

2023 年 10 月高等教育自学考试

C++ 程序设计试题

课程代码:04737

1. 请考生按规定用笔将所有试题的答案涂、写在答题纸上。

2. 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。

选择题部分

注意事项:

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。

- 对 C++ 语言的描述正确的是
 - 不再演化改进
 - 前身是低级语言 C
 - 不是面向用户的语言
 - 属于高级程序设计语言
- 下列选项中可以取代语句“`namespace { int x,y; }`”的是
 - `int x; int y;`
 - `int x,y;`
 - `static int x,y;`
 - `int x; y;`
- 把 `v1 (int v1;)` 和 `v2 (char v2[] = "ABCDW";)` 分别作为第一参数和第二参数来调用函数 `f1`, 已知函数 `f1` 的原形是“`void f1(int *a,char *b);`”, 则正确的调用语句是
 - `f1(v1,v2)`
 - `f1(&v1,v2)`
 - `f1(v1,&v2)`
 - `f1(&v1,&v2)`
- 要发生动态绑定, 则调用虚函数时应使用
 - 类名
 - 对象名
 - 基类的指针
 - 派生类的指针
- 关于类和对象的描述不正确的是
 - 对象是类的实例
 - 类封装了数据和操作
 - 一个类的对象只有一个
 - 一个对象必属于某个类
- 已知: `p` 是一个指向类 `A` 数据成员 `m` 的指针, `A1` 是类 `A` 的一个对象, 如果要给 `m` 赋值为 5, 下列表达式正确的是
 - `A1.p=5`
 - `A1->p=5`
 - `A1.*p=5`
 - `*A1.p=5`
- 在下列函数原型中, 可以作为类 `Test` 的构造函数的是
 - `Test (int);`
 - `int Test ();`
 - `~Test (int);`
 - `void Test (int);`

8. 对静态数据成员特性的理解不正确的是
- A. 说明静态数据成员时要加 `static`
 - B. 静态数据成员不是所有对象所共有的
 - C. 静态数据成员要在类体外进行初始化
 - D. 访问静态数据成员时，可在静态数据成员名前加类名和作用域运算符
9. 对运算符重载的理解不正确的是
- A. 不是所有的运算符都可以进行重载
 - B. 调用运算符函数必须使用关键字 `operator`
 - C. 不可以通过运算符重载创造出新的运算符
 - D. 运算符重载不能改变操作数的个数、运算符的优先级和结合性
10. 为了区别单目运算符“++”或“-”的前置式和后置式运算，在后置式运算符进行重载时，额外添加一个参数，其类型是
- A. `int`
 - B. `char`
 - C. `void`
 - D. `float`
11. 要用派生类的对象直接访问基类的私有成员，以下观点正确的是
- A. 不可能实现
 - B. 可采用保护继承
 - C. 可采用私有继承
 - D. 可采用公有继承
12. 关于派生类构造函数的描述不正确的是
- A. 派生类构造函数通常带有成员初始化列表
 - B. 派生类继承直接基类和所有间接基类的构造函数
 - C. 派生类默认构造函数中隐含包括直接基类的默认构造函数
 - D. 派生类对象的初始化是由基类和派生类的构造函数共同完成的
13. 在下列选项中不属于运行时的多态的是
- A. 根据 `if` 语句的求值决定程序流程
 - B. 根据参数个数、类型调用重载函数
 - C. 根据 `switch` 语句的求值决定程序流程
 - D. 根据基类指针指向对象调用成员函数
14. 关于虚函数的描述正确的是
- A. 虚函数不具有继承性
 - B. 虚函数是一个成员变量
 - C. 静态成员函数可以声明为虚函数
 - D. 在类的继承的层次结构中，虚函数是声明相同的函数
15. 定义基类 A，类 A 派生类 B，类 B 派生类 C，创建类 C 的对象 `cobj`，调用顺序是
- A. 构造函数调用顺序为 ABC
 - B. 构造函数调用顺序为 CBA
 - C. 析构函数调用顺序为 BCA
 - D. 析构函数调用顺序为 CAB
16. 在下列选项中属于流类 `istream` 的对象的是
- A. `cin`
 - B. `cerr`
 - C. `clog`
 - D. `cout`
17. 用于清除基数格式位，设置以十六进制输出的语句是
- A. `cout<<setf(ios::oct);`
 - B. `cout<<setf(ios::dec,ios::basefield);`
 - C. `cout<<setf(ios::hex);`
 - D. `cout<<setf(ios::hex,ios::basefield);`

