

# 第 1 讲 C++语言概述

## 第 1 讲了解计算机的基本工作原理

### 1、单选题：

以下都可用作计算机输入设备的是：

选项：

- A: 键盘，鼠标，扫描仪，打印机
- B: 键盘，数码相机，鼠标，绘图仪
- C: 键盘，数码相机，扫描仪，绘图仪
- D: 键盘，鼠标，扫描仪，数码相机

答案：【 键盘，鼠标，扫描仪，数码相机 】

### 2、单选题：

CPU 中包含控制器和：

选项：

- A: 运算器
- B: 存储器
- C: 输入设备
- D: 输出设备

答案：【 运算器 】

### 3、单选题：

下列语言不属于高级语言中的是：

选项：

- A: C 语言
- B: 机器语言
- C: FORTRAN 语言
- D: C++语言

答案：【 机器语言 】

### 4、单选题：

正确的 C++语言标识符是：

选项：

- A: 3d\_max
- B: if
- C: A&B
- D: sum\_2

答案：【 sum\_2 】

### 5、单选题：

对 C++语言和 C 语言的兼容性，描述正确的是：

选项：

- A: C++兼容 C  
B: C++部分兼容 C  
C: C++不兼容 C  
D: C 兼容 C++  
答案: 【C++兼容 C】

## 第 2 讲信息的表示与存储

### 第 2 讲了解各种进位计数制及常用数据在计算机中的存储

1、单选题:

下列数中最大的数为:

选项:

- A: (101001)B  
B: (52)O  
C: (43)D  
D: (2C)H

答案: 【(2C)H】

2、单选题:

如果 X 为负数, 由[X]补求[-X]补是将:

选项:

- A: [X]补各值保持不变  
B: [X]补符号位变反, 其它各位不变  
C: [X]补除符号位外, 各位变反, 末位加 1  
D: [X]补连同符号位一起各位变反, 末位加 1

答案: 【[X]补连同符号位一起各位变反, 末位加 1】

3、单选题:

以下叙述错误的是:

选项:

- A: 浮点数中, 阶码反映了小数点的位置  
B: 浮点数中, 阶码的位数越长, 能表示的精度越高  
C: 计算机中, 整数一般用定点数表示  
D: 浮点数和整数的最高位都是符号位

答案: 【浮点数中, 阶码的位数越长, 能表示的精度越高】

4、单选题:

二进制数 101101.101 转换为十进制数、八进制数和十六进制数的结果分别是:

选项:

- A: 45.525、55.5、2D.A  
B: 45.625、55.5、2D.B  
C: 45.625、55.5、2D.A

D: 45.625、55.6、2D.A

答案: 【 45.625、55.5、2D.A 】

5、单选题:

字符型数据'A'存放在计算机中:

选项:

A: 存放的就是字母'A'

B: 存放的是'A'的 ASCII 码值的补码

C: 实际存放的是'A'的 ASCII 码值 97

D: 实际存放的是'A'的 ASCII 码值 65

答案: 【 实际存放的是'A'的 ASCII 码值 65 】

## 第 3 讲程序中数据的表示

### 第 3 讲掌握常量、变量的表示及基本数据类型

1、单选题:

类型修饰符 unsigned 不能修饰

选项:

A: char

B: int

C: long int

D: float

答案: 【 float 】

2、单选题:

在 C++语言的数据类型中, int, short 等类型的长度是

选项:

A: 固定的

B: 任意的

C: 用户自定义的

D: 与机器字长有关的

答案: 【 与机器字长有关的 】

3、单选题:

下列选项中, 均是合法的整型常量的是:

选项:

A: 60-0xffff0011

B: -0xcdf01a0xe

C: -01986,0120668

D: -0x48a2e50x

答案: 【 60-0xffff0011 】

4、单选题：

下列选项中，均是合法的实型常量的是：

选项：

A: +1e+15e-9.403e2

B: -0.1012e-4-8e5

C: 123e1.2e-.4+2e-1

D: -e3.8e-45.e-0

答案：【 -0.1012e-4-8e5 】

5、单选题：

下列字符串常量表示中，哪个是错误的？

选项：

A: ""yes"or"No""

B: "OK!"

