

## 卷一

### 【单项选择题】

1. 在一个 `cpp` 文里面，定义了一个 `static` 类型的静态全局变量，下面说法正确的是
- A. 只能在该 `cpp` 中使用该变量
  - B. 该变量的值不可修改
  - C. 该变量不能出现在类成员函数中
  - D. 这种变量只能是基本类型，如 `char`，`int`，不能是类类型或者 `struct` 类型

【答案】A

### 【难度】

【解析】Static 全局变量和普通全局变量：针对：一个工程里有多个 `cpp` 文件时

相同点：存储方式相同，都是静态存储区；

不同点：作用域不同。普通全局变量——作用域是整个源程序（含有多个源文件），在各个源文件中都有效

Static 全局变量——作用域是当前源文件

### 【分数】

### 【不定项选择】

3. 以下哪一些是 `C++` 的属性(非 `C` 的属性)
- A. `union`
  - B. `class`
  - C. `template`
  - D. `virtual fuction`
  - E. `pragma`

【答案】BCD

### 【难度】

【解析】#Pragma Para 用法

<https://www.cnblogs.com/eleeye/archive/2011/11/22/2258517.html>

`union` 联合体

### 【分数】

### 【单项选择题】

4. 在基类中，关于访问标号以下描述错误的是
- A. `protected` 成员不可以被类调用访问

- B. private 成员不可以被类调用和友元访问
- C. public 成员可以被派生类访问
- D. protected 成员可以被类调用和友元访问

【答案】B

【难度】

【解析】友元：可以访问所有的成员

【分数】

5. struct T {}, sizeof(struct T)的值为

- A. 4
- B. 1
- C. 0
- D. 编译 or 运行错误

【答案】B

【难度】

【解析】

【分数】

【不定项选择】

6. 关于 C++的说法错误的是
- A. 一个类可以有多个析构函数
  - B. 析构函数可以为 virtual
  - C. 编译器不会为空类产生一个默认的构造函数
  - D. 可以用 free 释放一个对象

【答案】ACD

【难度】

【解析】free 与 delete 区别，free 去释放内存，delete 是调用析构函数，对象的销毁必须要调用析构函数，free 不会调用析构函数

【分数】

【多项选择题】

7. 一下说法不正确的是
- A. 类成员的定义必须放在类的内部

- B. 类的一个成员只能有一种访问控制属性
- C. 在类中, 不做特别声明的数据成员均为私有类型
- D. 构造函数和析构函数是特殊的成员函数, 所以不允许重载

【答案】AD

【难度】

【解析】

【分数】

8. 下面的陈述中哪个是正确的?

- A. 类必须提供至少一个构造函数
- B. 默认构造函数的形参列表中没有形参。
- C. 如果一个类没有有意义的默认值, 则该类不应该提供默认构造函数。
- D. 如果一个类没有定义默认构造函数, 则编译器会自动生成一个, 同时将每个数据成员初始化为相关类型的默认值。

【答案】A

【难度】

【解析】提供的构造函数包括拷贝构造函数, 拷贝构造函数是由参数的

【分数】

【多项选择题】

9. 下列关于 C++类的说法中错误的有哪些?

- A. 一个空类默认会生成构造函数, 拷贝构造函数, 赋值操作符, 析构函数
- B. 一个类可以有多个析构函数
- C. 类中析构函数可以为 virtual, 可以被重载
- D. 类的构造函数如果都不是 public 访问属性, 则类的实例无法创建

【答案】BCD

【难度】

【解析】一个类中只有一个析构函数, 必能被重载, d 选项不对, 单例模式

【分数】

10. 有哪几种情况, 成员变量只能用初始化列表

- A. 当类中含有 const 成员变量
- B. 当类中含有 reference 成员变量

- C. 当类中含有静态成员变量
- D. 当类中含有函数指针成员变量

【答案】AB

【难度】

【解析】静态成员变量赋值在类外

【分数】

11. 在 c++ 中, 关于 struct 和 class, 下面的说法不正确的有哪些

- A. struct 的成员默认是 public, class 的成员默认是 private
- B. struct 不能继承, class 可以继承
- C. struct 的成员变量只能是 public
- D. class 可以有参构造函数, 而 struct 不可以

【答案】BCD

【难度】

【解析】

【分数】

12. 关于虚函数的说法, 哪些是正确的

- A. 构造函数中调用该类的虚函数, 动态绑定机制会生效
- B. 静态函数不可以是虚函数
- C. 虚函数可以声明为 inline
- D. 构造函数和析构函数都可以声明为虚函数

【答案】BC

【难度】

【解析】构造函数不能被声明为虚函数

【分数】

【单项选择题】

13. 下面描述正确的是

```
int *p1 = new int[10];
```

```
int *p2 = new int[10]();
```

- A. p1 和 p2 申请的空间里面的值都是随机值
- B. p1 和 p2 申请的空间里的值都已经初始化
- C. p1 申请的空间里的值是随机值, p2 申请的空间里的值已经初始化

D. p1 申请的空间里的值已经初始化，p2 申请的空间里的值是随机值

【答案】C

【难度】

【解析】new 的使用方法

【分数】

14. 下面关于虚函数和函数重载的叙述不正确的是

A. 虚函数不是类的成员函数

B. 虚函数实现了 C++ 的多态性

C. 函数重载允许非成员函数，而虚函数则不行

D. 函数重载的调用根据参数的个数、序列来确定，而虚函数依据对象确定

【答案】A

【难度】

【解析】

【分数】

15. 设已经有 A, B, C, D 4 个类的定义，程序中 A, B, C, D 析构函数调用顺序为？

C c;

void main()

{

    A\*pa=new A();

    B b;

    static D d;

    delete pa;

}

A. A B C D

B. A B D C

C. A C D B

D. A C B D

【答案】B

【难度】

【解析】构造顺序，c->a->b->d，对象c和对象d存储空间在静态存储区，pa是堆存储区，b对象是栈区，根据代码所示，delete pa，调用a析构函数，执行到}这时，调用b的析构函数，c和d析构顺序与构造顺序相反，所有析构函数调用顺序a,b,d,c

【分数】

16. 写出下面程序的输出结果

```
class A
{
public:
    void FuncA()
    {
        printf( "FuncA called\n" );
    }
    virtual void FuncB()
    {
        printf( "FuncB called\n" );
    }
};

class B : public A
{
public:
    void FuncA()
    {
        A::FuncA();
        printf( "FuncAB called\n" );
    }
    virtual void FuncB()
    {
        printf( "FuncBB called\n" );
    }
};

void main( void )
{
```

```

    B b;
    A *pa;
    pa = &b;
    A *pa2 = new A;
    pa->FuncA(); ( 3)
    pa->FuncB(); ( 4)
    pa2->FuncA(); ( 5)
    pa2->FuncB();
    delete pa2;
}

```

A. FuncA called

FuncB called

FuncA called

FuncB called

B. FuncA called

FuncBB called

FuncA called

FuncB called

C. FuncA called

FuncBB called

FuncAB called

FuncBB called

D. FuncAB called

FuncBB called

FuncA called

FuncB called

**【答案】** B

**【难度】**

**【解析】** 动态多态 virtual 用法

**【分数】**

17. #include<iostream>

using namespace std;

```

class MyClass
{
public:
    MyClass(int i = 0)
    {
        cout << i;
    }
    MyClass(const MyClass &x)
    {
        cout << 2;
    }
    MyClass &operator=(const MyClass &x)
    {
        cout << 3;
        return *this;
    }
    ~MyClass()
    {
        cout << 4;
    }
};

int main()
{
    MyClass obj1(1), obj2(2);
    MyClass obj3 = obj1;
    return 0;
}

```

运行时的输出结果是（）

- A. 11214444
- B. 11314444
- C. 122444
- D. 123444



【答案】C

【难度】

【解析】

【分数】

18. 阅读 C++语言代码输出 ( )

```
int main()
{
    int arr[]={1, 2, 3, 4, 5, 6, 7};
    int *p=arr;
    *(p++)+=89;
    printf("%d, %d\n", *p, *(++p));
    return 0;
}
```

A. 3 3

B. 2 2

C. 2 3

D. 3 2

【答案】C

【难度】

【解析】

【分数】

19. 阅读 c++代码输出 ( )

```
class base1{
    private: int a,b;
    public:
    base1 ( int i ) : b(i+1), a(b) {}
    base1():b(0), a(b) {}
    int get_a() {return a;}
    int get_b() {return b;}
};
```

```

int main()
{
    base1 obj1(11);
    cout<<obj1.get_a()<<endl<<obj1.get_b()<<endl;
    return 0;
}

```

- A. 12 12
- B. 随机数 12
- C. 随机数 随机数
- D. 12 随机数

【答案】B

【难度】

【解析】初始化列表初始化顺序

【分数】

20. 64 位电脑 运行 c++结果输出 ( )

```

class A
{
    char a[2];
public:
    virtual void aa() {};
};

class B:public virtual A
{
    char b[2];
    char a[2];
public:
    virtual void bb() {};
    virtual void aa() {};
};

class C:public virtual B
{

```

```

    char a[2];
    char b[2];
    char c[2];
public:
    virtual void cc() {};
    virtual void aa() {};
    virtual void bb() {};
};
int main()
{
    cout<<sizeof(A)<<endl<<sizeof(B)<<endl<<sizeof(C);
    return 0;
}

```

- A. 8 16 24
- B. 16 32 36
- C. 16 32 48
- D. 8 20 24

【答案】C

【难度】

【解析】

【分数】

## 卷二

1. 运行下面的 c++代码，关于 ptrX ptrY ptrZ 三个指针下面说法正确的是

```

class BaseX
{
public:
    virtual ~BaseX() {};
    virtual void FunctionX() {};
};
class ClassY: public BaseX
{
public:

```

```

        virtual void FunctionY() {};
};

class FatherZ :public ClassY
{
    public:
};

int main()
{
    FatherZ aObject;
    BaseX* ptrX=&aObject;
    ClassY* ptrY=&aObject;
    FatherZ* ptrZ=&aObject;
    return 0;
}

```

- A. ptrX, ptrY, ptrZ 的取值相同.
- B. ptrZ=ptrX+ptrY
- C. ptrX 和 ptrY 不相同
- D. ptrZ 不等于 ptrX 也不等于 ptrY

**【答案】** A

**【难度】**

**【解析】**

**【分数】**

2. 有如下 C++代码:

```

#include <iostream>

using namespace std;

void fun(int *p1, int *p2, int *t)
{
    t = (int *)malloc(sizeof(int));
    *t = *p1 + *(p2++);
}

int main()
{

```

```

    int a[2] = { 1,2 };
    int b[2] = { 10,20 };

    int *t = a;

    fun(a, b, t);

    cout << *t << endl;

}

```

输出是多少?