18. C++程序中进行文件操作时，应包含头文件
 A. math B. fstream
 C. stdlib D. strstrea
19. 实现以读的方式打开 D 盘文件 fl.txt 的语句是
 A. ifstream infile("D:\\fl.txt", ios::in); B. ofstream outfile("D:\\fl.txt", ios::out);
 C. ifstream infile("D:\\fl.txt", ios::in); D. ofstream outfile("D:\\fl.txt", ios::out);
20. 设有函数： $T \max(T x, T y) \{ \text{return } x == y ? x : y ; \}$ ，其中 T 为模板类型，下列相关分析正确的是
 A. 该函数模板有两个模板类型参数
 B. T 所规定的类型不可以是自定义类型
 C. 生成模板函数中的 x 和 y 的类型必须相同
 D. 生成模板函数的参数和返回值的类型可以不同

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，每小题 1 分，共 15 分。

21. 执行“if (x || !x) cout << "ABC"; else cout << "XYZ";”，屏幕上显示的是_____。
22. 函数调用语句“aver(x+(x+y), ++z, sqrt(x+a), max(x, f(y)))”中，实参的个数是_____。
23. 定义一个对象时，系统只为该对象的_____成员开辟内存空间。
24. 将部分成员函数声明为_____，用于提供外界和这个类对象相互作用的接口，从而使其它函数也可以访问处理该类的对象。
25. 可以用_____限定符强制改变访问权限，如指向常量的指针就是在非常量指针声明前使用该限定符。
26. 利用“对象名.成员变量”形式访问的对象成员，仅限于类中声明为公有的成员，若要访问其他成员变量，需要通过_____或者公有成员函数。
27. 若要把类 FriendClass 定义为类 MyClass 的友元类，则应在类 MyClass 的定义中加入语句：_____。
28. C++流中重载了运算符<<，它是一个用于_____的非成员函数。
29. 若将一个二元运算符重载为类的友元函数，其形参个数应该是_____个。
30. 在基类中的_____成员不允许外界访问，但允许派生类的成员访问，这样既有一定的隐藏能力，又提供了开放的接口。
31. 静态成员函数对于每个类来说只有一份代码，所有对象共享这份代码，所以静态成员函数不能是_____。
32. C++进行 I/O 格式控制的方式，一般有使用流操纵符、设置_____和调用成员函数。
33. 语句序列“ifstream infile; infile.open("data.dat");”的功能用一个语句实现是_____。
34. 已知 $\text{int DBL}(\text{int } n) \{ \text{return } n+n; \}$ 和 $\text{long DBL}(\text{long } n) \{ \text{return } n+n; \}$ 是一个函数模板的两个实例，则该函数模板的定义是：“template<typename T> _____ {}”。
35. 若有声明“template <typename T> class Tclass { /*...*/ };”，用语句“Tclass Tobj;”建立一个 Tclass 对象，则该语句出错的原因是_____。

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. 将程序补充完整，使其输出：Time:2023 年 10 月

C++程序设计(11)

高等教育 C++程序设计自学考试

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    string s[]={"2023 年","10 月","_____(1)_____"};
    string c="C++程序设计";
    if(c.empty())
        cout<<"string is NULL"<<endl;
    else
        cout<<"Time:"<<_____(2)_____<<endl;
    cout<<c<<'('<<c.size()<<')'<<endl;
    cout<<s[2].insert(8,c)<<endl;
    return 0;
}
```

37. 程序用来求直角三角形斜边长度，运行结果为 5。

```
#include <iostream>
#include <cmath>
using namespace std;
class Point
{
private:
    double x,y;
    _____(1)_____;
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 &xp1,Point &xp2):p1(xp1),p2(xp2){}
```

```
double GetLength()
```

```
{
```

```
    double dx=p2.x-p1.x;
```

```
    double dy=p2.y-p1.y;
```

```
    return sqrt(dx*dx+dy*dy);
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    Point L1,L2(3,4);//两条直角边的长度
```

```
    Line L3(L1,L2);
```

```
    cout<<_____ (2) _____<<endl;
```

```
    return 0;
```

```
}
```

38. 以下程序输出 hello!hello!, 请将程序补充完整。

```
#include <iostream>
```

```
using namespace std;
```

```
class str
```

```
{
```

```
private:
```

```
    char *st;
```

```
public:
```

```
    str(char *a){ set(a); }
```

```
    _____ (1) _____
```

```
{
```

```
    set(a.st);
```

```
    return *this;
```

```
}
```

```
void show(){cout<<st<<"!";}
```

```
void set(char *s)
```

```
{
```

```
    st=new char[strlen(s)+1];
```

```
    strcpy(st,s);
```

```
}
```

```
};
```

```

int main()
{
    str s1("hello"),s2("world");
    _____(2)_____;
    s1.show();s2.show();
}

```

39. 以下程序输出 name:D Class Derived=80, 请将程序补充完整。