C: "abcdn"

D: "ABC"

答案：【 ""yes"or"No"" 】

6、单选题：

字符串"tv\whichn"的长度是：

选项：

A: 4

B: 3

C: 9

D: 字符串有非法字符，输出值不确定

答案：【 3 】

7、单选题：

以下不是 C++ 语言支持的存储类别的是：

选项：

A: auto

B: static

C: dynamic

D: register

答案：【 dynamic 】

8、单选题：

下列不是 C++ 语言的合法用户标识符的是：

选项：

A: a#b

B: \_int

C: a\_10

D: Pad

答案：【 a#b 】

9、单选题：

下列字符列中，可作为 C++ 语言程序自定义标识符是：

选项：

A: switch

B: file

C: break

D: do

答案：【 file 】

10、单选题：

下列属于逻辑运算的一组算式是：

选项：

A: 1/1=1

B: 1-1=0

C: 1+1=10

D: 1+1=1

答案：【 1+1=1 】

## 第 4 讲运算符与表达式

### 第 4 讲掌握各种表达式的表示

1、单选题：

命题“ $10 < m < 15$  或  $m > 20$ ”的 C++ 语言表达式是

选项：

A:  $((m > 10) \&\&(m < 15) || (m > 20))$

B:  $((m > 20) \&\&(m < 15) || (m > 10))$

C:  $(m > 10) || ((m < 15) \&\&(m > 20))$

D:  $((m > 10) || (m < 15) || (m > 20))$

答案：【  $((m > 10) \&\&(m < 15) || (m > 20))$  】

2、单选题：

设  $\text{int } a=3, b=4, c=5$ ; 表达式  $(a+b) > c \&\& b == c$  的值是：

选项：

A: 2

B: -1

C: 0

D: 1

答案：【 0 】

3、单选题：

若  $x$  是一个 `bool` 型变量， $y$  是一个值为 100 的 `int` 型变量，则表达式  $!x \&\& y > 0$  的值为：

选项:

A: 为 true

B: 为 false

C: 与 x 的值相同

D: 与 x 的值相反

答案: 【与 x 的值相反】

4、单选题:

设变量 m, n, a, b, c, d 均为 0, 执行(m = a==b)&&(n=c==d)后, m, n 的值是:

选项:

A: 0,0

B: 0,1

C: 1,0

D: 1,1

答案: 【1,1】

5、单选题:

设 a 和 b 均为 double 型变量, 且 a=5.5, b=2.5, 则表达式 (int) a+b/b 的值是:

选项:

A: 6.500000

B: 6

C: 5.500000

D: 6.000000

答案: 【6.000000】

6、单选题:

以下非法的赋值表达式是:

选项:

A: n=(i=2,i++)

B: j++

C: ++(i+1)

D: x=j>0

答案: 【++(i+1)】

7、单选题:

设 int x=2, y=4, z=7; 则执行 x=y--<=x || x+y!=z 后 x, y 的值分别为:

选项:

A: 0,3

B: 1,3

C: 2,3

D: 2,4

答案: 【1,3】

8、单选题：

表达式!x 等效于：

选项：

A: x==1

B: x==0

C: x!=1

D: x!=0

答案：【 x==0 】

9、单选题：

设以下变量均为 int 类型，则值不等于 7 的表达式是：

选项：

A: (x=y=6,x+y,x+1)

B: (x=y=6,x+y,y+1)

C: (x=6,x+1,y=6,x+y)

D: (y=6,y+1,x=y,x+1)

答案：【 (x=6,x+1,y=6,x+y) 】

10、单选题：

在下列成对的表达式中，运算符“+”的意义不相同的一对是：

选项：

A: 5.0+2.0 和 5.0+2

B: 5.0+2.0 和 5+2.0

C: 5.0+2.0 和 5+2

D: 5+2.0 和 5.0+2

答案：【 5.0+2.0 和 5+2 】

## 第 5 讲顺序结构的程序设计

### 第 5 讲掌握 C++输入输出的格式控制

1、单选题：

输入输出格式控制是在哪个头文件中定义的？

选项：

A: iostream.h

B: iomanip.h

C: istream.h

D: ostream.h

答案：【 iomanip.h 】

2、单选题：

对于语句 cout<<endl<<x;中的各个组成部分，下列叙述中错误的是：

选项：

A: "cout"是一个输出流对象  
B: "endl"的作用是输出回车换行  
C: "x"是一个变量  
D: "<<"称作提取运算符  
答案: 【 "<<"称作提取运算符 】

3、单选题:

在 ios 中提供控制格式的标志位中, 哪个是转换为十六进制形式的标志位?