- A. 11
- B. 10
- C. 1
- D. 2
- E. 12

【答案】C

【难度】

【解析】

【分数】

【单项选择题】

3、下述有关虚函数和纯虚函数说法错误的是?

- A. 被 virtual 关键字修饰的成员函数，就是虚函数
- B. 在基类中实现纯虚函数的方法是在函数原型后加=0; virtual void funtion1()=0
- C. 同时含有纯虚拟函数的类称为抽象类，它可以被实例化，但是对象不可以调用纯虚函数
- D. 使用纯虚函数的意义是在很多情况下，基类本身生成对象是不合情理的

【答案】C

【难度】

【解析】抽象类不能被实例化,所谓不能实例化就是不能声明对象

【分数】

4. 以下输出正确的是 ( )

```

class Base
{
    public:
    void virtual Func()
    {

```

```

        cout<<"Base"<<endl;
    }
};

class Derived : public Base
{
    public:
    void virtual Func()
    {
        cout<<"Derived " <<endl;
    }
};

int main ()
{
    Base* pBase = new Base();
    pBase ->Func();
    Derived * pDerived = (Derived*)pBase;
    pDerived->Func();
    delete pBase;

    pDerived = new Derived();
    pBase = pDerived;
    pBase->Func();

    delete pDerived
    return 0;
}

```

- A. Base Derived Base
- B. Base Base Derived
- C. Base Derived Derived
- D. Derived Base Derived

**【答案】** B

**【难度】**

**【解析】** 继承和多态

**【分数】**

5. 以下 C++ 代码在 64 位环境下会输出多少？

```
#include <iostream>

using namespace std;

struct Bar {
    char c[16];
};

struct Test {
    Bar& bar;
};

int main()
{
    cout << sizeof(Test::bar);
}
```

A. 1

B. 0

C. 16

D. 8

E. 4

**【答案】** C

**【难度】**

**【解析】**

**【分数】**

6. 下面代码的输出结果是什么（ ）

```
#include <iostream>

using namespace std;

class A
{
```

```

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

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

int main() {
    A *pa = new A();
    pa->f();
    B *pb=(B *)pa;
    pb->f();
    delete pa, pb;
    pa=new B();
    pa->f();
    pb=(B *)pa;
    pb->f();
    return 0;
}

```

- A. A A B A
- B. A B B A
- C. A A B B
- D. A A A B

**【答案】** C

**【难度】**

**【解析】** 多态

**【分数】**



7. 下面关于运算符重载说法正确的是?

- A. 当友元函数重载一个运算符时，其参数表中没有任何参数则说明该运算符是一元运算符
- B. 当成员函数重载一个运算符时，其参数表中有 1 个参数则说明该运算符是一元运算符
- C. 类 A 的成员函数重载一个+运算符时，A 的对象+1 与 1+A 的对象并没有区别
- D. 单目运算符最好重载为类的成员函数；双目运算符则最好重载为类的友元函数

【答案】D

【难度】

【解析】重载和友元函数的理解

【分数】

6. 以下选项中可能发生隐式类型转换的是（）

- A. long 到 int
- B. int 到 float
- C. int 到 char
- D. unsigned int 到 int

【答案】B

【难度】

【解析】小类型->大类型会发生隐式转换

【分数】

8. 以下函数调用会选择哪个重载函数（） /跟编译器相关/（待定）

func(1, 2);

- A. int func(int a)
- B. int func(int a, int b);
- C. int func(int a, float b);
- D. int func(int a, char \*b);

【答案】B

【难度】

【解析】函数的重载

【分数】

9. 请选择下列程序的执行结果 ( ) // 类的继承

```
#include<iostream>

using namespace std;

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

class B: public A{
    public:
    void f(){
        cout<<"B::f()";
    };
};

void main(){
    B b;
    b.f();
}
```

- A. A::f(); B::f() 都不可能
- B. A::f(); B::f() 都有可能
- C. A::f()
- D. B::f()

**【答案】**D

**【难度】**

**【解析】**同名覆盖

**【分数】**

10. 下面对于友元函数描述正确的是

- A. 友元函数的实现必须在类的内部定义
- B. 友元函数是类的成员函数
- C. 友元函数破坏了类的封装性和隐藏性
- D. 友元函数不能访问类的私有成员

【答案】C

【难度】

【解析】

【分数】

11. 下面关于 `inline` 说法正确的是

- A. 关键字 inline 是强制行为
- B. 关键字 inline 只是一种编译器建议
- C. inline 不能避免函数调用时的参数传递带来的栈开销
- D. inline 在运行时生效

【答案】B

【难度】

【解析】

【分数】

## 卷五

1. 关于重载和多态正确的是

- A. 如果父类和子类都有相同的方法, 参数个数不同, 将子类对象赋给父类后, 由于子类继承于父类, 所以使用父类指针调用父类方法时, 实际调用的是子类的方法
- B. 选项全部都不正确
- C. 重载和多态在 C++ 面向对象编程中经常用到的方法, 都只在实现子类的方法时才会使用
- D. 

```
class A{
```

```
    void test(float a) {cout<<"1";}
};
```

```
class B:public A{
    void test(int b) {cout<<"2";}
};
```

```
int main() {
    A *a=new A;
    B *b=new B;
    a=b;
    a->test(1.1);
    return 0;
```

```
}
```

结果是 1

【答案】D

【难度】

【解析】A 调用的是父类的同名方法，c 选项，很绝对，随便就能举出反例

【分数】

12. `int a=0;`

```
class someClass{
    int b;
    static int c;
};

int main(){
    int d=0;
    someClass *p=new someClass();
    return 0;
}
```

关于以上代码中的变量在内存中的存储位置描述正确的是()

- A. 堆、栈、全局区均有变量
- B. 不存在堆区的变量
- C. 不存在栈区的变量
- D. 不存在全局变量区的变量

【答案】A

【难度】

【解析】不同变量的存储区，const，全局，static 在全局存储区

【分数】

13. `#include<stdio.h>`

```
int main()
{
    unsigned char i=7;
    int j=0;
```

```
for(;i>0;i-=3)
{
    ++j;
}
printf("%d\n",j);
return 0;
}
```

请问该程序的输出是多少？

- A. 2
- B. 死循环
- C. 173
- D. 172

【答案】C

【难度】

【解析】应为是 unsigned char 所以没有负数，-1 对应 255，只有等于 0 时候跳出 for 循环

【分数】

14. 在 c++中的结构体是否可以有构造函数？

- A. 不可以，结构类型不支持成员函数
- B. 可以有
- C. 不可以，只有类允许有构造函数

【答案】B

【难度】

【解析】struct 在 c++中 和 class 一样，只是默认访问权限不同

【分数】

15. 下列哪项不属于 C++的基本数据类型？

- A. wide
- B. char
- C. bool

【答案】A

【难度】

【解析】

**【分数】**

16. T 是一个数据类型，关于 `std::vector::at` 和 `std::vector::operator[]` 描述正确的是：

- A. `at` 总是做边界检查，`operator[]` 不做边界检查.
- B. `at` 不做边界检查，`operator[]` 做边界检查.
- C. `at` 和 `operator[]` 是一样的

**【答案】** A

**【难度】**

**【解析】** 基本知识点

**【分数】**

17. 使用操作符 `setw()` 对数据进行格式输出时，应包含（ ）文件。

- A. `fstream.h`
- B. `stdlib.h`
- C. `iostream.h`
- D. `iomanip.h`

**【答案】** D

**【难度】**

**【解析】** 基本知识点

**【分数】**

18. 下列字符串可以用作 C++标识符的是：

- A. `_123`
- B. `2009var`
- C. `goto`
- D. `test - 2009`

**【答案】** A

**【难度】**

**【解析】** 命名规则

**【分数】**

19. 以下说法中正确的是（ ）。

- A. C++程序中的 main() 函数必须放在程序的开始部分
- B. C++程序的入口函数是 main 函数
- C. 在 C++程序中，要调用的函数必须在 main() 函数中

【答案】B

【难度】

【解析】

【分数】

20. STL 中的一级容器有：

- A. vector, deque, list, set, multiset, map, multimap.
- B. 序列容器，关联容器，容器适配器
- C. set, multiset, map, multimap.
- D. vector, deque, list.

