

考试科目：C++ 面向对象程序设计（计科） A 卷

题号	1	2	3	4	5	6	7	8	9	10	总分
得分											

得分	
----	--

一、选择题(每题 2 分，共 20 分)

1. 可用作 C++ 语言用户标识符的一组标识符是(B)。
- A. void define +WORD B. a3_b3 _123 YN
- C. for -abc Case D. 2a DO sizeof
2. 用 new 运算符创建一个含 10 个元素的一维整型数组的正确语句是(C)。
- A. int *p=new a[10]; B. int *p=new float[10];
- C. int *p=new int[10]; D. int *p=new int[10]={1,2,3,4,5}
3. 在成员函数中进行双目运算符重载时，其参数表中应带有（ B ）个参数。
- A. 0 B. 1 C. 2 D. 3
4. 循环语句 “for(int i=0; i<n; i++) cout<<i*i<<' ';” 中循环体执行的次数为(C)。
- A. 1 B. n-1 C. n D. n+1
5. 在下面的一维数组定义中，哪一个有语法错误。（ D ）
- A. int a[]={1,2,3}; B. int a[10]={0};
- C. int a[5]; D. int a[];
6. 下列（ C ）的调用方式是引用调用。
- A. 形参和实参都是变量 B.形参是指针，实参是地址值
- C. 形参是引用，实参是变量 D.形参是变量，实参是地址值
7. C++中函数返回值的类型是由（ B ）决定的。
- A. return 语句中表达式的类型 B.该函数定义时的类型
- C.调用函数时的调用语句 D.系统根据结果
8. 若一个二维数组的定义为 “int a[3][4]={{3,4},{2,8,6}};”, 则元素 a[1][2]的值为（ C ）。
- A. 2 B. 4 C. 6 D. 8
9. 面向对象方法的多态性是指（ A ）。
- A. 针对一消息，不同的对象可以以适合自身的方式加以响应
- B. 一个对象在不同的运行环境中可以有不同的变体
- C. 一个类可以派生出多个特殊类
- D. 一个对象可以是由多个其他对象组合而成的
10. 假定 AA 为一个类，a 为该类公有的数据成员，若要在该类的一个成员函数中访问它，则书写格式为（ A ）。
- A. a B. AA::a C. a() D. AA::a()

y * yx + x

得分	
----	--

二、填空（每空 1 分，共 20 分）

1. 若 x=5, y=10, 则计算 y*=++x 表达式后，x 和 y 的值分别为 6 和 60。
2. 重载一个函数的条件是：该函数必须在参数的 个数 或参数的 类型 上与其它同名函数有所不同。
3. 带有 纯虚函数 的类是抽象类。
4. 友元函数是在类中用关键字 friend 修饰的非成员函数。
5. p 是指向整数的指针，则用 *p 表示该整数对象，用 &p 表示指针变量 p 的地址。
6. 在 C++ 中，多态性有 4 类分别是 重载多态、强制多态、包含多态(虚函数)、和 参数多态(模板)。
7. 派生类的成员函数可直接访问基类的 公用 和 保护 成员，不能直接访问基类的 私有 成员。
8. 静态数据成员必须在所有函数的定义体外进行 定义。
9. 面向对象的程序设计方法的三个特性是封装、继承 和 多态。
10. 使用 const 语句定义一个标识符常量时，则必须对它同时进行 初始化（或定义）。
11. 一个类的 构造 函数实现对该类对象的初始化功能。

得分	
----	--

三、简答题（9 分，每题 3 分）

1. 什么是虚基类，并简述其特点。
- 答：以 virtual 方式继承基类就是虚基类。将共同基类设置为虚基类时，从不同路径继承过来的同名数据成员只有一个副本，同一个函数也只有一个映射。
2. 派生类对基类成员的继承方式有哪些？
- 答：公有继承（public），私有继承（private），保护继承（protected）
3. C++的作用域有哪几种？
- 答：（函数）原型作用域，局部（块）作用域，类作用域，命名空间（文件）作用域