选项:

A: hex

B: oct

C: dec

D: left

答案: 【 hex 】

4、单选题:

定义变量: char one\_char;则语句 cout<<one\_char;显示结果相当于 C 语言中的:

选项:

A: printf(one\_char);

B: printf("%c",one\_char);

C: scanf(one\_char);

D: scanf("%c",&one\_char);

答案: 【 printf("%c",one\_char); 】

5、单选题:

若有定义"int x=17;", 则语句"cout<<oct<<x;"的输出结果是:

选项:

A: 11

B: 0x11

C: 21

D: 021

答案: 【 21 】

6、单选题:

与 C 语言 printf("Hello,World \n"); 语句功能相同的 C++语句是:

选项:

A: cout>>"Hello,World \n";

B: cout<<"Hello,World \n";

C: cin>>"Hello,World \n";

D: cin<<"Hello,World \n";

答案: 【 cout<<"Hello,World \n"; 】

7、单选题:

与语句 cout<<endl; 不等价的是:

选项:



A: `cout<<' n';`

B: `cout<<'12';`

C: `cout<<' xA';`

D: `cout<<' '`

答案: 【 `D) cout<<' '` 】

8、单选题:

下列程序的运行结果是:`#include <iostream>using namespace std;int main(){int a=2;int b=a+1;cout<<a/b<<endl;return 0;}`

选项:

A: 0.66667

B: 0

C: 0.7

D: 0.6666666...

答案: 【 0 】

9、单选题:

执行下列代码, 程序的输出结果是(用下划线表示空格): `int a=29, b=100; cout <<setw(3) << a << b << endl;`

选项:

A: 29\_100

B: \_29\_100

C: 29100

D: \_29100

答案: 【 \_29100 】

10、单选题:

执行下列代码, 程序的输出结果是: `cout <<"Hex: " << hex << 255;`

选项:

A: ff

B: hex:ff

C: Hex:ff

D: f

答案: 【 Hex:ff 】

## 第 6 讲选择结构的程序设计

### 第 6 讲掌握选择结构的程序设计

1、单选题:

流程控制语句的基本控制结构有三种, 不属于这一种结构的是:

选项:

A: 顺序结构

B: 选择结构

C: 循环结构

D: 计算结构

答案: 【 计算结构】

2、单选题:

在设计程序时, 应采纳的原则之一是:

选项:

A: 不限制 goto 语句的使用

B: 减少或取消注解行

C: 程序越短越好

D: 程序结构应有助于读者理解

答案: 【 程序结构应有助于读者理解】

3、单选题:

if 语句的语法格式可描述为: 格式 1: if(<条件>)<语句>或 if(<条件>)<语句 1>else <语句 2>关于上面的语法格式, 下列表述中错误的是:

选项:

A: <条件>部分可以是一个 if 语句, 例如 if(if(a==0).....).....

B: <语句>部分可以是一个 if 语句, 例如 if(.....)if (.....) .....

C: 如果在<条件>前加上逻辑非运算符! 并交换<语句 1>和<语句 2>的位置, 语句功能不变

D: <语句>部分可以是一个循环语句, 例如 if(.....) while (.....) .....

答案: 【 <条件>部分可以是一个 if 语句, 例如 if(if(a==0).....).....】

4、单选题:

在 if 语句中的表达式是:

选项:

A: 只能是表达式

B: 只能是关系表达式和逻辑表达式

C: 只能是逻辑表达式

D: 可以是任意表达式

答案: 【 可以是任意表达式】

5、单选题:

执行如下语句序列, 不可能出现的情况是: int x; cin>>x; if(x>250) cout<<'A'; if(x<250) cout<<'B'; else cout<<'A';

选项:

A: 显示: A

B: 显示: B

C: 显示: AB

D: 显示: AA

答案: 【 显示: AB】

6、单选题：

阅读下面的程序，如果两次执行上述程序，且键盘输入分别是 4 和 6，则输出结果是：  
`#include <iostream>using namespace std;int main(){int x;cin>>x;if (x++>5) cout <<x<<endl;elsecout<<x--<<endl;return 0;}`

选项：

A: 4,6

B: 3,6

C: 4,7

D: 5,7

答案：【 5,7】

7、单选题：

下列描述正确的是：

选项：

A: 表示  $m > n$  为 false 或  $m < n$  为 true 的表达式为  $(m > n \&\& m < n)$

B: switch 语句结构中必须有 default 语句

C: if 语句结构中必须有 default 语句

D: 如果至少有一个操作数为 true，则包含 || 运算符的表达式为 true

答案：【 如果至少有一个操作数为 true，则包含 || 运算符的表达式为 true】

8、单选题：

执行如下语句序列，不可能出现的情况是：

```
int n; cin>>n; switch(n){ case 1: case 2: cout<<'A'; case 3: case 4: cout<<'B'; break; default:cout<<'C'; }
```

选项：

A: 显示： A

B: 显示： B

C: 显示： C

D: 显示： AB

答案：【 显示： A】

9、单选题：

关于 switch 语句的说明中，错误的是：

选项：

A: default 语句是可缺省的

B: 各个分支中的 break 语句起着退出 switch 语句的作用

C: switch 结构不可以嵌套

D: 每个 case 语句中不必用 { }，而整体的 switch 结构一定要写一对花括号 { }

答案：【 switch 结构不可以嵌套】

10、单选题：

将以下程序写成三目运算表达式语句是： `if (a>b) max=a;else max=b;`

选项：

A: `max=(a>b)?;`

B: `(max=a>b)?a:b;`

C: `c=(a>b)?a:b;`

D: `max=(a>b)?a:b;`

答案: 【 `max=(a>b)?a:b;` 】

## 第 7 讲循环结构的程序设计

### 第 7 讲掌握循环结构的程序设计

1、单选题:

`while(!x)`中的`(!x)`与下面哪个条件等价?

选项:

A: `x==1`

B: `x!=1`

C: `x!=0`

D: `x==0`

答案: 【 `x==0` 】

2、单选题:

已知: `int i=5`, 下列 `do-while` 循环语句的循环次数为: `do{cout<<i--<<endl;i--;}while(i!=0);`

选项:

A: 0

B: 1

C: 5

D: 无限

答案: 【 无限 】

3、单选题:

下面程序段: `x=3; do{y=x--;if(!y) {cout<<"x"; continue;}cout<<"#";}while(x>=1 && x<=2);`

选项:

A: 将输出##

B: 是死循环

C: 将输出###

D: 含有不合法的控制表达式

答案: 【 将输出### 】

4、单选题:

`for(int x=0,y=0;!x&& y<=5;y++)`语句执行循环的次数是:

选项:

A: 0

B: 5

C: 6

D: 无限

答案: 【 6 】

5、单选题:

以下程序的输出结果是: `#include <iostream>using namespace std;int main(){char s[]="abcdefbcde",*p=s;int v1=0,v2=0,v3=0,v4=0; for (p;*p;p++) switch(*p){ case 'a':v1++;break; case 'b':v3++;break; case 'e':v2++;break; default: v4++;} cout<<v1<<" "<<v2<<" "<<v3<<" "<<v4<<endl;return 0;}`

选项:

A: 1,2,2,5

B: 4,7,5,8

C: 1,5,3,10

D: 8,8,8,8

答案: 【 1,2,2,5 】

6、单选题:

下面有关 for 循环的正确描述是:

选项:

A: for 循环只能用于循环次数已经确定的情况

B: for 循环是先执行循环体语句, 后判断表达式

C: 在 for 循环中, 不能用 break 语句跳出循环体

D: for 循环的循环体语句中, 可以包含多条语句, 但必须用大括号括起来

答案: 【 for 循环的循环体语句中, 可以包含多条语句, 但必须用大括号括起来 】

7、单选题:

C++语言的跳转语句中, 对于 break 和 continue 说法正确的是:

选项:

A: break 语句只应用与循环体中

B: continue 语句只应用与循环体中

C: break 是无条件跳转语句, continue 不是

D: break 和 continue 的跳转范围不够明确, 容易产生问题

答案: 【 continue 语句只应用与循环体中 】

8、单选题:

下列循环语句的循环次数是: `while (int i=0) i--;`

选项:

A: 0

B: 1

C: 2

D: 3

答案: 【 0 】

9、单选题:

下面程序的输出结果是: `#include<iostream>using namespace std;int main(){for(int i=-1;i<4;i++)cout<<(i?'0':'*');return 0;}`

选项:

A: 0\*0

B: 0000

C: 0\*00

D: 0\*000

答案: 【 0\*000 】

10、单选题:

下面程序的输出结果是: #include "iostream"using namespace std;int main() {int i=17;while(i>=10)if(--i%4==3) continue;elsecout<<"i="<<i--<<endl;return 0;}

选项:

A: i=16i=15i=14i=13

B: i=16i=14i=12

C: i=16i=14i=12i=10

D: i=17i=15i=13i=11

答案: 【 i=16i=14i=12i=10 】

## 第 8 讲循环结构的设计

### 第 8 讲掌握循环结构的程序设计

1、填空题:

程序填空完成功能: 求出 1000 以内的全部素数。#include<iostream>using namespace std;int main(){ const int m=1000; int i,j,isprime; for (i=2;i<1000;i++){ isprime=1; for (j=i-1;j>1;j--) if (i%j==0) 【 】 ; if (isprime) cout<<i<<','; } return 0;}

答案: 【 isprime=0 】

2、填空题:

程序填空完成功能: 求分数序列 2/1,3/2,5/3,8/5,13/8 .....的前 20 项之和。#include <iostream>using namespace std;int main(){ double i,n=1,m=1,t,s=0 ; for (i=1;i<=20;i++){ t = n ; n = m ; 【 】 ; s = s + m/n ; } cout<<s<<endl; return 0;}

答案:

【 m=t+m##\_YZPRLFH\_###m=m+t##\_YZPRLFH\_###m=t+n##\_YZPRLFH\_###m =n+t 】

3、填空题:

程序填空完成功能: 有一堆机器零件 (零件个数小于 100), 3 个为一组则剩下 2 个, 5 个为一组则剩下 1 个, 7 个为一组则剩下 6 个, 求这堆零件一共有多少个? #include <iostream>using namespace std;int main(){ int i; for (i=1;i<100;i++){ if( 【 】 ) cout<<i<<endl; } return 0;}

答案:

【 i%3==2&& i%5==1&& i%7==6##\_YZPRLFH\_###(i%3==2)&&(i%5==1)&&(i%7==6)##\_YZPRLFH\_###(i%3==2&& i%5==1&& i%7==6)】

## 第 9 讲函数的定义和使用

### 第 9 讲掌握函数的定义和函数的调用方式

#### 1、单选题：

下列特性中，C 与 C++共有的是：

选项：

- A: 继承
- B: 封装
- C: 多态性
- D: 函数定义不能嵌套

答案：【函数定义不能嵌套】

#### 2、单选题：

必须用一对大括号括起来的程序段是：

选项：

- A: switch 语句中的 case 标号语句
- B: if 语句的分支
- C: 循环语句的循环体
- D: 函数的函数体

答案：【函数的函数体】

#### 3、单选题：

关于 C++主函数特性，下列叙述正确的是：

选项：

- A: 主函数在同一个 C++文件中可以有两个
- B: 主函数类型必须是 void 类型
- C: 主函数一定有返回值
- D: 每个 C++程序都必须有一个 main()函数

答案：【每个 C++程序都必须有一个 main()函数】

#### 4、单选题：

在 C++语言中，main 函数默认返回一个（ ）类型的值

选项：

- A: int
- B: float
- C: char
- D: void

答案：【int】

5、单选题：

C++语言中规定函数的返回值的类型是由：

选项：

- A: return 语句中的表达式类型所决定
- B: 调用该函数时的主调用函数类型所决定
- C: 调用该函数时系统临时决定
- D: 在定义该函数时多指定的数据类型所决定