【答案】D

【难度】

【解析】

【分数】

### 卷三

1. 如果有一个类是 myClass ， 关于下面代码正确描述的是：

```
myClass::~~myClass() {  
    delete this;  
    this = NULL;  
}
```

- A. 正确，我们避免了内存泄漏
- B. 它会导致栈溢出
- C. 无法编译通过
- D. 这是不正确的，它没有释放任何成员变量。

【答案】C

【难度】

【解析】this = null 报错，Myclass \*const this, const 修饰的是 this，表明指针为常量，指针指向的内容可以修改，但指针不能修改，this=NULL 是把指针修改

【分数】

2. 下面的程序输出可能是什么？

```
class Printer{
public:
    Printer(std::string name) {std::cout << name;}
};

class Container{
public:
    Container() : b("b"), a("a") {}

    Printer a;
    Printer b;
};

int main() {
    Container c;
    return 0;
}
```

- A. 可能是 "ab" 或 "ba"。 依赖于具体的实现
- B. 一直都是 "ba"
- C. 一直都是 "ab"

【答案】C

【难度】

【解析】初始化列表的执行顺序

【分数】

3. 在重载一个运算符为成员函数时，其参数表中没有任何参数，这说明该运算符是（ ）。

- A. 无操作数的运算符
- B. 二元运算符
- C. 前缀一元运算符



D. 后缀一元运算符

【答案】C

【难度】

【解析】

【分数】

4. 已知 fun(int)是类 Test 的公有成员函数，p 是指向成员函数 fun() 的指针，采用 ( ) 是正确的。

A. p=fun

B. p=Test::fun()

C. p=fun()

D. p=&Test::fun

【答案】D

【难度】

【解析】

【分数】

5. 有如下程序：

```
#include<iostream>
using namespace std;
class MyClass{
    public:
        MyClass(int i=0) {cout<<1;}
        MyClass(const MyClass&x) {cout<<2;}
        MyClass& operator=(const MyClass&x) {cout<<3; return*this;}
        ~MyClass() {cout<<4;}
};
int main() {
    MyClass obj1(1), obj2(2), obj3(obj1);
    return 0;
}
```

运行时的输出结果是

- A. 121, 444
- B. 112, 444
- C. 11, 114, 444
- D. 11, 314, 445
- E. 11, 314, 444

**【答案】** B

**【难度】**

**【解析】** 构造函数和析构函数的调用

**【分数】**

6. 关于以下代码，哪个说法是正确的？

```
myClass::foo() {  
    delete this;  
}  
..  
void func() {  
    myClass *a = new myClass();  
    a->foo();  
}
```

- A. 它会引起栈溢出
- B. 都不正确
- C. 它不能编译
- D. 它会引起段错误

**【答案】** B

**【难度】**

**【解析】** DELETE THIS 在不同函数中调用的位置不同，会产生不同结果 delete this 在析构函数中调用，会导致循环调用，导致栈溢出，在其他成员函数中调用，没有问题

**【分数】**

7. 下列程序的输出结果是

```
#include<iostream.h>  
  
void f() {
```

```

        static int i=15;

        i++;

        cout<<"i="<<i<<endl;
    }

    void main() {
        for(int k=0;k<2;k++)
            f();
    }

```

- A. i=16      i=15
- B. i=15      i=17
- C. i=16      i=17
- D. i=15      i=16
- E. i=15      i=18

【答案】C

【难度】

【解析】STATIC 修饰变量，只初始化一次，然后跳过初始化

【分数】

8. 关于子类型的描述中，（ ）是错误的？

- A. 在公有继承下, 派生类是基类的子类型
- B. 子类型关系是不可逆的
- C. 子类型就是指派生类是基类的子类型
- D. 一种类型当它至少提供了另一种类型的行为, 则这种类型是另一种类型的子类型

【答案】C

【难度】

【解析】子类型必须是子类继承了父类的所有可继承特性，也即公有继承，才能说是子类型，否则就只是单纯的子类

【分数】

9. 假设 A 为抽象类，下列声明（ ）是正确的。

- A. int fun(A);
- B. A Obj;

C. A fun(int);

D. A \*p;

【答案】D

【难度】

【解析】抽象类不能初始化，不能当做返回值，不能当做参数，可以作为指针变量，因为此时还没有初始化

【分数】

10. 下列关于多态性说法不正确的是（ ）。

A. 多态性是指同名函数对应多种不同的实现

B. 重载方式仅有函数重载

C. 重载方式包含函数重载和运算符重载

D. 多态性表现为静态和动态两种方式

【答案】B

【难度】

【解析】

【分数】

11. 如果类B继承类A，A::x()被声明为虚函数，B::x()重载了A::x()方法，在下述语句中哪个x()方法会被调用：

B b;

b.x();

A. A::x()

B. B::x()

C. A::x() B::x()

D. B::x() A::x()

【答案】B

【难度】

【解析】多态

【分数】

12. 哪个操作符不能作为类成员函数被重载？

A. ?:

- B. ++
- C. []
- D. ==
- E. \*

【答案】A

【难度】

【解析】不可重载运算符包括 ?:, ::, ., \* 这四个

【分数】

13. 下列对重载函数的描述中，()是错误的。

- A. 重载函数中不允许使用默认参数
- B. 重载函数中编译时根据参数表进行选择
- C. 构造函数重载将会给初始化带来多种方式
- D. 不要使用重载函数来描述毫无相干的函数

【答案】A

【难度】

【解析】

【分数】

14. 下面程序的输出结果是()

```
#include <iostream.h>
#define SQR(A) A*A
void main() {
    int x=6, y=3, z=2;
    x/=SQR(y+z)/SQR(y+z);
    cout<< x<< endl;
}
```

- A. 5
- B. 6
- C. 1
- D. 0

【答案】D

【难度】

【解析】宏的简单替换

【分数】

15. C++是否支持嵌套函数定义，如下代码所示？

```
fun1() {  
    fun2() {  
        fun3() {  
        }  
    }  
}
```

- A. 允许
- B. 不允许

【答案】B

【难度】

【解析】支持嵌套使用，不支持嵌套定义

【分数】

16. 在 C++语言中，下列说法正确的是：

- A. inline 函数表示编译器一定会将函数体直接插入到调用此函数的地方，这样可以加快程序的运行速度
- B. 局部变量和全局变量不能重名
- C. 静态局部变量内存是在运行期分配的，存储在堆栈区
- D. C++不是类型安全的语言

【答案】D

【难度】

【解析】首先，可以排除 B 和 C。B 选项，因为局部变量和全局变量是可以重名的，而且一般来说在某个函数体内，局部变量是会覆盖全局变量的。C 选项，静态局部变量存在静态存储区，而局部变量存储在堆栈区，确切的说是栈区。再看 A 选项，内联函数与其他普通函数的区别是，内联函数在生成可执行文件时，其代码块是直接嵌入到调用处的，以此减少函数调用的开销提高程序性能，它与宏很类似。但是，C++ primer 上明明白白的写过这么一句话：内联说明（inline specification）对于编译器来说只是一个建议，编译器可以选择忽略这个建议。也就是说，是否以嵌入式方式存在，是由编译器决定的，并不是一

定。最后看 D 选项，什么叫类型安全的语言？本人很浅陋的认识是：就 C++ 而言，我们可以把 0 作为 false，非零作为 true。一个函数就算是 bool 类型的，我们还是可以返回 int 类型，并且自动把 0 转化成 false，非零转化成 true。相比之下 java 不能把 int 类型转化成 bool 类型。所以 java 是一种类型安全的语言，而 C++ 并不是。 答案选 D

【分数】

17. C++ 中以下关于函数调用的说法哪个是正确的？

- A. 传地址后实参和形参指向不同的对象
- B. 传引用后实参和形参是不同的对象
- C. 传值后对形参的修改会改变实参的值
- D. 其他三项都不对

【答案】D

【难度】

【解析】传地址或者引用时，形参与实参都有相同的地址；传值调用时，只是将数据进行了复制，不会改变实参的值

【分数】

18. 下面 C++ 程序的输出结果：

```
int i=0, a=1, b=2, c=3;
i=++a || ++b || ++c;
printf("%d %d %d %d", i, a, b, c);
```

- A. 1 2 2 3
- B. 1 2 3 4
- C. 3 2 3 4
- D. 3 3 3 4

【答案】D

【难度】

【解析】整个表达式是逻辑表达式，I 的最终值是逻辑操作，只能是 0 or 1，逻辑或一旦为真，后面的都不执行了，所以选 a

【分数】

19. 下面有关 C++ 中为什么用模板类的原因，描述错误的是？

- A. 可用来创建动态增长和减小的数据结构
- B. 它是类型无关的，因此具有很高的可复用性
- C. 它运行时检查数据类型，保证了类型安全
- D. 它是平台无关的，可移植性

【答案】C

【难度】

【解析】（1）可用来创建动态增长和减小的数据结构 （2）它是类型无关的，因此具有很高的可复用性。（3）它在编译时而不是运行时检查数据类型，保证了类型安全 （4）它是平台无关的，可移植性 （5）可用于基本数据类型

【分数】

20. 下面有关继承、多态、组合的描述，说法错误的是？

- A. 封装，把客观事物封装成抽象的类，并且类可以把自己的数据和方法只让可信的类或者对象操作，对不可信的进行信息隐藏
- B. 继承可以使用现有类的所有功能，并在无需重新编写原来的类的情况下对这些功能进行扩展
- C. 隐藏是指派生类中的函数把基类中相同名字的函数屏蔽掉了
- D. 覆盖是指不同的函数使用相同的函数名，但是函数的参数个数或类型不同

【答案】D

【难度】

【解析】重载：

只有在 同一类定义中的同名成员函数才存在重载关系，主要特点是 函数的参数类型和数目有所不同，但 不能出现函数参数的个数和类型均相同，仅仅依靠返回值类型不同来区分的函数，这和普通函数的重载是完全一致的。另外，重载和成员函数是否是虚函数无关

