

得 分

一、单项选择(每空 1 分，共 20 分)

(注意选项填写到下面表格中!)

1. 类 aClass 的定义如下：

```
class aClass
{public:
    aClass(){}
    aClass(int i){value=new int(i);}
    int *value;
}
```

若要对类中的 value 赋值，则下面的语句中，正确的是（ ）。

- A) aClass my; my.value=10;
- B) aClass my; *my.value=10;
- C) aClass my; my.*value=10;
- D) aClass my(10);

2. 如果类 A 被说明成类 B 的友元，则（ ）。

- A) 类 A 的成员函数不得访问类 B 的所有成员
- B) 类 A 的成员函数可以访问类 B 的所有成员
- C) 类 A 的成员即类 B 的成员
- D) 类 B 的成员即类 A 的成员

3. 类对象占用内存的大小与（ ）有关。

- A) 数据成员 B) 成员函数
- C) 数据成员和成员函数 D) 私有成员

4. 假定 MyClass 为一类，执行 MyClass a,b(2),*p[2];语句时会自动调用该类构造函数（ ）次。

- A) 2 B) 3 C) 4 D) 5

5. 下列关于成员函数特征的描述中，错误的是（ ）。

- A) 成员函数可以重载
- B) 成员函数一定是内联函数
- C) 成员函数可以设置参数的默认值
- D) 成员函数可以是静态的

6. 下列关于析构函数的说法错误的是（ ）。

- A) 析构函数有且仅有一个
- B) 析构函数无任何函数类型
- C) 析构函数和构造函数一样可以有参数
- D) 析构函数的作用之一是在对象被撤销时收回先前分配的内存空间

7. 下列虚基类的声明中，正确的是（ ）。

- A) `class virtual B:public A`
- B) `virtual class B:public A`
- C) `class B: public A virtual`
- D) `class B: virtual public A`

