

C++ 模拟试题 1

一、 选择题（在 A、B、C、D 四个答案中选择一个正确的填入后面的括号中）

1. 下面 (D) 是对类 ttt 的声明。

- A. void class ttt;
- B. class int ttt;
- C. class ttt();
- D. class ttt;

2. 用于类中虚成员函数说明的关键字是 (B)

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

3. 下面是关于静态成员的说法，其中不正确的是 (D)。

- A. 静态成员有类作用域，但与普通非静态成员有所不同
- B. 静态函数没有 this 指针，同一个类的不同对象拥有相同的静态数据成员
- C. 静态数据成员的初始化必须在类外进行
- D. 静态函数同样可以访问非静态数据成员 *通过对象访问*

4. 下面的说法中，不正确的一句是 (A)。

- A. 类的友元函数是类的特殊成员函数，它对所有的类成员、保护成员和私有成员享有完全访问特权。
- B. 一个类的友元类中的成员函数都是这个类的友元函数
- C. 适用于类成员函数和数据的访问控制同样适用于枚举和嵌套类
- D. 与结构类似，类也可以互相嵌套，适用于嵌套结构的规则也适用于嵌套类

5. (D) 语言是面向对象的语言。

- A. C
- B. Pascal
- C. Fortran
- D. C++

6. C++ 风格的源文件包含的输入输出头文件为 (C)

- A. stdio.h
- B. stdafx.h
- C. iostream.h
- D. stream.h

7. 与 C 语言 printf("Hello word\n") ; 语句功能相同的 C++ 语句是 (C)

- A. cout<<"Hello word\n"
- B. cin<<"Hello word\n"
- C. cout<<"Hello word\n"
- D. cin<<"Hello word\n"

8. 在下面几个元素中，不是面向对象的要素的是 (A)；

- A. 重载
- B. 对象
- C. 类
- D. 继承

9. 对于 int (*pa)[5] 的描述，下面叙述正确的是 (C)；

- A. pa 是一个指向数组的指针，所指向的数组是 5 个 int 型元素； *int(*p)[5]*
- B. pa 是一个指向某数组中第 5 个元素的指针，该元素是 int 型变量；
- C. pa 是一个具有 5 个元素的指针数组，每个元素是一个 int 型指针； *看一些知识点*
- D. pa[5] 表示某个数组的第 5 个元素的值。

10. 下面程序使用 new 与 delete 命令：

```
#include <iostream.h>
```

```
main()
```

```
{
```

```
    long *p, i;
```

```
    p=new long;
```

```

if(!p) {
    cout<<"Allocation error\n";
    return 1;
}
*p=1000;
i=(long)p;
cout<<"Here is integer at p:"<<*p<<"\n";
cout<<"Here is i: "<<i<<"\n";
delete p;
return 0;
}

```

下面是对这段程序的叙述，其中正确的是 ()。

- A. 有错误，编译不能通过
- B. 正确，执行结果为：1000 1000
- C. 正确，但执行结果的第二个数值不定
- D. 正确，执行结果的各个数值均不为 1000

二、判断下列描述正确性，对者划√，错者划×（每小题 2 分）。

1. C++中标识符内的大小写字母是没有区别的。✓
2. 宏定义命令是分号结束的。✓
3. C++的程序中，对变量一定要先说明再使用，说明只要在使用之前就可以。✓
4. while 循环语句的循环体至少执行一次。✓
5. 在 C++中，所有函数在调用前都要说明。✗ 定义在前，就不需要说明
6. 析构函数是一种函数体为空的成员函数。✗
7. 如果一个函数没有返回值时，定义时需要用 void 说明。✓
8. 派生类是从基类派生出来的，它不能再生成新的派生类。✗
9. 对象成员的表示与结构变量成员表示相同，使用运算符. 或->。✓
10. 构造函数和析构函数都可以重载，因为它们都是成员函数。✓

三、写出下面程序的输出结果。

```

1. #include<iostream.h>
void main()
{ int a,b,c;
  a=1,b=2,c=a+b+3;
  cout<<a<<" "<<b<<" "<<c<<endl;
  c=(a++,a+=6,a+b); cout<<c<<endl;
}