覆盖：

在派生类中覆盖基类中的同名函数，要求两个函数的参数个数、参数类型、返回类型都相同，且基类函数必须是虚函数。

隐藏：

派生类中的函数屏蔽了基类中的同名函数，



2 个函数参数相同，但基类函数不是虚函数（和覆盖的区别在于基类函数是否是虚函数）。2 个函数参数不同，无论基类函数是否是虚函数，基类函数都会被屏蔽（和重载的区别在于两个函数不在同一类中）。

【分数】

#### 卷四

1. 下面对静态数据成员的描述中，正确的是？

- A. 静态数据成员可以在类体内进行初始化
- B. 静态数据成员不可以被类的对象调用
- C. 静态数据成员不受 `private` 控制符的作用
- D. 静态数据成员可以直接用类名调用

【答案】D

【难度】

【解析】`static` 成员属性性质

【分数】

2. 下列一段 C++ 代码的输出是？

```
#include "stdio.h"

class Base
{
public:
    int Bar(char x)
    {
        return (int)(x);
    }
    virtual int Bar(int x)
    {
        return (2 * x);
    }
};

class Derived : public Base
{
```

```

public:
    int Bar(char x)
    {
        return (int)(-x);
    }

    int Bar(int x)
    {
        return (x / 2);
    }
};

int main(void)
{
    Derived Obj;
    Base *pObj = &Obj;
    printf("%d, ", pObj->Bar((char)(100)));
    printf("%d, ", pObj->Bar(100));
}

```

- A. 100, -100
- B. 100, 50
- C. 200, -100
- D. 200, 50

【答案】B

【难度】

【解析】多态

【分数】

【多项选择题】

3. C++类体系中，不能被派生类继承的有？
- A. 构造函数
  - B. 静态成员函数
  - C. 非静态成员函数
  - D. 赋值操作函数

【答案】ABD

【难度】

【解析】继承能继承的范围

【分数】

4. 下面有关 malloc 和 new, 说法错误的是?

- A. new 建立的是一个对象, malloc 分配的是一块内存.
- B. new 初始化对象, 调用对象的构造函数, 对应的 delete 调用相应的析构函数, malloc 仅仅分配内存, free 仅仅回收内存
- C. new 和 malloc 都是保留字, 不需要头文件支持
- D. new 和 malloc 都可用于申请动态内存, new 是一个操作符, malloc 是一个函数

【答案】C

【难度】

【解析】new 和 malloc 区别, new 调用构造函数, 而 malloc 只分配内存

【分数】

5. 拷贝构造函数的特点是 ( )

- A. 该函数名同类名, 也是一种构造函数, 该函数返回自身引用
- B. 该函数只有一个参数, 是对某个对象的引用
- C. 每个类都必须有一个拷贝初始化构造函数, 如果类中没有说明拷贝构造函数, 则编译器系统会自动生成一个缺省拷贝构造函数, 作为该类的保护成员
- D. 拷贝初始化构造函数的作用是将一个已知对象的数据成员值拷贝给正在创建的另一个同类的对象

【答案】D

【难度】

【解析】

【分数】

6. 下列关于虚函数的说法正确的是 ( )

- A. 在构造函数中调用类自己的虚函数, 虚函数的动态绑定机制还会生效
- B. 在析构函数中调用类自己的虚函数, 虚函数的动态绑定机制还会生效
- C. 静态函数不可以是虚函数
- D. 虚函数可以声明为 inline

【答案】CD

【难度】

【解析】

【分数】

7. 下面描述中，错误的是（）

- A. 基类定义的 public 成员在公有继承的派生类中可见，也能在类外被访问
- B. 基类定义的 public 和 protected 成员在私有继承的派生类中可见，在类外可以被访问
- C. 基类定义的 public 和 protected 成员在保护继承的派生类中不可见
- D. 基类定义的 protected 成员在 protected 继承的派生类中可见，也能在类外被访问

【答案】BCD

【难度】

【解析】

【分数】

8. 关于内联函数正确的是（）

- A. 类的私有成员函数不能作为内联函数
- B. 在所有类说明中内部定义的成员函数都是内联函数
- C. 类的保护成员函数不能作为内联函数
- D. 使用内联函数的地方会在运行阶段用内联函数体替换掉

【答案】B

【难度】

【解析】

【分数】

9. 考虑函数原型 `void hello(int a, int b=7, char* pszC="*")`，下面的函数调用中，属于不合法调用的是

- A. `hello(5)`
- B. `hello(5, 8)`
- C. `hello(6, "#")`
- D. `hello(0, 0, "#")`

【答案】C

【难度】

【解析】不解释

【分数】

10. 下面有关重载函数的说法中正确的是

- A. 重载函数必须具有不同的返回值类型
- B. 重载函数形参个数必须不同
- C. 重载函数必须有不同的形参列表
- D. 重载函数名可以不同

【答案】C

【难度】

【解析】1、函数有不同参数，2、函数有不同参数类型，3、同一个作用域

【分数】

11. 分析一下程序的运行结果：

```
class CBase
{
public:
    CBase() {cout<<" constructing CBase class" <<endl;}
    ~CBase() {cout<<" destructing CBase class" <<endl;}
};
```

```
class CSub : public CBase
{
public:
    CSub() {cout<<" constructing CSub class" <<endl;}
    ~CSub() {cout<<" destructing CSub class" <<endl;}
};
```

```
void main()
{
    CSub obj;
}
```

- A. constructing CSub class  
    constructing CBase class  
    destructing CSub class  
    destructing CBase class
- B. constructing CBase class  
    constructing CSub class  
    destructing CBase class  
    destructing CSub class
- C. constructing CBase class  
    constructing CSub class  
    destructing CSub class  
    destructing CBase class
- D. constructing CSub class  
    constructing CBase class  
    destructing CBase class  
    destructing CSub class

【答案】C

【难度】

【解析】继承的构造和析构顺序，构造：先父亲，再朋友，后自己，析构与构造相反

【分数】

13. 有如下一段代码：

```
#define ADD(x,y) x+y
```

```
int m=3;
```

```
m+=m*ADD(m,m);
```

则 m 的值为：

- A. 15
- B. 12
- C. 18
- D. 58

【答案】A

【难度】

【解析】宏定义：简单替换

【分数】

14. 下面的模板声明中, 正确的是:

- A. `template<typename T1, T2>`
- B. `template<class T1, T2>`
- C. `template<class T1, class T2>`
- D. `template<typename T1; typename T2>`

【答案】C

【难度】

【解析】`template<class T>`和 `template<typename T>`都可以用来定义函数模板和类模板, 在使用上, 他们俩没有本质的区别

【分数】

15. 下面关于多态性的描述, 错误的是:

- A. C++语言的多态性分为编译时的多态性和运行时的多态性
- B. 编译时的多态性可通过函数重载实现
- C. 运行时的多态性可通过模板和虚函数实现
- D. 实现运行时多态性的机制称为动态绑定

【答案】C

【难度】

【解析】A, 正确, 分编译时多态和运行时多态; B, 编译时多态可以通过函数重载实现, 具体表现在根据参数的个数和类型不同选择合适的同名函数; C, 运行时多态通过虚函数实现, 就是运行时根据对象类型自动选择正确的调用接口。模板属于编译时多态性, 因为编译时自动根据模板生成模板函数; D, 运行时多态是根据对象类型自动选择正确的调用函数, 也叫动态绑定。

【分数】

16. 下面关于类和对象的描述中, 错误的是:

- A. 类就是 C 语言中的结构体类型, 对象就是 C 语言中的结构体变量
- B. 类和对象之间的关系是抽象和具体的关系
- C. 对象是类的实例, 一个对象必须属于一个已知的类
- D. 类是具有共同行为的若干对象的统一描述体

【答案】A

【难度】

【解析】C 中的 struct 和 c++的 struct 根本不是一个概念

【分数】

17. 设 a、b、c、d、m、n 均为 int 型变量，且 a=5、b=6、c=7、d=8、m=2、n=2, 则逻辑表达式 (m=a>b) &&(n=c>d) 运算后，n 的值为：

A. 0

B. 1

C. 2

D. 7

【答案】C

【难度】

【解析】m=a>b 后 m=0，表达式为假，&&后半部分不会操作，因此 n 为初始值 2

【分数】

18. 建立派生类对象时，3 种构造函数分别是 a(基类的构造函数)、b(成员对象的构造函数)、c(派生类的构造函数)这 3 种构造函数的调用顺序为：

A. abc

B. acb

C. cab

D. cba

【答案】A

【难度】

【解析】继承的构造和析构顺序，构造：先父亲，再朋友，后自己，析构与构造相反

【分数】

19. 下面模板声明中，哪些是非法的 ( )

A. `template<class Type>class C1 {};`

B. `template<class T, U, class V>class C2 {};`

C. `template<class C1, typename C2>class C3 {};`

D. `template<typename myT, class myT>class C4 {};`

【答案】B D



【难度】

【解析】class 和 typename 一样，B 选项 u 没有指定类型，D 选项指定名字 myT 相同

【分数】

20. 拷贝构造函数的特点是 ( )

- A. 该函数名同类名，也是一种构造函数，该函数返回自身引用
- B. 该函数只有一个参数，是对某个对象的引用
- C. 每个类都必须有一个拷贝初始化构造函数，如果类中没有说明拷贝构造函数，则编译器系统会自动生成一个缺省拷贝构造函数，作为该类的保护成员
- D. 拷贝初始化构造函数的作用是将一个已知对象的数据成员值拷贝给正在创建的另一个同类的对象