8. 基类的（ ）在私有继承时在派生类中为私有成员函数，在公有和保护继承时在派生类中为保护成员函数。

- A) 私有成员函数
- B) 保护成员函数
- C) 公有成员
- D) 保护数据成员

9. 多继承的构造函数顺序可分为如下 4 步：

①所有非虚基类的构造函数按照它们被继承的顺序构造。

②所有虚基类的构造函数按照它们被继承的顺序构造。

③所有子对象的构造函数按照它们声明的顺序构造。

④派生类自己的构造函数体

这四个步骤的正确顺序是（ ）。

- A) ④③①②
- B) ②④③①
- C) ②①③④
- D) ③④①②

10. 在派生类中重新定义虚函数时，除了（ ），其他方面都应基类中相应的虚函数保持一致。

A) 函数名 B) 参数 C) 函数体 D) 返回值类型

11. 关于纯虚函数和抽象类的描述中，错误的是（ ）。

A) 纯虚函数是一种特殊的虚函数，它没有具体的实现。

B) 抽象类是指具有纯虚函数的类。

C) 一个基类中说明有纯虚函数，该类的派生类一定不再是抽象类。

D) 抽象类作为基类来使用，其纯虚函数的实现由派生类给出。

12. 虚函数必须是类的（ ）。

A) 友元函数 B) 成员函数 C) 构造函数 D) 析构函数

13. 友元运算符 `obj1>obj2` 被 C++ 编译器解释为（ ）。

A) `operator>(obj1,obj2)` B) `>(obj1,obj2)`

C) `obj1.operator>(obj2)` D) `obj2.operator>(obj1)`

14. 下列运算符中，（ ）在 C++ 中不能重载。

A) `?:` B) `+` C) `-` D) `<=`

15. 下列关于运算符重载正确的是（ ）。

A) 运算符重载可以改变运算符的操作数个数

B) 运算符重载可以改变运算符的优先级

C) 运算符重载可以改变运算符的结合性

D) 运算符重载不可以改变语法结构

16. 使用 `fstream` 流类定义流对象并打开磁盘文件时，文件的隐含打开方式是（ ）。

A) `ios::in` B) `ios::out` C) `ios::in|ios::out` D) 没有默认，必须指定打开方式

17. 下列语句不能用于打开 C 盘根目录下文件 `test.txt` 的语句是（ ）。

A) `ifstream fin; fin.open("c:\\test.txt");`

B) `ifstream fin("c:\\test.txt");`

C) A 和 B

D) `ifstream fin; fin("c:\\test.txt");`

18. 下列输出字符 'A' 的方法中，（ ）是错误的。

A) `cout<<put('A');` B) `cout<<'A';`

C) `cout.put('A');` D) `char A='A'; cout<<A;`

19. 类模板的类型参数（ ）。

- A) 只可作为数据成员的类型
- B) 只可作为成员函数的返回类型
- C) 只可作为成员函数的参数类型
- D) 以上三者皆可

20. () 是一个通用函数的描述，它可以对不同类型的数据完成同一个任务。

- A) 函数模板 B) 重载函数 C) 递归函数 D) 模板函数

得 分

二、判断题（每空 1 分，共 10 分）

注意：请将判断题结果（正确为 **T**，错误为 **F**）

写在下面相应位置上：

① ② ③ ④ ⑤

⑥ ⑦ ⑧ ⑨ ⑩

1. 当某个对象调用类的成员函数时，该对象的地址自动传递给 **this** 指针。
2. 在一个类中声明了某个静态数据成员，并在类外给出其定义和初始化后，若未定义该类的对象，该静态数据成员则不存在。
3. 构造函数不能被派生类继承。
4. 派生类至少有一个基类。
5. C++ 语言类定义中默认访问权限是 **public**。
6. 对虚函数的调用，系统都是采用动态联编。
7. 假定要对类 **AB** 定义加号操作符重载成员函数，实现两个 **AB** 类对象的加法，并返回相加结果，则该成员函数的声明语句为：

AB operator+(AB &,AB &);

8. **cin** 是 **istream** 的一个对象，处理标准输入。
9. **getline()** 函数只能用于从键盘上读取字符串。
10. **template <class T1,T2>** 是正确的模板说明。

得 分

三、读程序写结果（每空 2 分，共 38 分）

(注意将结果写在下面相应横线上!)

(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11)_ (12)

(13) (14) (15) (16)

(17) (18) (19)

1. 写出下面程序的运行结果。

```
#include<iostream.h>

class Sample
{ char ch;
public:
Sample() {ch='a';cout<<ch<<endl;}
Sample(char character) {ch=character;cout<<ch<<endl;}
~Sample()
{ cout<<"destruct"<<ch<<endl; }
};

void main()
{ Sample *first=new Sample;
Sample second('b') ,third('c');
delete first;
}
```

运行结果：

(1)

(2)

(3)

(4)

(5)

(6)

2. 写出下面程序的运行结果。

```
#include<iostream.h>

class Table
{
public:
    Table(){cout<<"table"<<endl;}
    ~Table(){cout<<"table"<<endl;}
};

class Circle
{
public:
    Circle(){cout<<"circle"<<endl;}
    ~Circle(){cout<<"circle"<<endl;}
};

class RoundTable:public Circle,public Table
{
public:
    RoundTable():Table(),Circle()
    { cout<<"roundtable"<<endl; }
    ~RoundTable(){cout<<"roundtable"<<endl;}
};

void main()
```

```
{  
RoundTable rt;  
}
```

运行结果：

(7)

(8)

(9)

(10)

(11)

(12)

3. 写出下面程序的运行结果。

```
#include<iostream.h>  
  
class Instrument  
{  
public:  
virtual void play(){cout<<"play instrument"<<endl;}  
void display(){cout<<"instrument"<<endl;}  
};  
  
class Wind:public Instrument  
{  
public:  
void play(){cout<<"play wind"<<endl;}  
void display(){cout<<"wind"<<endl;}  
};  
  
class Stringed:public Instrument
```

```

{
public:
void play(){cout<<"play stringed"<<endl;}
void display(){cout<<"stringed"<<endl;}
};

void main()
{
Wind w;
Stringed s;
Instrument *p=&w;
p->play();
w.play();
p=&s;
p->display();
s.display();
w.display();
}

```

运行结果：

(13)

(14)

(15)

(16)

(17)