```

#include <iostream>
using namespace std;
class Base
{
public:
    _____(1)_____;
    Base(const char *na){strcpy(name,na);}
};
class _____(2)_____
{
public:
    Derived(const char *nm):Base(nm){}
    void show();
};
void Derived::show(){ cout<<"name:"<<name; }
int main()
{
    Derived d1("D");
    d1.show();
    cout<<" Class Derived="<<sizeof(Derived)<<endl;
}

```

40. 以下程序输出23,10.1,c, 请将程序补充完整。

```

#include <iostream>
using namespace std;
template <typename AT>
_____ (1) _____
{
    return (x>y)?x:y;
}

```

```

int main()
{
    int i1=20,i2=23;
    double d1=10.1,d2=4.1;
    char c1='a',c2='c';
    cout<<max1(i1,i2)<<" ";
    _____(2)_____ ;
    cout<<max1(c1,c2)<<endl;
}

```

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 Time
{
public:
 Time(int h, int m){hours=h; minutes=m;}
 void Time12();
 friend void Time24(Time time);
private: int hours, minutes;
};
void Time::Time12()
{
 if(hours>12) cout<<hours-12<<":"<<minutes<<"PM==";
 else cout<<hours<<":"<<minutes<<"AM==";
}
void Time24(Time time){ cout<<time.hours<<":"<<time.minutes<<"\n";}
int main()
{
 Time T1(23,30), T2(10,45);
 T1.Time12();
 Time24(T1);
 T2.Time12();
 Time24(T2);
}

```

42. #include<iostream>
using namespace std;
class data
{
    int x;
public:
    data(int X){data::x=x;cout<<"class data"<<endl;}
};
class School
{
    data d1;
public:
    School(int x):d1(x){cout<<"School"<<endl;}
};
class Course:public School
{
    data d2;
public:
    Course(int x):School(x),d2(x){cout<<"Course"<<endl;}
};
class Score:public Course
{
public:
    Score(int x):Course(x){cout<<"Score"<<endl;}
};
int main()
{
    Score obj(1);
}

```

```

43. #include<iostream>
using namespace std;
class Math
{
protected:
    int k;
public:
    Math(int n=5):k(n){cout<<k<<endl;}
    ~Math(){cout<<"Math";}
    virtual void f() const=0;
};

```



```

inline void Math::f()const{ cout<<k-3<<endl; }
class Add:public Math
{
public:
    ~Add(){cout<<"Add";}
    void f() const
    {
        cout<<k+3<<endl;
        Math::f();
    }
};
int main()
{
    Math &p=* new Add;
    p.f();
    delete &p;
    return 0;
}

```

```

44. #include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    double x=10.456;
    cout<<showpos<<x<<endl;
    cout<<"10(十六)"<<hex<<(int)x<<endl;
    cout.width(10);cout.fill('#');cout<<x<<endl;
    cout.setf(ios::left);cout<<x<<endl;
    cout.setf(ios::scientific);cout<<x<<endl;
    cout.put('a'+25);
    return 0;
}

```

```

45. #include <iostream>
using namespace std;
class Complex
{
private:
    double realPart, imagePart;

```

```

public:
    Complex(double real = 0, double image = 0): realPart(real), imagePart(image){ }
    Complex operator- (const Complex &a) const{
        return Complex(realPart - a.realPart, imagePart - a.imagePart); }
    void Show() const{ cout << realPart << "+" << imagePart << "i" << endl; }
};
int main()
{
    Complex a(2, 9), b(1, 1), c;
    c = a - b;
    cout << "a="; a.Show();
    cout << "b="; b.Show();
    cout << "c="; c.Show();
    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. 已定义类 Figure 和 main 函数，main 函数中调用派生类函数 showarea()的不同实现版本，求得直角三角形和矩形的面积。请编写此程序的派生类 Triangle 和 Square。

```

class Figure
{
protected :
    double x,y;
public:
    void set(double i, double j=0) { x=i; y=j; }
    virtual void showarea()const = 0 ;
};
int main()
{
    Triangle t;
    Square s;
    t.set(10.48,50);
    t.showarea();
    s.set(888,100);
    s.showarea();
}

```

47. 文件 data.txt 中存放大量整数，输入要读取的整数的个数，打开该文件并在屏幕上显示读取结果，再对这些整数从小到大排序，将排序结果存入 res.txt 文件。