【答案】D

【难度】

【解析】拷贝函数和构造函数没有返回值，A 错；拷贝构造函数的参数可以使一个或多个，但左起第一个必须是类的引用对象，B 错；若类定义中没有声明拷贝构造函数，则编译器会自动生成一个缺省的拷贝构造函数，但是不会是该类的保护成员，C 错；通过拷贝函数可以将另一个对象作为对象的初值，D 对

【分数】

## 卷五

1. 下列关于一个类的静态成员的描述中，不正确的是

- A. 该类的对象共享其静态成员变量的值
- B. 静态成员变量可被该类的所有方法访问
- C. 该类的静态方法能访问该类的静态成员变量
- D. 该类的静态数据成员变量的值不可修改

【答案】D

【难度】

【解析】类的静态成员与类直接相关，与对象无关，在一个类的所有实例之间共享同一个静态成员，A 正确；静态成员函数中不能调用非静态成员，C 正确；非静态成员函数中可以调用静态成员，B 正确；D 静态成员变量的值可以改变，只初始化一次

【分数】

2. C++将父类的析构函数定义为虚函数，下列正确的是哪个？

- A. 释放父类指针时能正确释放子类对象
- B. 释放子类指针时能正确释放父类对象
- C. 这样做是错误的
- D. 以上全错

【答案】A

【难度】

【解析】C++中假设有基类为 father，它的派生类为 son，如果有 `father* fa = new son()`；在 `delete fa` 或者释放\*fa 的时候将只会调用基类的析构函数；如果基类的析构函数为虚函数，在 `delete fa` 或者释放\*fa 的时候会先调用派生类（这里也就是 son）的析构函数，再调用基类的析构函数。

【分数】

3. 下列 for 循环的循环体执行次数为

`for(int i=10, j=1; i=j=0; i++, j--)`

- A. 0
- B. 1
- C. 无限
- D. 以上都不对

【答案】A

【难度】

【解析】FOR(初始化;逻辑判断;循环次数) `i=j=0` 表达式结果是 0 为假，不进入循环，所以选 A

【分数】

4. 类定义的外部，一定可以被访问的成员有（ ）。

- A. 所有类成员
- B. private 或 protected 的类成员
- C. public 的类成员
- D. public 或 private 的类成员

【答案】C

【难度】

【解析】

**【分数】**

5. 静态变量通常存储在进程哪个区？（）

- A. 栈区
- B. 堆区
- C. 全局区
- D. 代码区

**【答案】**C

**【难度】**

**【解析】**

**【分数】**

6. 下面哪种面向对象的方法可以让你变得富有()

- A. 继承
- B. 封装
- C. 多态
- D. 抽象

**【答案】**A

**【难度】**

**【解析】**继承可以继承父类的属性和行为。封装是只提供接口并隐藏内部过程的行为。多态是指因继承产生的相关不同类，但是因为我没有多人让我继承，我也没有兄弟姐妹（就算有，他们是否富有也与我无关）。抽象是一种简化复杂现实问题的途径。简单来说就是富二代

**【分数】**

7. 假定 CSomething 是一个类，执行下面这些语句之后，内存里创建了\_\_\_\_个 CSomething 对象。

```
CSomething a();  
CSomething b(2);  
CSomething c[3];  
CSomething &ra = b;  
CSomething d=b;
```

```
CSomething *pA = c;
```

```
CSomething *p = new CSomething(4);
```

- A. 10
- B. 9
- C. 8
- D. 7
- E. 6
- F. 5

【答案】E

【难度】

【解析】CSomething a(); // 没有创建对象，这里不是使用默认构造函数，而是定义了一个函数。

CSomething b(2); // 使用一个参数的构造函数，创建了一个对象。

CSomething c[3]; // 使用无参构造函数，创建了3个对象。

CSomething &ra=b; // ra 引用 b，没有创建新对象。

CSomething d=b; // 使用拷贝构造函数，创建了一个新的对象 d。

CSomething \*pA = c; // 创建指针，指向对象 c，没有构造新对象。

CSomething \*p = new CSomething(4); // 新建一个对象。

【分数】

8. 有一个类 A，其数据成员如下：

```
class A {  
    ...  
private:  
    int a;  
public:  
    const int b;  
    float* &c;  
    static const char* d;  
    static double* e;  
};
```

则构造函数中，成员变量一定要通过初始化列表来初始化的是：\_\_\_\_\_。

- A. a b c

- B. b c
- C. b c d e
- D. b c d
- E. b
- F. c

【答案】B

【难度】

【解析】初始化列表，对象属性，const 成员属性，引用成员属性

【分数】

9. 当一个类的某个函数被说明为 virtual，则在该类的所有派生类中的同原型函数\_\_\_\_\_？
- A. 只有 被重新说明时才识虚函数
  - B. 只有被重新说明为 virtual 时才是虚函数
  - C. 都不是虚函数
  - D. 都是虚函数

【答案】D

【难度】

【解析】虚函数的概念

【分数】

10. 关于函数的描述正确的是\_\_\_\_\_。
- A. 虚函数是一个 static 型的函数
  - B. 派生类的虚函数与基类的虚函数具有不同的参数个数和类型
  - C. 虚函数是一个非成员函数
  - D. 基类中说明了虚函数后，派生类中起对应的函数可以不必说明为虚函数

【答案】D

【难度】

【解析】虚函数的概念

【分数】

11. 以下代码输出什么()

```

void main()
{
    int a[5]={1,2,3,4,5};
    int *p=(int *)(&a+1);
    printf("%d",*(p-1));
}

```

- A. 1
- B. 2
- C. 5
- D. 出现错误

【答案】C

【难度】

【解析】a 是数组的首地址，\*p = &a + 1 表示数组指针+1，指像 a[5]，强转 int\*，\*(p - 1) = a[4] = 5

【分数】

12. 有如下 C++代码：

```

struct A{
    void foo() {printf("foo");}
    virtual void bar() {printf("bar");}
    A() {bar();}
};

struct B:A{
    void foo() {printf("b_foo");}
    void bar() {printf("b_bar");}
};

```

那么

```
A *p=new B;
```

```
p->foo();
```

```
p->bar();
```

输出为：

- A. barfoob\_bar

- B. foobarb\_bar
- C. barfoob\_foo
- D. foobarb\_fpp

【答案】A

【难度】

【解析】虚函数，B 继承自 A，A \*p=new B，构造了一个 B 对象，先构造 A 在构造 B，构造 A 时，调用 A 的 bar（）函数，输出 bar，应为 foo 不是虚函数，所以 p->foo()，输出 foo，bar（）是虚函数，调用 B 的 bar（），输出 b\_bar，所以输出是 barfoob\_bar

【分数】

13. 以下代码是否完全正确，执行可能得到的结果是\_\_\_\_\_。

```
class A{
    int i;
};
class B{
    A *p;
public:
    B() {p=new A;}
    ~B() {delete p;}
};
void sayHello(B b) {
}
int main() {
    B b;
    sayHello(b);
}
```

- A. 程序正常运行
- B. 程序编译错误
- C. 程序崩溃
- D. 程序死循环

【答案】C

【难度】

【解析】当执行 sayHello(B b)函数传参时，调用拷贝构造函数，该函数执行结束后，调用 B 的析构函数 delete p 一次，在 main 函数结束后，又调用 B 的析构函数，又 delete 一次 p，删除了两次 p，程序崩溃

【分数】

14. 有这样一个类：

```
class Eye
{
public:
void Look(void);
};
```

现在希望定义一个 Head 类，也想实现 Look 的功能，应该使用（）方法，实现代码重用。

- A. 继承
- B. 组合
- C. 模板
- D. 过滤

【答案】B

【难度】

【解析】

【分数】

15. 以下符号在 C++语法中不能作为运算符的是（）

- A. !
- B. @
- C. ^
- D. |

【答案】B

【难度】

【解析】

【分数】

16. 以下 C 语言和 C++语言比较中错误的是（）



- A. C 语言绝大部分语法适用于 C++
- B. C 语言和 C++的 struct 意义和用法一致
- C. C 语言不能使用继承技术，但 C++可以
- D. C++是面向对象的语言，C 语言是面向过程的语言

【答案】B

【难度】

【解析】

【分数】

17. 关于类成员函数的重载、覆盖和隐藏，下面说法错误的是（）

- A. 重载函数必须要在同一个类中
- B. 派生类函数覆盖基类函数，基类函数最好不要有 virtual 关键字
- C. 如果派生类的函数与基类的函数名相同，通过派生类的实例也只能调用到派生类的函数。
- D. 重载函数不能使用 virtual 关键字修饰

【答案】B

【难度】

【解析】

【分数】

18. 以下输出正确的是（）

```
class Base
{
    public:
    void virtual Func()
    {
        cout<<"Base"<<endl;
    }
};
```

```
class Derived : public Base
{
```

```

    public:
    void virtual Func()
    {
        cout<<"Derived " <<endl;
    }
};

int main ()
{
    Base* pBase = new Base();
    pBase ->Func();
    Derived * pDerived = (Derived*)pBase;
    pDerived->Func();
    delete pBase;

    pDerived = new Derived();
    pBase = pDerived;
    pBase->Func();

    delete pDerived
    return 0;
}

```

- A. Base Derived Base
- B. Base Base Derived
- C. Base Derived Derived
- D. Derived Base Derived

**【答案】** B

**【难度】**

**【解析】**

**【分数】**

18. 一个空类默认产生哪些成员函数

- A. 默认构造函数
- B. 析构函数
- C. 拷贝构造函数
- D. 赋值函数

【答案】ABCD

【难度】

【解析】默认构造函数式无参数的，拷贝构造函数，析构函数，赋值函数就是  
`A&operator=(const A&);`

【分数】

```
19. struct T{
    T(int) {};
    T() {};
    void func() {}
};

int main() {
    Test a(1); // 语句 1
    a.func(); // 语句 2
    Test b(); // 语句 3
    b.func(); // 语句 4
    return 0;
}
```

