

2023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. C++语言对C语言做了很多改进, 相对于C语言最根本的变化是
  - A. 引进了类和对象的概念
  - B. 增加了一些新的运算符
  - C. 规定函数说明符必须用原型
  - D. 允许函数重载, 并允许设置缺省参数
2. 如想使用命名空间myspace, 正确的程序语句是
  - A. using myspace;
  - B. using namespace;
  - C. namespace myspace;
  - D. using namespace myspace;
3. 下列选项中, 代码复用粒度最大的是
  - A. 类
  - B. 对象
  - C. 函数
  - D. 变量
4. 下列关于类的叙述, 不正确的是
  - A. 类中可以定义不是任何类的成员的函数
  - B. 类的成员变量和成员函数不能定义在类外
  - C. 类中的成员按功能划分, 包括成员变量和成员函数
  - D. 类中的成员按访问权限划分, 包括公有成员、私有成员和保护成员
5. 析构函数被调用的时间点是
  - A. 对象被创建时
  - B. 对象被复制时
  - C. 对象被销毁时
  - D. 对象被引用时
6. 使用new创建的对象, 对其生命周期描述正确的是
  - A. 具有固定生存期
  - B. 具有动态生存期
  - C. 无引用时由系统自动回收
  - D. 从定义处开始, 到程序块结束处结束

7. 下列关于运算符重载的叙述, 不正确的是
  - A. 运算符重载不能创建新运算符
  - B. 运算符重载不能重载为全局函数
  - C. 运算符重载不能改变运算符的原有语义
  - D. 运算符重载不能改变运算符操作数的个数
8. 使用浅拷贝可能带来的问题不包括
  - A. 程序访问非法地址
  - B. 变量类型转换错误
  - C. 重复释放同一块空间
  - D. 变量之间的多余关联
9. 下列关于空类的叙述, 不正确的是
  - A. 无成员变量
  - B. 无成员函数
  - C. 可以作为基类
  - D. 不能派生子类
10. 不允许外界直接访问, 但允许派生类的成员直接访问的是
  - A. 基类的公有成员
  - B. 基类的保护成员
  - C. 基类的私有成员
  - D. 派生类中定义的公有成员
11. 在实现多态的正常代码中, 去掉基类的virtual, 会出现
  - A. 无法通过编译
  - B. 可以通过编译, 但无法正常运行
  - C. 无影响, 运行结果不变
  - D. 可以正常运行, 但运行结果可能改变
12. 类B派生自类A, A不使用虚析构函数。执行语句A \*p=new B(); delete p; 结果为
  - A. 只执行B的析构函数
  - B. 只执行A的析构函数
  - C. 接连执行A和B的析构函数
  - D. 接连执行B和A的析构函数
13. 实现多态时, 必须满足的条件是
  - A. 创建基类对象
  - B. 使用基类引用调用其声明的函数
  - C. 声明派生类指针或引用
  - D. 使用基类指针调用其声明的虚函数
14. 下列关于抽象类的描述中, 不正确的是
  - A. 抽象类中至少应该有一个纯虚函数
  - B. 抽象类的派生类必定是具体类
  - C. 抽象类通常用作类族中最顶层的类
  - D. 抽象类可以定义对象指针和引用
15. iostream是C++标准类库中的一个关键的“流类”, 从该类派生的“流类”为
  - A. ifstream
  - B. fsream
  - C. ofstream
  - D. ostream
16. 下列关于cin的叙述, 不正确的是
  - A. cin是一个预定义类
  - B. cin可以从键盘输入数据
  - C. cin可以从文件输入数据
  - D. cin与标准输入设备相关联
17. 将1到100输出为“001、002、……、100”, 需要使用流操纵符
  - A. setw和left
  - B. setfill和fixed
  - C. setw和right
  - D. setfill和showpos

18. 已有“ifstream input;”，能将input流对象的读指针移到距当前位置后（文件尾方向）100个字节处的语句是
- A. input.seekg(100,ios::beg)                      B. input.seekg(-100,ios::cur)  
C. input.seekg(100,ios::cur)                      D. input.seekg(-100,ios::end)
19. 下列关于文件读写操作的叙述，正确的是
- A. 每个文件均有一个文件结束符  
B. C++将文件看成随机排列的无结构的字节流  
C. 使用文件流打开对象后，文件只能成为一个输入流  
D. 文本文件中通常将一个记录放在一列中，用换行符分隔
20. 假设有函数模板声明如下，则选项中不正确的是
- ```
template <typename T>
T add(T a, T b, T c) { return a+b+c; }
```
- A. add('a', 'b', 'c');                      B. add(10,100,1000);  
C. add("a", "b", "c");                      D. add(0.1, 0.01, 0.001);

