, 若采用运算符函数调用格式, 则可表示为  
A. operator(a+b)  
B. a.operator+(b)  
C. operator+(a,b)  
D. b.operator+(a)
9. 函数重载是指  
A. 两个以上的函数名字不同, 但形参的个数或类型相同  
B. 两个以上的函数取相同的函数名, 而且函数的返回类型相同  
C. 两个以上的函数取相同的函数名, 但形参的个数或类型不同  
D. 两个以上的函数取相同的名字且有相同的参数个数, 但形参的类型不同
10. 如果一个类的成员函数 print() 不修改类的数据成员值, 则应将其声明为  
A. void print() const;  
B. void print(const);  
C. void const print();  
D. const void print();
11. 从 "student wanghao("王浩","M",22); wanghao.id("2005131000");" 中, 可判断 id 是  
A. 私有成员数据  
B. 私有成员函数  
C. 公有成员数据  
D. 公有成员函数
12. 若一个类的成员函数前用 static 关键字修饰, 则该成员函数  
A. 没有 this 指针  
B. 只能用对象名来调用  
C. 可以被声明为 const  
D. 可以访问该类的所有成员
13. C++ 中实现接口重用的是  
A. 虚函数  
B. 内联函数  
C. 重载函数  
D. 模板函数
14. C++ 中的虚基类机制能够保证  
A. 限定基类只经过一条路径派生出派生类  
B. 当一个类多次间接从基类派生以后, 其基类只被一次继承  
C. 允许基类经过多条路径派生出派生类, 派生类也就能多次继承该基类  
D. 当一个类多次间接从基类派生以后, 派生类对象能保留多份间接基类的成员
15. 将标准输入重定向到文件 input.dat 的语句是  
A. freopen(stdin, "r", "input.dat");  
B. freopen(stdout, "r", "input.dat");  
C. freopen("input.dat", "r", stdin);  
D. freopen("input.dat", "r", stdout);
16. 语句 "cout<<setbase(8)<<16<<setw(3)<<18<<endl;" 的输出结果是  
A. 1618  
B. 16 18  
C. 2022  
D. 20 22
17. 关于文本文件和二进制文件的描述正确的是  
A. 二进制文件数据兼容性好  
B. 二进制文件以文本形式存储  
C. 文本文件具有较高的兼容性  
D. 文本文件便于对数据随机访问

18. 关于对象概念的描述中, 错误的是  
A. 对象就是 C 语言中的结构变量  
B. 对象之间的信息传递是通过消息进行的  
C. 对象代表着正在创建的系统中的实体  
D. 对象是一个状态和操作 (或方法) 的封装体
19. 关于析构函数的叙述中, 正确的是  
A. 用户必须定义类的析构函数  
B. 析构函数没有参数, 也没有返回值  
C. 析构函数的定义只能在类体内  
D. 系统在任何情况下都能正确析构对象
20. 假设有类模板声明如下, 则选项中正确的是  
template <typename T>  
class Tclass{int a;  
public:Tclass(int b){a=b;}};  
A. Tclass t(9);  
B. Tclass(double) t(9)  
C. Tclass (0.9>t(9));  
D. Tclass<double> t(9);

## 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. C++ 语言支持的两种多态性分别是编译时的多态性和\_\_\_\_\_的多态性。
22. 当执行 cout 语句输出 endl 数据项时, 将使 C++ 显示输出屏幕上的光标从当前位置移动到\_\_\_\_\_的开始位置。
23.  $x > 0 \vee y == 5$  的相反表示式为\_\_\_\_\_。
24. 如果类 A 继承类 B, 则类 A 称为\_\_\_\_\_, 类 B 称为基类。
25. 要想访问基类的 private 成员变量, 只能通过基类提供的\_\_\_\_\_进行间接访问。
26. strlen("apple") 的值为\_\_\_\_\_。
27. 假定用户为类 AB 定义了一个构造函数 "AB(int aa){a=aa;}", 则该构造函数实现对数据\_\_\_\_\_的初始化。
28. 通过指向基类的指针调用派生类对象的函数时, 采用\_\_\_\_\_调用。
29. 复制构造函数使用一个\_\_\_\_\_初始化创建中的对象。
30. 在 C++ 中, 对于构造函数和析构函数来说, \_\_\_\_\_是不可以重载的。
31. 设有一时间类 Time, 现欲用成员函数方法重载运算符+, 实现一个时间与一个整数的相加运算, 所得结果为一个新时间, 则成员函数的声明语句为\_\_\_\_\_。
32. 在表达式 "x+=y" 中, "+=" 是作为非成员函数重载的运算符, 若是使用显式的函数调用代替直接使用运算符 "+=", 这个表达式还可以表示为\_\_\_\_\_。
33. 假定 a 是一个一维指针数组, 则 a+i 所指对象的地址比 a 大\_\_\_\_\_字节。
34. 如果一个派生类只有一个唯一的基类, 则这样的继承称为\_\_\_\_\_。
35. C++ 虚函数有两种调用方式, 分别是\_\_\_\_\_和多态调用方式。

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. 程序的输出结果为: helo:third

```
#include <iostream>
using namespace std;
int main()
{   int x=3,y=3;
    string s1="first\n",s2="second\n",s3="third\n";
    switch(x%2)
    {
        case 1: _____
        {
            case 0: cout<<s1;
            case 1: cout<<s2; break;
            default: cout<<"hello: ";
        }
        case 2: _____;
    }
    return 0;
}
```

37. 函数 Convert() 将一个数字字符串转换为对应的整数

```
#include <iostream>
using namespace std;
int Convert(char *str);
int main()
{
    char s[10] = "12345";
    int n = Convert(s);
    cout << n << endl;
    return 0;
}
_____
{
    int num=0,digit;
    for(int i=0; i < strlen(str); i++)
    {
        digit = _____ - 48;
        num = num*10 + digit;
    }
    return num;
}
```