- A. 语句 1
- B. 语句 2
- C. 语句 3
- D. 语句 4

【答案】D

【难度】

【解析】语句 3 不是调用默认构造函数，而是声明一个函数，但是编译器不会报错，执行到语句 4 的时候，b 不是一个对象，调用不了成员函数，所以语句 4 报错

【分数】

20. 下述有关虚函数和纯虚函数说法错误的是？

- A. 被 virtual 关键字修饰的成员函数，就是虚函数
- B. 在基类中实现纯虚函数的方法是在函数原型后加 virtual void funtion1()=0
- C. 同时含有纯虚拟函数的类称为抽象类，它可以被实例化，但是对象不可以调用纯虚函数
- D. 使用纯虚函数的意义是在很多情况下，基类本身生成对象是不合情理的

【答案】C

【难度】

【解析】抽象类不可以实例化

【分数】

## 卷六

1. 以下 C 语言和 C++语言比较中错误的是（）

- A. C 语言绝大部分语法适用于 C++
- B. C 语言和 C++的 struct 意义和用法一致
- C. C 语言不能使用继承技术，但 C++可以
- D. C++是面向对象的语言，C 语言是面向过程的语言

【答案】B

【难度】

【解析】struct 和 class 在 c++都是声明类，C 中的 struct 只是不同数据类型的组合

【分数】

2. 关于类成员函数的重载、覆盖和隐藏，下面说法错误的是（）

正确答案：B 你的答案：空（错误）

- A. 重载函数必须要在同一个类中
- B. 派生类函数覆盖基类函数，基类函数最好不要有 virtual 关键字
- C. 如果派生类的函数与基类的函数名相同，通过派生类的实例也只能调用到派生类的函数。
- D. 重载函数不能使用 virtual 关键字修饰

【答案】B

【难度】

【解析】B 选项，如果没有 virtual 关键字声明，只能调用基类的同名函数，不能达到多态的效果

【分数】

3. 类中访问修饰符作用范围由大到小是

- A. private-protected-default-public
- B. private-default-protected-public
- C. public-protected-default-private
- D. public-default-protected-private

【答案】C

【难度】

【解析】访问权限

【分数】

4. 下列程序编译时会出现错误，请根据行号选择错误位置( )

```
#include <iostream>
using namespace std;
class A{
    int a1;
protected:
    int a2;
public:
    int a3;
};
class B: public A{
    int b1;
protected:
    int b2;
public:
    int b3;
};
```

```

class C:private B{
    int c1;
protected:
    int c2;
public:
    int c3;
};

int main() {
    B obb;
    C obc;
    cout<<obb.a1;//1
    cout<<obb.a2;//2
    cout<<obb.a3;//3
    cout<<obc.b1;//4
    cout<<obc.b2;//5
    cout<<obc.b3;//6
    cout<<obc.c3;//7
    return 0;
}

```

- A. 1, 2
- B. 2, 5, 7
- C. 3, 4, 7
- D. 4, 5, 6

**【答案】**AD

**【难度】**

**【解析】**1. B 的对象不能访问 A 的私有成员，1 错，public 继承，protected 属性继承过来还是 protected 属性，类外不能直接访问，2 错，public 属性类外可以直接访问，3 对，C 类私有继承自 B 类，父类中的 public 和 protected 在子类中变成 private，外部不能直接访问，所以 456 全错，c3 是 public 为外部可以直接访问 7 对，所以选 AD

**【分数】**

5. 以下对于方法覆盖的说法正确的有（）

- A. 方法覆盖发生在同一类中
- B. 方法的覆盖发生在子类型中
- C. 方法名一定要一样
- D. 参数类型一定要一样
- E. 返回类型一定要一样
- F. 访问权限只能一样

【答案】BCDE

【难度】

【解析】函数覆盖发生在父类与子类之间，其函数名(方法名)、参数类型、返回值类型必须同父类中的相对应被覆盖的函数严格一致，子类方法不能缩小所覆盖方法的访问权限，子类方法不能抛出比所覆盖方法更多的异常

【分数】

6. 继承机制的作用是 (            )
- A. 信息隐藏
  - B. 数据封装
  - C. 派生新类
  - D. 数据抽象

【答案】C

【难度】

【解析】

【分数】

7. 一个类中的静态成员方法的调用可以：( )

- A. 通过类名调用
- B. 通过实例化的对象调用
- C. 在主方法中直接调用
- D. 在同一个类中的任何方法中直接调用

**【答案】**A

**【难度】**

**【解析】**

**【分数】**

8. 下列关于基类和派生类关系的叙述中，正确的是 ( )

- A. 每个类最多只能有一个直接基类
- B. 派生类中的成员可以访问基类中的任何成员
- C. 基类的构造函数必须在派生类的构造函数体中调用
- D. 派生类除了继承基类的成员，还可以定义新的成员

**【答案】**D

**【难度】**

**【解析】**

**【分数】**

9. 有关类和对象的说法不正确的是 ( )

- A. 对象是类的实例
- B. 一个类只有一个对象
- C. 任何一个对象只能属于一个类
- D. 类与对象的关系和数据与变量的关系相似

**【答案】**B

**【难度】**

**【解析】**

**【分数】**



10. C++语言中，若只需要通过一个成员函数读取数据成员的值，而不需要修改它，则应在函数头的后面加上 static 关键字。（      ）

- A. 对
- B. 错

**【答案】**B

**【难度】**

**【解析】**CONST

**【分数】**

11. 关于 virtual void Draw()=0，下面说法正确的有几个？

- (1)它是纯虚函数
  - (2)它在定义它的类中不能实现
  - (3)定义它的类不可实例化
  - (4)如果一个类要继承一个 ADT 类，必须要实现其中的所有纯虚函数
- A. 1
  - B. 2
  - C. 3
  - D. 4

**【答案】**C

**【难度】**

**【解析】**(1)(2)(4)对，有纯虚函数的类不一定是接口类，所以（3），不对

**【分数】**

12. Class A;

Class B;

void F() {

    A a;

    B b;

}

在函数 F 中，本地变量 a 和 b 的构造函数(constructor)和析构函数(destructor)的调用顺序是：

- A. b 构造 a 构造 a 析构 b 析构
- B. a 构造 a 析构 b 构造 b 析构
- C. b 构造 a 构造 b 析构 a 析构
- D. a 构造 b 构造 b 析构 a 析构

【答案】D

【难度】

【解析】构造顺序与析构顺序相反

【分数】

13. 假定指针变量 p 定义为 “int \*p=new int(100);”，要释放 p 所指向的动态内存，应使用语句（ ）

delete p;  
delete \*p;  
delete &p;  
delete []p;

【答案】A

【难度】

【解析】delete 是关键字，释放变量：delete p；释放数组时 delete [] p；

【分数】

14. 在 C++，下列哪一个可以做为对象继承之间的转换

- A. static\_cast
- B. dynamic\_cast
- C. const\_cast
- D. reinterpret\_cast

【答案】B

【难度】

【解析】dynamic\_cast：继承体系安全向下转型或跨系转型；找出某对象占用内存的起始点；static\_cast：同旧式C转型，如int到double；const\_cast：常用于去除某个对象的常量性；reinterpret\_cast：不具备移植性，常见用途是转化函数指针类型

【分数】

15. 当参数\*x=1, \*y=1, \*z=1时，下列不可能是函数add的返回值的( )?

```
int add(int *x, int *y, int *z) {  
    *x += *x;  
    *y += *x;  
    *z += *y;  
    return *z;  
}
```

正确答案: D 你的答案: 空 (错误)

- A. 4
- B. 5
- C. 6
- D. 7

【答案】A

【难度】

【解析】

```
#include "StdAfx.h"  
#include<stdio.h>  
int add(int *x, int *y, int *z) {  
    *x += *x;  
    *y += *x;  
    *z += *y;  
    return *z;  
}  
  
void main()  
{  
    //划分三块内存区域
```

```

int a=1, b=1, c=1;
//注意经过一次 add() 之后, a, b, c 的值有可能发生改变, 注意要值要归 1
printf("指向同一内存区域%d\n", add(&a, &a, &a)); //8
a=1, b=1, c=1;
printf("指向不同内存区域%d\n", add(&a, &b, &c)); //4
a=1, b=1, c=1;
printf("x.y 指向同一内存区域%d\n", add(&a, &a, &b)); //5
a=1, b=1, c=1;
printf("x.z 指向同一内存区域%d\n", add(&a, &c, &a)); //5
a=1, b=1, c=1;
printf("y.z 指向同一内存区域%d\n", add(&a, &c, &c)); //6
getchar();
}

```

**【分数】**

16. 下面哪一个是 sort 的 template 的正确写法

```

void sort(class A first, class A last, class B pred)
void template(class A, class B) sort(A first, A last, B pred)
template<class A><class B> void sort(A first, A last, B pred)
template<class A, class B> void sort(A first, A last, B pred)

```

**【答案】** D

**【难度】**

**【解析】** template<class A, class B> void sort(A first, A last, B pred)

**【分数】**

17. 有一个类 B 继承自类 A, 他们数据成员如下:

```

class A {
...
private:
    int &a;
};

class B : public A {

```

```

...
private:
    int a;
public:
    const int b;
    A c;
    static const char* d;
    A* e;
};

```

则构造函数中，成员变量一定要通过初始化列表来初始化的是\_\_\_\_\_。

- A. b c
- B. b c e
- C. b c d e
- D. c e
- E. b d
- F. b e

**【答案】**A

**【难度】**

**【解析】**const 成员和对象成员在初始化列表中初始化

**【分数】**

18. 以下程序输出结果是\_\_\_\_\_。

```

class A
{
public:
    virtual void func(int val = 1)
    { std::cout<<"A->"<<val <<std::endl;}
    virtual void test()
    { func();}
};

class B : public A
{
public:

```

```

        void func(int val=0)
    {std::cout<<"B->"<<val <<std::endl;}
};

int main(int argc ,char* argv[])
{
    B*p = new B;
    p->test();
return 0;
}

```

正确答案: B 你的答案: 空 (错误)

- A. A->0
- B. B->1
- C. A->1
- D. B->0
- E. 编译出错
- F. 以上都不对

【答案】B

【难度】

【解析】B类中没有重写test(),所以调用A类中的test(),test()调用func()函数,func(),声明为virtual,所以运行B类中的func(),参数静态绑定,输出A类中默认参数。所以输出B->1

【分数】

19. 有如下类模板定义: ( )

```

template<class T> class BigNumber{
    long n;
public:
    BigNumber(T i):n(i) {}
    BigNumber operator+(BigNumber b)
    {
        return BigNumber(n+b.n);
    }
};

```

已知 b1, b2 是 BigNumber 的两个对象, 则下列表达式中错误的是?