## 第二部分 非选择题

二、填空题：本大题共 15 小题，每小题 1 分，共 15 分。

21. 内联函数的函数体是在\_\_\_\_\_时被替换，一般函数的函数体是在调用时被转去执行。
22. 在 C++标识符的作用域中，最大的是\_\_\_\_\_作用域。
23. 如果一个类没有任何构造函数，则由系统自动添加一个\_\_\_\_\_。
24. 友元机制虽然破坏了类的封装性，但是有助于\_\_\_\_\_。
25. A::A(int a, int \*b) { this->x = a; this->y = b; }中，this 的类型是\_\_\_\_\_。
26. 将运算符>>重载为类 CS 的友元函数的格式是：friend istream & operator >>\_\_\_\_\_。
27. 控制基类成员在派生类中的访问属性，一般都使用继承方式说明符\_\_\_\_\_。
28. 派生类对象的初始化工作应该由基类和派生类的\_\_\_\_\_共同实现。
29. 除继承关系之外，类和类之间还可以是\_\_\_\_\_关系。
30. 在面向对象设计思想中，多态可以理解为“一种接口，\_\_\_\_\_”。
31. 包含纯虚函数的类被称为抽象类，它不能\_\_\_\_\_。
32. 流类中提供的 getline()函数可以从输入流中读取一行数据，行结束符是\_\_\_\_\_。
33. 在非标准文件操作中，应该先定义流对象，再打开文件，必须使用\_\_\_\_\_函数。
34. 函数模板实例化过程由编译器完成，生成\_\_\_\_\_。
35. 如果类模板的成员函数定义在类体中，则其自动成为\_\_\_\_\_。

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. 数据文件 a.txt 中保存了空格分隔的 8 个整数，计算它们的最大值。

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int a[8];
    int i = 0;
    cout << "从文件中读取 8 个整数....." << endl;
    _____ (1) _____; //将标准输入重定向到文件 a.txt
    for(i=0; i<8; i++)
        _____ (2) _____;
    int max=a[0];
    int index=0;
    for(i=0; i<8; i++)
    {
        if(max<a[i])
        {
            max = a[i];
            index = i;
        }
    }
    cout << "最大值为: " << max << " 序号为: " << index << endl;
}
```

37. 在程序的下划线处填上正确的语句，使其输出结果为：x=2,y=3

```
#include <iostream>
using namespace std;
class Sample
{
private:
    int x,y;
public:
    Sample (int a,int b);
    void display() { cout<<"x="<<x<<" , y="<<y<<endl; }
};
```

```

    (1) (int a,int b)
    {
        x=a;
        y=b;
    }
int main()
{
    Sample (2);
    Sample *p=&s;
    p->display();
    return 0;
}

```

38. 友元函数访问对象, 输出结果: 2023/4/1  
2023/10/1

```

#include <iostream>
using namespace std;
class Date
{
public:
    Date(int y,int m,int d)
    {
        year=y;
        month=m;
        day=d;
    }
    (1) (Date &);
private:
    int year,month,day;
};
void Print(Date &d)
{
    cout<<d.year<<"/"<<d.month<<"/"<<d.day<<endl;
}
int main()
{
    Date d1(2023,4,1),(2);
    Print(d1);
    Print(d2);
    return 0;
}

```

39. 为类 Position 重载运算符 "+", 从而实现 Position 对象的加法操作。

```

class Position
{
    int x,y;
public:
    Position(int px, int py) { x=px; y=py; }
    (1) (const Position &p)
    {
        return (2);
    }
};

```

40. D1定义一个派生类D2, 在D2中重写函数pvf(), 创建一个D2类的对象调用函数f()、vf()、pvf()。