38. 程序的输出结果为: 1234

```
class A{
private: int X,Y;
protected:int Z;
public:
    A(int a, int b, int c) { X=a; Y=b; Z=c;}
    int GetX() { return X; }
    int GetY() { return Y; }
};

class B:public A{
private: int K;
public:
    B(int a,int b,int c,int d): A(a,b,c) { K=d;}
    void Show() { cout<<GetX()<<GetY()<<Z<<K<<endl; };
int main()
{
    B b(1,2,3,4);
    _____;
    return 0;
}
```

39. Shape 类是一个表示形状的抽象类, area()是求图形面积的函数, total()是一个通用的用以求不同形状的图形面积总和的函数, Shape 类派生三角形类 Triangle

```
class Shape{
public:
    virtual float area()=0;
};

float total(Shape *s[],int n)
{
    float sum=0.0;
    for(int i=0;i<n;i++)
        _____;
    return sum;
}

_____ {
public:
    Triangle(double h,double w){H=h;W=w;}
    double area() const{return H*W*0.5;}
private:
    double H,W;
};
```

40. 下面的程序将一个已有文件的内容复制到另一个文件中, 请将程序补充完整

```
#include<fstream>
using namespace std;
int main()
{
    _____
    infile.open("d:\\file1.txt",ios::in);
    if(!infile)
        cout<<"file1.txt can't open.\n";
    outfile.open("d:\\file2.txt",ios::out);
    if(!outfile)
        cout<<"file2.txt can't open.\n";
    char str[80]="\0";
    while(!infile.eof())
    {
        infile.read(str,sizeof(str));
        _____
        cout<<str;
    }
    infile.close();
    outfile.close();
    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;
enum BoolConst { False=0 , True=1 };
class Boolean{
public:
    Boolean(BoolConst x = False) { logic = x; }
    void print() const { logic? cout<<" TRUE " : cout<<" FALSE "; }
    friend Boolean operator *(const Boolean & obj1, const Boolean & obj2);
protected: BoolConst logic;
};

Boolean operator* ( const Boolean & obj1, const Boolean & obj2 )
{ return (obj1.logic && obj2.logic) ? Boolean(True) : Boolean(False); }

int main()
{
    Boolean a(False), b(True), c;
    c = a * b;
    a.print(); b.print(); c.print();
}
```

```
42. #include <iostream>
using namespace std;
template<class T>
void func1(T s[])
{   int i;
    for(i=0;i<6;i++)
    {   s[i]=i+1;
        cout<<s[i]<<' ';
    }
}
void func2(double s[], double w[])
{   double sum=0;
    int i,k;
    for(k=0,i=0;i<6;i++)
    {   sum += s[i];
        if((i+1)%3==0)
        {   w[k] = sum/3;
            cout<<w[k]<<' ';
            sum = 0;
            k++;
        }
    }
}
int main()
{   double s[6], w[3];
    func1(s);
    cout<<endl<<"平均值: ";
    func2(s,w);
    return 0;
}
```

```
43. #include <iostream>
using namespace std;
class Samp{
public:
    void SetMuti(int a, int b){   i=a; j=b;   }
    ~Samp(){   cout<<"析构..."<<i<<endl;   }
    int GetMuti(){   return i*j;   }
protected:   int i,j;
};
```

```
int main()
{   Samp *p;
    int N=3;
    p= new Samp[N];
    if(!p)
    {   cout<<"内存分配错误\n";
        return 1;
    }
    for(int i=0;i<N;i++)
        p[i].SetMuti(i,i+1);
    for(int i=0;i<N;i++)
        cout<<"Muti["<<i<<"]值是:"<<p[i].GetMuti()<<endl;
    delete []p;
    return 0;
}
```

```
44. #include <iostream>
using namespace std;
class Computer{
public:
    Computer() { cout << "Computer constructor" << endl; }
    virtual void subscribe() = 0;
    void read() { cout << "Read paper" << endl; }
    ~Computer() { cout << "Computer destructor" << endl; }
};
class Design : public Computer{
public:
    Design() {cout << "Design constructor" << endl; }
    virtual void subscribe() { cout << "Subscribing Design" << endl; }
    void read() {cout << "Reading Design" << endl; }
    ~Design() { cout << "Design destructor" << endl; }
};
int main()
{   Design journal;
    Computer *p_journal= &journal;
    journal.subscribe();
    p_journal->read();
    return 0;
}
```

```
45. #include <iostream>
using namespace std;
class A{
private:
    int n;
public:
    A(int i){ n=i; }
    A operator ++() { n++; }
    A operator ++(int) { n+=2; }
    void Display() { cout<<"n="<<n<<endl; }
};
int main()
{
    A a(2), b(2);
    a++;
    ++b;
    a.Display();
    b.Display();
}
```

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. 根据下面类中 Sum 函数成员的原型和注释写出它的类外定义。

```
#include <iostream>
using namespace std;
class AA{
    int *a, n,MS;
public:
    void InitAA(int aa[],int nn,int ms)
    {
        if(nn>ms){cout<<"Error!"<<endl;return;}
        MS=ms;
        a=new int[MS];
        for(int i=0;i<n;i++)a[i]=aa[i];
    }
    int Sum();//求出并返回数组 a 中前 n 个元素之和
};
```

47. 请声明 Student 类，类中包含数据成员姓名和分数，成员函数 compare（用于比较两名学生的分数）。在主函数中输入 n 名学生的信息后，输出其中最高分和最低分学生的姓名及分数。