正确答案: D 你的答案: 空 (错误)

- A. 3+3
- B. b1+3
- C. b1+b2
- D. 3+b1

【答案】D

【难度】

【解析】3+3 : 普通的 int 相加;b1+3 : 相当于 b1 + BigNumber(3), 再调用重载的+运算;b1+b2 : 调用重载的+运算;3+b1 : 错误, b1 无法自动转换成 int。

【分数】

20. 下面对析构函数的正确描述是:

- A. 系统不能提供默认的析构函数
- B. 析构函数必须由用户定义
- C. 析构函数没有参数
- D. 析构函数可以设置默认参数

【答案】C

【难度】

【解析】

【分数】

## 卷七

1. 在公有派生的情况下, 派生类中定义的成员函数只能访问原基类的()

- A. 公有成员和私有成员
- B. 私有成员和保护成员
- C. 公有成员和保护成员
- D. 私有成员, 保护成员和公有成员

【答案】C

【难度】

【解析】

【分数】

2. 下面关于类定义的说法中, 正确的是:

- A. 类定义中包括数据成员和函数成员的声明

- B. 类成员的缺省访问权限是保护的
- C. 数据成员必须被声明为私有的
- D. 成员函数只能在类体外进行定义

【答案】A

【难度】

【解析】

【分数】

3. 关于内联函数说法错误的是：

- A. 不是任何一个函数都可定义成内联函数
- B. 内联函数的函数体内不能含有复杂的结构控制语句
- C. 递归函数可以被用来作为内联函数
- D. 内联函数一般适合于只有 1~5 行语句的小函数

【答案】C

【难度】

【解析】内联函数是使用 inline 关键字声明的函数，也成内嵌函数，它主要的作用是解决程序的运行效率。

使用内联函数的时候要注意：

- 1. 递归函数不能定义为内联函数
- 2. 内联函数一般适合于不存在 while 和 switch 等复杂的结构且只有 1~5 条语句的小函数上，否则编译系统将该函数视为普通函数。
- 3. 内联函数只能先定义后使用，否则编译系统也会把它认为是普通函数。
- 4. 对内联函数不能进行异常的接口声明。

【分数】

4. 如果一个方法或变量是“private”访问级别，那么它的访问范围是：

正确答案：D 你的答案：空（错误）

- A. 在当前类，或者子类中
- B. 在当前类或者它的父类中
- C. 在当前类，或者它所有的父类中
- D. 在当前类中

【答案】D

【难度】

【解析】

【分数】



```

5. #include<iostream>

using namespace std;

class TestClass{

    char x;

public:

    TestClass() { cout << 'A'; }

    TestClass(char c) { cout << c; }

    ~TestClass() { cout << 'B'; }

};

int main() {

    TestClass p1, *p2;

    p2 = new TestClass('X');

    delete p2;

    return 0;

}

```

执行这个程序的结果是（）

- A. XB
- B. AXBB
- C. AAXB
- D. AAXBBB

**【答案】** B

**【难度】**

**【解析】**

**【分数】**

6. 以下是 C++ 的不同数据类型值的比较语句，请问这些判断语句中作为条件部分的语句编写有问题的有：

- A. 如果变量 bVar 是布尔类型：if(false==bVar){doSomething();}
- B. 如果变量 nVar 是 int 型：if(0==nVar){doSomething();}
- C. 如果变量 fVar 为浮点型：if(0.02=fVar){doSomething();}
- D. 如果变量 sVar 为字符串型：if( ""==sVar){doSomething();}

**【答案】** C

**【难度】**

【解析】浮点型判断不能用==来判断，用一个范围来判断

【分数】

7. 请选择下列程序的运行结果

```
#include<iostream>

using namespace std;

class B0//基类 B0 声明
{
public://外部接口
virtual void display()//虚成员函数
{
    cout<<"B0::display0"<<endl;}
};

class B1:public B0//公有派生
{
public:
    void display() { cout<<"B1::display0"<<endl; }
};

class D1: public B1//公有派生
{
public:
    void display() { cout<<"D1::display0"<<endl; }
};

void fun(B0 ptr)//普通函数
{
    ptr.display();
}

int main()//主函数
{
    B0 b0;//声明基类对象和指针
    B1 b1;//声明派生类对象
    D1 d1;//声明派生类对象
    fun(b0);//调用基类 B0 函数成员
    fun(b1);//调用派生类 B1 函数成员
```

```

        fun(d1); //调用派生类 D1 函数成员
    }

```

正确答案: A 你的答案: 空 (错误)

- A. B0::display0 B0::display0 B0::display0
- B. B0::display0 B0::display0 D1::display0
- C. B0::display0 B1::display0 D1::display0
- D. B0::display0 B1::display0 B1::display0

【答案】A

【难度】

【解析】此题的关键点在于 fun 函数，传入的参数是一个类的对象，这样，派生类作为参数传入的时候，会自动的类型转换为基类对象，这样，display 就只是执行基类的函数了。选 B0::display() B0::display() B0::display()

【分数】

8. 查看下面两段代码

(1)

```

class classA {...};

class classB
{
public:
    classB(classA a) {mA = a;}

private:
    classA mA;
};

```

(2)

```

class classA {...};

class classB
{
public:
    classB(classA a): mA(a) {}

private:
    classA mA;
};

```

正确答案: B 你的答案: 空 (错误)

- A. 两种方法产生的结果和效率完全相同
- B. 通常情况下 (2) 的效率更高
- C. 通常情况下 (1) 的效率更高
- D. 两种调用方法会产生不同的结果

【答案】C

【难度】

【解析】初始化列表在构造函数之前执行

【分数】

9. 当一个类对象的生命周期结束后, 关于调用析构函数的描述正确的是: ( )

- A. 如果派生类没有定义析构函数, 则只调用基类的析构函数
- B. 如果基类没有定义析构函数, 则只调用派生类的析构函数
- C. 先调用派生类的析构函数, 后调用基类的析构函数
- D. 先调用基类的析构函数, 后调用派生类的析构函数

【答案】C

【难度】

【解析】没有定义析构函数和构造函数, 编译器会自动提供析构函数和构造函数

【分数】

10. 下列情况中, 不会调用拷贝构造函数的是 ( )

- A. 用一个对象去初始化同一个类的另一个新对象时
- B. 将类的一个对象赋值给该类的另一个对象时
- C. 函数的形参对象, 调用函数进行形参和实参结合时
- D. 函数的返回值是类的对象, 函数执行返回调用时

【答案】B

【难度】

【解析】B 调用的=重载函数

【分数】

11. 下面描述中, 正确的是

- A. 虚函数是没有实现的函数
- B. 纯虚函数的实现是在派生类中
- C. 抽象类是没有纯虚函数的类

D. 抽象类指针可以指向不同的派生类

【答案】BD

【难度】

【解析】

【分数】

12. 假定 Qiniuome 是一个类，执行下面这些语句之后，内存里创建了几个 Qiniuome 对象。

```
Qiniuome a(); // 1
Qiniuome b(2); // 2
Qiniuome c[3]; //3
Qiniuome &ra = b; // 4
Qiniuome *pA = c; //5
Qiniuome *p = new Qiniuome(4); //6
```

正确答案: A 你的答案: 空 (错误)

A. 5

B. 6

C. 7

D. 8

【答案】A

【难度】

【解析】1 不会调用过早函数，2，3 会调用四次，4，5 是指针和引用不会调用构造，6 会调用过早函数，所以一共调用 5 次

【分数】

13. 对静态成员的不正确描述是（）？

A. 静态数据成员要在类外定义和初始化

B. 非静态成员函数也可以操作静态数据成员

C. 静态成员不属于对象，是类的共享成员

D. 调用静态成员函数时要通过类或对象激活，所以静态成员函数拥有 this 指针

【答案】D

【难度】

【解析】静态成员是属于类的，不是属于对象的，而 this 是当前对象的指针，所以静态成员没有 this 指针

【分数】

14. 下列有关运算符重载的叙述中，正确的是（ ）？

- A. C++中所有运算符都可以作为非成员函数重载
- B. 重载运算符时可以改变其结合性
- C. C++中可以通过运算符重载创造新的运算符
- D. 运算符重载是多态的一种表现

【答案】D

【难度】

【解析】

【分数】

15. 在 C++里，堆中存储的数据是（ ）

- A. 局部变量
- B. 函数返回值
- C. 函数参数
- D. new 和 malloc 申请的内存

【答案】D

【难度】

【解析】

【分数】

16. 下列函数中，不能声明为虚函数的是（ ）？

- A. 构造函数
- B. 析构函数
- C. 私有成员函数
- D. 公有成员函数

【答案】A

【难度】

【解析】

【分数】

17. 类 B 和类 A 的公有派生类，类 A 和类 B 中都定义了虚函数 func()，p 是一个指向类 A 对象的指针，则 p->A::func() 将 ( ) ?