```

#include <iostream>
using namespace std;
class B1{
public:
    virtual void vf() { cout<<"B1::vf()"<<endl; }
    void f() { cout<<"B1::f()"<<endl; }
    virtual void pvf()=0;
};
class D1:public B1{
public:
    virtual void vf() { cout<<"D1::vf()"<<endl; }
    void f() { cout<<"D1::f()"<<endl; }
};
class (1) {
public:
    virtual void pvf() { cout<<"D2::pvf()"<<endl; }
};
int main()
{
    D2 d2;
    d2.f();
    (2);
    d2.pvf();
    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 Count
{
public:
    static int num;
    static int stat( int a)
    {
        return num=a+num;
    }
};
int Count::num=2;
int main()
{
    for (int i=0;i<3;i++)
        cout<<"第"<<i+1<<"次"<<Count::stat(5)<<endl;
    return 0;
}

42. #include <iostream>
using namespace std;
int global = 1;
class Demo
{
private:
    int i;
public:
    Demo()
    {
        i=global++;
        cout <<i<<"constructor"<<endl;
    }
    ~Demo()
    {
        cout <<i<<"destructor"<<endl;
    }
    int getValue()
    {
        return i;
    }
};
```

```
int main()
{
    Demo *pd=new Demo[2];
    cout<<"pd[0]="<<pd[0].getValue ()<<","pd[1]="<<pd[1].getValue ()<<endl;
    delete[] pd;
    return 0;
}

43. #include<iostream>
using namespace std;
class TriCoor{
public:
    TriCoor(int mx=0, int my=0, int mz=0) { x=mx ; y=my ; z=mz ; }
    void show() { cout<<x<<","<<y<<","<<z<<"\n"; }
    TriCoor operator -- ( )
    {
        x--; y--; z--;
        return *this ;
    }
    TriCoor operator = ( TriCoor t)
    {
        x=t.x ; y=t.y ; z=t.z ;
        return * this ;
    }
    void assign( int mx, int my, int mz ) { x=mx; y=my; z=mz; }
private:
    int x, y, z ;
};

int main()
{
    TriCoor a( 9,8,7 ), b, c ;
    a.show(); b.show(); c.show();
    for( int i = 0; i < 5; i ++ )
        -- a;
    a.show();
    b.assign(3,3,3);
    b.show();
    c = b = a ;
    c.show();
}
```

```

44. #include <iostream>
using namespace std;
class Person{
public:
    void study() { cout << "Person::study" << endl; }
    virtual void rest() { cout << "Person::rest" << endl; }
};
class Student : public Person
{
public:
    virtual void study() { cout << "Student::study" << endl; }
    void rest() { cout << "Student::rest" << endl; }
};
int main()
{
    Student s;
    Person *pp = &s;
    Student *ps = &s;
    pp->study();
    pp->rest();
    ps->study();
    ps->rest();
    return 0;
}

```

```

45. #include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    int a=234;
    cout.setf(ios::dec,ios::basefield);
    cout<<a<<endl;
    cout<<setfill('*')<<setw(8)<<a<<"ok"<<endl;
    double b=1.234567;
    cout<<b<<endl;
    cout<<setw(8)<<setprecision(3)<<b<<endl;
    char c='a';
    cout<<"cout.put(c):"<<endl;
    cout.put(c+2);
}

```

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. 读入当前文件夹下文件 dev1.cpp，为该文件的每行代码加上行号后存入 dev2.txt 文件中。

47. 参考 Point (点) 类和主函数的信息，逐步完成派生类 Circle (圆) 的设计，要求：

- (1) 增加数据成员 r (半径)；
- (2) 设计 Circle 类中构造、析构、输出、设置数据成员和获取数据成员的函数；
- (3) 设计计算圆面积的函数。

```

#include<iostream>
using namespace std;
const double PI=3.1415926;
class Point
{
protected:
    double x,y;//数据成员 x、y (坐标点)
public:
    Point(double xx=0,double yy=0):x(xx),y(yy){}
    ~Point(){cout<<"执行 Point 析构函数"<<endl;}
    void Input()
    {
        cout<<"Input x:"; cin>>x;
        cout<<"Input y:"; cin>>y;
    }
    void Show();
    void SetPoint(double ,double );
    double GetX(){ return x;}
    double GetY(){ return y;}
};
void Point::Show() { cout<<"(x,y)="<<x<<","<<y<<"<<endl; }
void Point::SetPoint(double xx,double yy){ x=xx;y=yy; }
.....
int main()
{
    Point p1;
    cout<<"请输入点的信息: "<<endl;
    p1.Input(); p1.Show();
    Circle c1;
    cout<<"请输入圆的信息: "<<endl;
    c1.Input(); c1.Show();
    cout<<"c1 的面积是: "<<c1.Area()<<endl;
    return 0;
}

```