```

考点是逗号运算符

c=6

1, 2, 6

10

Att

```

2. #include <iostream.h>
void swap(int &x, int &y)
{
    int temp;
    temp=x;
    x=y;
    y=temp;
}

```

引用

C=12

```

    }
    cout<<"x="<<x<<"", "<<"y="<<y<<endl;
}

void main()
{
    int a(5), b(9);
    swap(a, b);
    cout<<"a="<<a<<"", "<<"b="<<b<<endl;
}

```

注意这里有输出

3. #include <iostream.h>

```

class A
{
public:
    A();
    A(int i, int j);
    void print();
private:
    int a, b;
};

```

```

A::A()
{
    a=b=0;
    cout<<"Default constructor called.\n";
}

```

```

A::A(int i, int j)
{
    a=i; b=j;
    cout<<"Constructor called.\n";
}

```

```

void A::print()
{
    cout<<"a="<<a<<"", b="<<b<<endl;
}

```

```

void main()
{
    A m, n(4, 8);
    m.print();
    n.print();
}

```

Default constructor called.
Constructor called.

a=0, b=0
a=4, b=8

四、编程或程序填空。

1. 在下划线处填上正确的语句并给出程序的输出结果（6分）。

```

#include <iostream.h>
#include <cmath>
class strA
{
public:
    strA(float, float, float);
}

```

```

float fun1( )
double fun2( )
public:
    float a, b, c;
};
float strA::fun1( )
{
    return a+b+c;
}
strA::strA(float i, float j, float k)
{
    a=i; b=j; c=k;
}
double strA::fun2( )
{
    float s;
    s=float((a+b+c)/2.0);
    return sqrt(s*(s-a)*(s-b)*(s-c));
}
void maina( )
{
    strA tri(3, 4, 5);
    cout<<tri.fun1( )<<endl;
    cout<<tri.fun2( )<<endl;
}

```

2. 编写分别可求两个整型、浮点型和双精度型数和的重载函数，并写主函数调用之；
3. 使用引用作为函数参数，编写一个交换两变量值的函数，并写主函数调用之。

C++模拟试题 2

一、选择题（在 A、B、C、D 四个答案中选择一个正确的填入后面的括号中）

1. C++中，封装是借助于（B）达到的。
E. 指针 B. 类 C. 数组 D. 函数
2. 宏定义由（C）结束。
A. 分号 B. 句号 C. 新行 D. 冒号
3. 下面是关于一个类的友元的说法，其中不正确的是（C）
A. 友元函数可以访问该类的私有数据成员
B. 友元的声明必须放在类的内部
C. 友元函数可以是类的成员函数
D. 类与类之间的友元关系不能被继承
4. C++中声明常量的关键字是（A）。
A. const B. extern C. public D. enum
5. 关于动态联编的下列描述中，错误的是（D）。
A. 动态联编是以虚函数为基础的

- B. 动态联编时在运行时确定所调用的函数代码的
 C. 动态联编调用函数操作是使用指向对象的指针或对象引用
 D. 动态联编是在编译时确定操作函数的 运行
6. 下面函数 (C) 的表示方法说明它使用对象的引用作为参数。
 A. test(pt *p) B. test(pt p)
 C. test(pt &p) D. test(const P)
7. 虚函数的调用 (D)。
 A. 一定使用动态联编
 B. 必须使用动态联编
 C. 一定使用静态联编
 D. 不一定使用动态联编
- [解析] 虚函数在运行阶段和类的对象绑定在一起, 这样成为动态联编。虚函数声明只能出现在类声明中虚函数原型中, 而不能在成员的函数体实现的时候。必须由成员函数来调用或者通过指针、引用来访问虚函数。如果通过对象名来访问虚函数, 则联编在编译过程中认为是静态联编。
8. 关于 C++ 与 C 语言的关系的描述中, 错误的是 (D);
 A. C 语言是 C++ 的子集; B. C 语言与 C++ 是兼容的;
 C. C++ 对 C 语言进行了一些改进; D. C++ 和 C 语言都是面向对象的。
9. 重载函数在调用时选择的依据中, (D) 是错误的。
 A. 参数个数 B. 参数类型 C. 函数名字 D. 函数类型
10. 下面的程序是关于函数承载的, 假定 A, B, C, D 是它们可能的输出之一, 其中正确的输出是 (A)。
 A. 25 10 B. 20 10 C. 20 11.4 D. 20 14.076