- A. 调用类 B 中函数 func()
- B. 即调用类 A 中函数，也调用类 B 中的函数
- C. 调用类 A 中函数 func()
- D. 根据 p 所指的物体类型而确定调用类 A 中或类 B 中的函数 func()

**【答案】** C

**【难度】**

**【解析】** 加作用域，调用 A 的 func()；

**【分数】**

18. 请选择下列程序的执行结果 ( )

```
#include<iostream>
using namespace std;
class A{
    public:
    void f() {
        cout<<"A::f()";
    }
};
class B: public A{
    public:
    void f() {
        cout<<"B::f()";
    };
};
void main() {
    B b;
    b.f();
}
```

- A. A::f(); B::f() 都不可能
- B. A::f(); B::f() 都有可能

- C. A::f()
- D. B::f()

【答案】D

【难度】

【解析】

【分数】

19. 面向对象程序设计语言的三个基本特征是？（ ）

- A. 封装、接口和多态
- B. 封装、继承和动态
- C. 跨平台、接口和动态
- D. 封装、继承和多态

【答案】D

【难度】

【解析】

【分数】

20. 下面描述中，正确的是

- A. 虚函数是没有实现的函数
- B. 纯虚函数的实现是在派生类中
- C. 抽象类是没有纯虚函数的类
- D. 抽象类指针可以指向不同的派生类

【答案】BD

【难度】

【解析】

【分数】



## 卷八

1. 关于 new 运算符的下列描述中，（ ）是错的。
- A. 它可以用来动态创建对象和对象数组
  - B. 使用它创建的对象和对象数组可以使用运算符 delete 删除
  - C. 使用它创建对象时要调用构造函数
  - D. 使用它创建对象数组时必须指定初始值

【答案】D

【难度】

【解析】反例：int\* x = new int();

【分数】

2. 下面叙述不正确的是（ ）
- A. 派生类一般都用公有派生
  - B. 对基类成员的访问必须是无二义性的
  - C. 赋值兼容规则也适用于多重继承的组合
  - D. 父类的公有成员在派生类中仍然是公有的

【答案】D

【难度】

【解析】跟继承关系有关，不一定是共有继承

【分数】

3. 有以下一段代码：

```
class A
{
public:
    A()
    {
        printf("A");
    }
    ~A()
    {
        printf("~A");
    }
}
```

```

void B()
{
    printf("B");
}
};

void main()
{
    A* a = (A*)malloc(sizeof(A));
    a->B();
    free(a);
}

```

请问程序最后输出结果是？

- A. A
- B. ~A
- C. B
- D. AB~A

**【答案】** C

**【难度】**

**【解析】** new delete 和 malloc free 的区别：new 一个类对象时会执行类的构造函数，delete 一个 类对象时会执行类的析构函数。所以当涉及到类的对象时，malloc free 和 new delete 不能混搭，其他情况可以混搭

**【分数】**

4. 对于派生类的构造函数，在定义对象时构造函数的执行顺序为？

- 1 ： 成员对象的构造函数
- 2 ： 基类的构造函数
- 3 ： 派生类本身的构造函数

- A. 123
- B. 231
- C. 321
- D. 213

**【答案】** D

**【难度】** 构造顺序：先父母再朋友最后自己，析构顺序：与构造顺序相反

**【解析】**

**【分数】**

5. 下列程序的输出是:

```
#define add(a+b) a+b  
  
int main()  
{  
    printf("%d\n", 5*add(3+4));  
    return 0;  
}
```

- A. 23
- B. 35
- C. 16
- D. 19

**【答案】**D

**【难度】**

**【解析】**宏定义: 简单替换  $5 * 3 + 4$

**【分数】**

6. 下列的各类函数中, ( ) 不是类的成员函数。

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝初始化构造函数

**【答案】**C

**【难度】**

**【解析】**

**【分数】**

7. 作用域运算符“::”的功能是\_\_\_\_\_

- A. 标识作用域的级别的
- B. 指出作用域的范围的
- C. 给定作用域的大小的
- D. 标识成员是属于哪个类的

**【答案】**D

**【难度】**

**【解析】**

【分数】

8. ( ) 是析构函数的特征。

- A. 一个类中只能定义一个析构函数
- B. 析构函数名与类名不同
- C. 析构函数的定义只能在类体内
- D. 析构函数可以有一个或多个参数

【答案】A

【难度】

【解析】

【分数】

9. 关于 new 运算符的下列描述中，\_\_\_\_\_是错的。

- A. 它可以用来动态创建对象和对象数组
- B. 使用它创建的对象或对象数组可以使用运算符 delete 删除
- C. 使用它创建对象时要调用构造函数
- D. 使用它创建对象数组时必须指定初始值

【答案】D

【难度】

【解析】

【分数】

10. 下列对派生类的描述中，\_\_\_\_\_是错的。

正确答案：D 你的答案：空（错误）

- A. 一个派生类可以作另一派生类的基类
- B. 派生类至少有一个基类
- C. 派生类的成员除了它自己的成员外，还包含了它的基类的成员
- D. 派生类中继承的基类成员的访问权限到派生类中保持不变

【答案】D

【难度】

【解析】

【分数】

11. 派生类的构造函数的成员初始化列表中, 不能包含()

- A. 基类的构造函数
- B. 派生类中子对象的初始化
- C. 派生类中一般数据成员的初始化
- D. 基类的子对象初始化

【答案】D

【难度】

【解析】

【分数】

12. 下列关于异常的说法, 正确的是()

- A. RuntimeException 及其子类的异常可以不做处理
- B. Catch 段中的语句, 不允许再次出现异常
- C. 在方法定义中以 throws 标识出的异常, 在调用该方法中的方法必须处理
- D. 程序中所有的可能出现的异常必须在 catch 中捕获, 否则将引起编译错误

【答案】A

【难度】

【解析】

【分数】

13. C++中, 能作为函数重载判断依据的是?

- A. 返回类型
- B. const
- C. 参数个数
- D. 参数类型

【答案】BCD

【难度】

【解析】

1. 重载函数依据形参取判定是否可以重载;
2. 形参个数不同, 或者类型不同都是可以重载的类型;
3. 对于形参为引用或者指针的情况下, const 版本和非 const 版本是有区别的, 因为传引用是有可能改变原来的参数的

**【分数】**

14. C++ 中，关于 new 运算符的错误描述是？

- A. 使用 new 运算符创建的对象使用完成后会自动销毁并回收
- B. 使用运算符 new 创建对象数组时必须定义初始值
- C. 使用运算符 new 创建对象时，会调用类的构造函数
- D. 运算符 new 可以用来动态创建对象和对象数组

**【答案】** AB

**【难度】**

**【解析】**

**【分数】**

15. 下列说法正确的有（ ）

- A. 在类的一般方法中可用 this 来调用本类的属性
- B. 在类的静态方法中可以访问本类中的属性
- C. 在类方法中绝对不能调用实例方法
- D. 在类方法中只能调用本类中的类方法

**【答案】** A

**【难度】**

**【解析】**

**【分数】**

16. 下列说法正确的有（ ）

- A. 构造方法的方法名必须与类名相同
- B. 构造方法也没有返回值，但可以定义为 void
- C. 在子类构造方法中调用父类的构造方法，super（）必须写在子类构造方法的第一行，否则编译不通过
- D. 一个类可以定义多个构造方法，如果在定义类时没有定义构造方法，则编译系统会自动插入一个默认的构造方法，这个构造方法不执行任何代码

**【答案】** ACD

**【难度】**

**【解析】**

**【分数】**

17. 以下代码是哪一句可能导致的编译错误?

```
#include <iostream>

using namespace std;

class Test
{
public:
    Test() { cout << "i like zk"; }
    Test(int) { cout << "i like wangyi"; }
    void exe() { cout << "this is my choice"; }
};

int main()
{
    Test a(1); //1
    a.exe(); //2
    Test b(); //3
    b.exe(); //4
    Test c; //5
    c.exe(); //6

    return 0;
}
```

A. 1

B. 2

C. 3

D. 4

E. 5

F. 6

**【答案】**D

**【难度】**

**【解析】**

**【分数】**

18. 以下表达式选择结果。( )

```
int a = 0;
```

```
int b = (a=-1) ? 2:3;
int c = (a=0) ? 2:3;
```

- A. b=2, c=2
- B. b=3, c=3
- C. b=2, c=3
- D. b=3, c=2

【答案】C

【难度】

【解析】

【分数】

如果定义如下类：

```
class Empty{}
```

请选择编译器为之生成的函数有哪些？（ ）

```
1Empty() { ... }
2Empty(const Empty& rhs) { ... }
3Empty& operator=(const Empty& rhs) { ... }
4~Empty() { ... }
```

- A. 124
- B. 1234
- C. 123
- D. 14

【答案】B

【难度】

【解析】默认构造函数（空的和拷贝构造函数），析构函数，赋值运算符重载

【分数】

20. 关于纯虚函数，下列表述正确的是？（ ）

- A. 派生类必须实现基类的纯虚函数
- B. 纯虚函数的声明总是以“=0”结束
- C. 纯虚函数是给出实现版本（即无函数体定义）的虚函数
- D. 含有纯虚函数的类一定是派生类

【答案】B



【难度】

【解析】

【分数】