4. 读程序，按要求写出结果。


```

#include<fstream.h>
#include<stdlib.h>
void main()
{
ofstream outfile1("file1.txt");
ofstream outfile2("file2.txt");
if(!outfile1||!outfile2)
{
cout<<"file can't open"<<endl;
abort();
}
char str1[]="abcd";
outfile1<<str1;
char str2[]="efg";
outfile2<<str2;
outfile1.close();
outfile2.close();
ifstream infile("file2.txt");
ofstream outfile("file1.txt",ios::app);
char ch;
while(infile.get(ch))
outfile.put(ch);
infile.close();
outfile.close();
}

```

程序运行后，

file1.txt 文件内容为： (18)

file2.txt 文件内容为： (19)

得 分

四、程序填空（每空 2 分，共 32 分）

(注意将结果写在下面相应横线上!)

(1) (2) (3) (4)

(5) (6)

(7) (8)

(9) (10) (11) (12)

(13) (14) (15) (16)

1. 请将下面程序补充完整。

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

```
#include<string.h>
```

```
class Student
```

```
{public:
```

```
void (1) ( (2) )
```

```
{ strcpy(name,s); score=x; sum=sum+score; }
```

```
(3) double GetSum(){return sum;}
```

```
private:
```

```
char name[20];
```

```
double score;
```

```
static double sum;
```

```
};
```

```
(4) ; //静态数据成员初始化为 0
```

```

void main()
{ Student s1,s2;
s1.Setdate("zhang",99);
s2.Setdate("li",88);
cout<<Student::GetSum()<<endl;
}

```

2. 下面程序输出为:

zhang

计算机

99

wang

请将程序补充完整。

```

#include<iostream.h>
#include<string.h>
class Person
{ char name[20];
public:
Person(char*n)
{ strcpy(name,n); }
void display()
{ cout<<name<<endl; }
};
class Teacher: (5)
{protected:
char course[20];
public:
Teacher(char *n,char*c):Person(n)
{ strcpy(course,c); }
};

```

```

class Student: (6)
{ protected:
double score;
public:
Student(char *n,double s):Person(n)
{ score=s; }
};

class GraduateStudent:public Student,public Teacher
{ char advisor[20];
public:
GraduateStudent(char *n,char*c,double s,char *a): (7)
{ strcpy(advisor,a); }
void display()
{
(8) ;

cout<<course<<endl<<score<<endl<<advisor<<endl;
}
};

void main()
{ GraduateStudent gs("zhang","计算机",99,"wang");
gs.display ();
}

```

3. 请将下面程序补充完整。

```

#include<iostream.h>

class Point
{
double x,y;
public:

```

(9) (double a,double b)

{

x=a;

y=b;

}

(10) int operator==(Point&,Point&);

Point& operator+=(Point&);

void display()

{ cout<<x<<" "<<y<<endl; }

};

Point& (11) (Point& c)

{

x=x+c.x;

y=y+c.y;

return (12) ;

}

int operator==(Point& a,Point& b)

{

if(a.x==b.x&& a.y==b.y)

return 1;

else

return 0;

}

void main()

{

Point p1(3,4),p2(4,5);

if(p1==p2)

cout<<"相等"<<endl;

```

else
cout<<"不相等"<<endl;

p1+=p2;

p1.display();

p2.display();

}

```

4. 下面程序的功能是：先将 3 条学生记录写入文件，然后查找姓名为李四的学生记录，从文件中将其读出并输出到屏幕上，请将程序补充完整。

```

#include<fstream.h>

#include<stdlib.h>

#include<string.h>

class Student
{
char name[20];

double score;

public:

Student(char*n="",double s=0)
{ strcpy(name,n);score=s; }

void input()
{ cin>>name>>score; }

void output()
{ cout<<name<<endl<<score<<endl; }

char* getname()
{ return name; }

};

void main()
{

(13)  ifstream;

ifstream("file.dat", ios::in|ios::out|ios::binary);

```

```
if(!iofile)
{
cout<<" file.dat can't open"<<endl;
abort();
}

Student s[3]={Student("张三",99),Student("李四",88),Student("王五",77)};

iofile.write((char *)s, (14) );

iofile.seekg(0, (15) );

Student stu;

while(iofile.read( (16) ,sizeof(Student)))
{
if(strcmp(stu.getname(),"李四")==0)
stu.output();
}

iofile.close();
}
```