得分	
----	--

四、程序填空（每题 6 分，共 12 分）

1. 斐波那契数列的第 1 和第 2 个数分别为 0 和 1，从第三个数开始，每个数等于其前两个数之和。求斐波那契数列中的前 20 个数，要求每行输出 5 个数。
- ```
#include <iostream>

using namespace std;

int main() {
```

```
int f[20] = {0,1};
for (int i = 2; i < 20; i++)
 __ (1) __
for (i=0;i<20;i++) {
 if (__ (2) __) cout << endl;
 cout.width(12);
 __ (3) __;
}
return 0;
}
```

答案： (1) f[i] = f[i - 2] + f[i - 1]; (2) i%5==0 (3) cout << f[i];

2. 在下面一段类定义中，Derived 类是由直接基类 Base 1 和 Base 2 公有派生的，Derived 类包含有两个间接基类 BaseBase，在 Derived 类的初始化函数 Init 中，需要把 x1 和 x2 的值分别赋给属于基类 Base1 的 x 成员和属于基类 Base2 的 x 成员。

```
class BaseBase {
protected: int x;
public: BaseBase(){ x = 1;}
};
class Base1: public BaseBase {
public: Base1(){ }
};
class Base2: public BaseBase {
public: Base2(){ }
};
class Derived: __ (1) __
{
public: Derived() { }
void Init(int x1, int x2) {
 __ (2) __;
 __ (3) __;
}
void output() {cout<<Base1::x<<' '<<Base2::x<<endl;}
```

};
答案： (1) public Base1, public Base2 (2) Base1::x=x1 (3) Base2::x=x2

|    |  |
|----|--|
| 得分 |  |
|----|--|

五、指出函数功能（每题 6 分，共 12 分）

```
1. #include<iostream.h>
void main()
{ int i,p=1,s=0;
 int N;
 cout<<"输入一个正整数:";
 cin>>N;
 for(i=1;i<=N;i++) {
 p*=i; s+=p;
 }
 cout<<s<<endl;
```

$\sum_{i=1}^N i!$  答案： 计算并输出  $\sum_{i=1}^N i!$  的值，其中 N 值由键盘输入。

```
2. int f(const char *s)
{ int len=0;
 while(*s++) len++;
 return len;
};答案： 求出并返回字符指针参数 s 所指向的字符串长度。
```

|    |  |
|----|--|
| 得分 |  |
|----|--|

六、程序改错（每题 6 分，共 12 分）

1. 下面是一个类的定义，存在着 3 处语法错误，请指出错误行的行号并改正。

```
class CE { //1 行
private: //2 行
 int a, b; //3 行
 int getmin() {return (a<b? a:b);} //4 行
public //5 行
 int c; //6 行
```

```
int SetValue(int x1, int x2, int x3) { //7 行
 a=x1; b=x2; c=x3; //8 行
} //9 行
int GetMin(); //10 行
}; //11 行
int GetMin() //12 行
{ //13 行
 int d=getmin(); //14 行
 return (d<c? d:c); //15 行
} //16 行
```

```
int d=1; //18 行
for(int i=0; i<4; i++) d*=x.a[i]; //19 行
int f=SumA(6); //20 行
cout<<"d="<<d<<','; //21 行
cout<<"f="<<f<<endl; //22 行
} //23 行
```

答案：5            7            12

public:     void SetValue(int x1, int x2, int x3) {            int CE::GetMin()

2. 假定要求下面程序输出结果为 “d=800,f=60”，在第 4-23 行中存在着三条语句错误，请指出错误语句的行号并改正。

```
#include<iostream.h>
class A { //1 行
 int a[10]; int n; //2 行
public: //3 行
 A(int aa[], int nn): n(nn) { //4 行
 for(int i=0; i<n; i++) aa[i]=a[i]; //5 行
 } //6 行
 int Get(int i) {return a[i];} //7 行
 int SumA(int n); //8 行
}; //9 行
int A::SumA(int n) { //10 行
 int s=0; //11 行
 for(int j=0; j<n; j++) s+=a[j]; //12 行
 return s; //13 行
} //14 行
void main() { //15 行
 int a[]={2,5,8,10,15,20}; //16 行
 A x(a,6); //17 行
```