



7. 以下程序的输出结果是

```
#include <iostream>
void main( )
{char *p[10]={"abc","aabdfg","dcdbe","abbd","cd"};
  cout<<p[3]<<endl;}

```

- A. dcdbe
- B. abbd
- C. abc
- D. abb

8. 下面的哪个保留字不能作为函数的返回类型

- A. void
- B. int
- C. new
- D. long

9. 当派生类从一个基类保护继承时，基类中的一些成员在派生类中成为保护成员，这些成员在基类中原有的访问属性是

- A. 任何
- B. 公有或保护
- C. 保护或私有
- D. 私有

10. 在公有派生情况下，有关派生类对象和基类对象的关系，下列叙述不正确的是

- A. 派生类的对象可以赋给基类的对象
- B. 派生类的对象可以初始化基类的引用
- C. 派生类的对象可以直接访问基类中的成员
- D. 派生类的对象的地址可以赋给指向基类的指针

11. 友员的作用之一是

- A. 提高程序的运行效率
- B. 加强类的封装
- C. 实现数据的隐蔽性
- D. 增加成员函数的种类

12. 若要对 Data 类中重载的加法运算符成员函数进行声明，下列选项中正确的是

- A. Data +(Data);
- B. Data operator+(Data);
- C. Data +operator(Data);
- D. operator+(Data,Data);

13. 下列关于函数模板的描述中，正确的是

- A. 函数模板是一个实例函数
- B. 使用函数模板定义的函数没有返回类型
- C. 函数模板的类型参数与函数的参数相同
- D. 通过使用不同的类型参数，可以从函数模板得到不同的实例函数

14. f1(int)是类 A 的公有成员函数，p 是指向成员函数 f1()的指针，正确的语句是

- A. p=f1;
- B. p=A::f1;
- C. p=A::f1();
- D. p=f1();

15. 下面对静态数据成员的描述中，正确的是
- A. 类的不同对象有不同的静态数据成员值
  - B. 类的每个对象都有自己的静态数据成员
  - C. 静态数据成员是类的所有对象共享的数据
  - D. 静态数据成员不能通过类的对象调用
16. 关于封装，下列说法中不正确的是
- A. 通过封装，对象的全部属性和操作结合在一起，形成一个整体
  - B. 通过封装，一个对象的实现细节被尽可能地隐藏起来
  - C. 通过封装，每个对象都成为相对独立的实体
  - D. 通过封装，对象的属性都是不可见的
17. 关于 new 运算符的下列描述中，错误的是
- 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
18. 在公有继承的情况下，基类成员在派生类中的访问权限
- A. 受限制
  - B. 保持不变
  - C. 受保护
  - D. 不受保护
19. 定义析构函数时，应该注意
- A. 其名与类名完全相同
  - B. 返回类型是 void 类型
  - C. 无形参，也不可重载
  - D. 函数体中必须有 delete 语句
20. 所谓多态性是指
- A. 不同的对象调用不同名称的函数
  - B. 不同的对象调用相同名称的函数
  - C. 一个对象调用不同名称的函数
  - D. 一个对象调用不同名称的对象

## 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 20 空，每空 1 分，共 20 分。

21. C++语言支持的两种多态性分别是编译时的多态性和\_\_\_\_\_的多态性。
22. 当不需要函数返回任何值时，则应把该函数类型定义为\_\_\_\_\_。
23. 在函数体内部定义的变量称为\_\_\_\_\_，在所有函数体外部定义的变量称为全局变量。

24. new 用来动态开辟空间，常用来产生动态数组及\_\_\_\_\_。
25. 指针使用成员有两种方法，“->”指向运算符和“.”\_\_\_\_\_。
26. 假如一个类的名称为 MyClass，使用这个类的一个对象初始化该类的另一个对象时，可以调用\_\_\_\_\_构造函数来完成此功能。
27. 一个类拥有多个构造函数，则这些构造函数之间为\_\_\_\_\_关系。
28. 在面向对象的程序设计中，将一组对象的共同特性抽象出来形成\_\_\_\_\_。
29. 对赋值运算符进行重载时，应声明为\_\_\_\_\_函数。
30. String 类的\_\_\_\_\_方法返回查找到的字符串在主串的位置。
31. 取子字符串“substr(5, 3);”第 1 个参数表示\_\_\_\_\_。
32. 派生类从基类保护继承时，基类的公有成员在派生类中改变为\_\_\_\_\_成员。
33. 复制构造函数的参数有些特别，是\_\_\_\_\_的对象。
34. 在类 City 的定义中加入语句“ friend void Street::f(); ”，是把\_\_\_\_\_类成员函数 f() 声明为了类 City 的友元函数。
35. 类模板用来表达具有\_\_\_\_\_的模板类对象集。
36. this 指针是隐藏的指针，它指向\_\_\_\_\_。
37. 假定一个结构类型定义为：“struct D { int a;union{int b;double c;} ;D \* d[2];};”，则该类型的大小为\_\_\_\_\_字节。
38. 若用数组名作为函数调用时的实参，则实际上传递给形参的是\_\_\_\_\_。
39. 内联函数在编译时是将该函数的\_\_\_\_\_替换其调用表达式。
40. 利用 C++编程语言可以建立一个通用函数，其函数类型和形参类型不具体指定，用一个虚拟的类型来代表。这个通用函数就称为\_\_\_\_\_。