源程序如下:

```
#include<iostream.h>
int Multiply(int n1, int n2)
{
    return n1*n2;
}
float Multiply(float f1, float f2)
{
    return f1*f2;
}
void main()
{
    int a1=10;
    int a2=2.5;
    float b1=2.55;
    float b2=5.52;
    cout<<Multiply(a1,a2)<<endl;
    cout<<Multiply(b1,b2)<<endl;
}
```

20

答案为

20 11.04

二、判断下列描述正确性, 对者划√, 错者划×。

1. C++ 中数组元素的下标是从 0 开始, 数组元素是连续存储在内存单元中的。√
2. 函数一定要有函数体。× 引用 指针
3. 在说明语句 int a(5), &b=a, *p=&a; 中 b 的值和 *p 值是相等的。√
4. for 循环是只有可以确定的循环次数时才可使用, 否则不能用 for 循环。×

5. 在C++中, 定义函数时必须给出函数的类型。✓
6. 对象引用作函数参数不如使用对象指针更方便。X 3 | 用更多更
7. 说明或定义对象时, 类名前必须要加class关键字。X 不同加
8. C++语言中, 允许单一继承, 不允许多重继承。X 四大
9. 面向对象方法具有三大特性: 封装性、继承性和继承性。X 封装性
10. 在共有继承中, 派生类可以访问基类中的私有成员。X

三、按要求写结果:

1. 计算下列表达式的值:

(1) 已知: `int i(10), j(5);`

A. `++i--`

B. `i*=j;`

C. `i=3/2*(j=1)`

(2) 已知: `int a(5), b(3);` 计算下面表达式的值及 a 和 b 的值。

A. `!a && b++;`

B. `a || b + 4 && a*b;`

C. `a=1, b=2, a>b ? ++a : ++b;`

A 0 a=5 b=3

B 1 a=5 b=3

C 3 a=1 b=3

2. 写出下面程序的输出结果。

(1) `#include <iostream.h>`

`void fun()`

`{`

`static int n=25;`

`n--;`

`cout <<"n="<<n<<endl;`

`}`

`void main()`

`{`

`for(int i=0; i<2; i++)`

`fun();`

`}`

(2) `#include <iostream.h>`

`class Base`

`{`

`public :`

`void Setc(char a) {c2=(c1=a)-32;}`

`void Show()`

`{ cout<<c1<<" can be upwritten as: "<<c2<<endl;}`

`private:`

`char c1, c2;`

`};`

`void main()`

`{`

`Base a, b;`

`a.Setc('a');`

`b.Setc('b');`

`a.Show();`

$c_2 = a - 32$

$c_2 = A$

a can be upwritten as: A

b can be upwritten as: B

```

        b. Show( );
    }

```

(3) 定义类 Tpoint 如下:

```

class TPoint
{
public:
    TPoint(int x,int y){X=x;Y=y;}
    TPoint(TPoint &p);
    ~ TPoint( ) {cout<<"destructor Called.\n";}
    int GetX( ) {return X;}
    int GetY( ) {return Y;}
private :
    int X,Y;
};
TPoint::TPoint(TPoint &p)
{
    X=p.X;
    Y=p.Y;
}

```

并将其存入 tpoint.h 文件中, 分析并写出下面主函数的执行结果:

```

#include <iostream.h>
#include <tpoint.h>
void main( )
{
    TPoint P1(5,7);
    TPoint P2(P1);
    cout<<"P2="<<P2.GetX( )<<"", "<<P2.GetY( )<<endl;
}

```

*P2 = 5, 7
destructor called.
destructor called.*

四、编程或程序填空。

1. 在下划线处填上正确的语句成分, 并编写一个测试主程序。

```

#include<iostream.h>
class Test
{
public :
    Test(int a,int b){ x=a; x=b; (1) }
    void show( ) { (2) }
private :
    int x,y;
};

```

*(1) x=a; x=b;
(2) cout << x << y << endl;*

2. 编写分别可计算 2、3 或 4 个 int 型数的和的重载函数, 并写主函数调用之;
3. 设计一个类, 使它具有一个求两个整数中较大者的成员函数以及输出成员函数, 并写一个测试程序。