三、改错题：本大题共 5 小题，每小题 4 分，共 20 分。以下程序中各有一处错误，请抄写有错误的语句并说明错误原因。

```
41. #include<iostream>
using namespace std;
class Student{
int Num=0,Score=0;
public:
    Student(int a,int b){Num=a;Score=b;}
    void get(){cout<<Num<<":"<<Score<<endl;}
};
void main(){
    Student student(1,96);
```

```

        student.get();
    }
}
42. #include<iostream>
using namespace std;
int m[3][3]={ {1}, {2}, {3} };n[3][3]={ 1, 2, 3 };
int main()
{   cout<<m[1][0]+n[0][0]<<'\n'<<m[0][1]+n[1][0]<<'\n';
    return 1;
}
43. #include<iostream>
using namespace std;
class A
{private:int m;
public:
    void A(int i=0) {
        m=i;cout<<"constructor called."<<m<<'\n';}
    void Set(int i) {m=i;}
    void Print() const {cout<<m<<endl;}
    ~A() {cout<< "destructor called."<<m<<'\n';}
};
void fun(const A&c){c.Print();}
void main()
{   fun(5); }
44. #include<iostream>
using namespace std;
class Count{
static int count ;
public:
    Count() {cout<<count++ ;}
    static int Getc() { return count ;}
    ~Count() {count--;}
};
int Count::count=5,obj;
int main()
{   cout<<obj.Getc()<<endl;
    return 0;
}

```



```

    virtual void disp() {cout<<"Person  ";}
};

-----

{public:
    void disp() {cout<<"Address"<<endl;}
};
class PhoneNumber:public Person
{ public:
    void disp() {cout<<"PhoneNumber"<<endl;}
};
void main()
{
    Person *p;
    Address a;
    PhoneNumber pn;
    p=&a;
    p->Person::disp();
    p->disp();
    p=&pn;

    -----

    p->disp();
}

```

48. 完成程序，使其运行结果如下：

欢迎使用测试程序！

共生成过 3 个对象。

下次再见！

```

#include <iostream>
using namespace std;
int time=0,end=0;
class Test
{ public:
    Test()
        { if(_____)cout<<"欢迎使用测试程序！ "<<endl;
          time=time+1;          }
    ~Test()
        {end=end+1;

```

```

        if(end==time)cout<<"下次再见! "<<endl;    }
        void print() {cout<<"共生成过"_____<<endl; }
};
void main()
{   Test obj1,obj2,obj3;
    obj1.print();}

```

49. #include <iostream.h>

```

class toy
{private:
    int num, price;
public:
    toy(int q, int p)
    {   num=q;
        price=p; }
    int get_num(){return num;}
_____
};

```

```

int main()
{   toy op[2][2]={toy(10,20),toy(30,48),toy(50,68),toy(70,80)};
    int i;
    for(i=0;i<2;i++)
        for(_____)
        {
            cout<<op[i][j].get_num()<<" ";
            cout<<op[i][j].get_price()<<"\n";
        }
    return 0;
}

```

50. 程序完成后的运行结果为: Welcome!

```

#include <iostream>
#include <string>
using namespace std;
class ChangeStr{
private:char * s;
public:
    void set(char *str){
        str=new char[strlen(s)+1];

```



```

        strcpy(str,s);}
ChangeStr& _____{
    delete str;
    set(a.str);
    return *this;}
void show(){cout<<str<<endl;}
ChangeStr(char *a){set(a);}
~ChangeStr(){delete str;}
};
int main()
{   ChangeStr s1("Welcome!"),s2("Hello!");
    _____;
    s2.show();
    return 1;
}

```

五、程序分析题：本大题共 2 小题，每小题 5 分，共 10 分。阅读程序后，填写程序的正确运行结果。

51. #include<iostream>  
using namespace std;  
class A {  
int a,b;  
public:  
A(){a=b=0;}  
A(int aa,int bb):a(aa),b(bb) {  
cout <<"a="<<a<<"\n"<<"b="<<b<<endl;  
}  
~A(){cout <<"Destructor!"<<endl; }  
};  
void main()  
{  
A x,y(2,3);  
}

52. #include <iostream>  
using namespace std;  
class A  
{ public:

```

    A(int i){x=i;}
    void dispa(){cout<<"A:"<<x<<endl;}
private:
    int x;
};
class B: public A
{ public:
    B(int i):A(i+10){x=i;}
    void dispb(){ dispa();cout<<"B:"<<x<<endl;}
private:
    int x;
};
int main()
{
    B b(10);
    b.dispb();
    return 0;
}

```

## 六、程序设计题：共 10 分。

53. 下面是一个类的测试程序，请设计出能使用如下测试程序的类。

```

int main()
{
    Test a;
    a.init(2019,100);
    a.print();
    return 0;
}

```

其执行结果为：

测试结果：2019-100=1949