答案：【在定义该函数时多指定的数据类型所决定】

6、单选题：

若调用一个函数，且此函数中没有 return 语句，则正确的说法是该函数：

选项：

- A: 没有返回值
- B: 返回若干个系统默认值
- C: 有返回值，但返回一个不确定的值
- D: 回一个用户所希望的函数值

答案：【有返回值，但返回一个不确定的值】

7、单选题：

下列叙述中错误的是：

选项：

- A: 一个函数中可以有多条 return 语句
- B: 调用函数必须在一条独立的语句中完成
- C: 函数中通过 return 语句传递函数值
- D: 主函数名 main 也可以带有形参

答案：【调用函数必须在一条独立的语句中完成】

8、单选题：

在 C++中把不返回任何类型的函数应该说明为：

选项：

- A: int
- B: float
- C: char
- D: void

答案：【void】

9、单选题：

在函数的定义格式中，下面各组成部分中，（ ）是可以省略的。

选项：

- A: 函数名
- B: 函数数据类型说明
- C: 函数体
- D: 函数参数

答案：【函数参数】



10、单选题：

函数调用 `func((exp1,exp2),(exp3,exp4,exp5))` 中所含实参的个数为：

选项：

A: 1

B: 2

C: 3

D: 4

答案：【2】

## 第 10 讲函数的设计

### 第 10 讲灵活运用函数重载、默认参数和函数模板

1、单选题：

指出下列对定义重载函数的要求中，（ ）是错误的。

选项：

A: 要求参数的个数相同

B: 要求参数的类型相同时，参数个数不同

C: 函数的返回值可以不同

D: 要求参数的个数相同时，参数类型不同

答案：【要求参数的个数相同】

2、单选题：

一个函数为 `void x(int, char ch='a')`，另一个函数为 `void x(int)`，则它们：

选项：

A: 不能在同一程序中定义

B: 可以在同一程序中定义并可重载

C: 可以在同一程序中定义，但不可以重载

D: 以上说法均不正确

答案：【可以在同一程序中定义，但不可以重载】

3、单选题：

重载函数在调用时选择的依据中，错误的是：

选项：

A: 函数的参数

B: 参数的类型

C: 函数的名字

D: 函数返回值类型

答案：【函数返回值类型】

4、单选题：

下列函数原型声明中错误的是：

选项：

A: void fun (int x=0,int y=0) ;

B: void fun(int x,int y);

C: void fun(int x,int y=0);

D: void fun(int x=0,int y);

答案: 【 void fun(int x=0,int y); 】

5、单选题:

一个函数带有参数说明时, 则参数的默认值应该在 ( ) 中给出。

选项:

A: 函数定义

B: 函数声明

C: 函数定义或声明

D: 函数调用

答案: 【 函数定义或声明 】

6、单选题:

下列关于函数参数默认值的描述中, 正确的是:

选项:

A: 函数参数的默认值只能设置一个

B: 若一个函数含有多个参数, 其中一个参数设置成默认值后, 其后所有参数都必须设置默认值

C: 若一个函数含有多个参数, 则设置默认参数时可以不连续设置默认值

D: C++语言中函数都必须设有默认值

答案: 【 若一个函数含有多个参数, 其中一个参数设置成默认值后, 其后所有参数都必须设置默认值 】

7、单选题:

模板函数的真正代码是在 ( ) 时候产生。

选项:

A: 源程序中声明函数时

B: 源程序中定义函数时

C: 源程序中的调用函数时

D: 运行执行函数时

答案: 【 源程序中的调用函数时 】

8、单选题:

以下关于函数模板叙述正确的是:

选项:

A: 函数模板也是一个具体类型的函数

B: 函数模板的类型参数与函数的参数是同一个概念

C: 通过使用不同的类型参数, 函数模板可以生成不同类型的函数

D: 用函数模板定义的函数没有类型

答案: 【 通过使用不同的类型参数, 函数模板可以生成不同类型的函数 】

9、单选题：

常情况下，函数模板中的类型参数个数不能是：

选项：

A: 0

B: 1

C: 2

D: 3

答案：【 0 】

10、单选题：

实现两个相同类型数加法的函数模板的声明是：

选项：

A: add(T x,T y)

B: T add(x,y)

C: T add(T x,y)

D: T add(T x,T y)

答案：【 T add(T x,T y) 】

## 第 11 讲函数的调用

### 第 11 讲掌握函数的嵌套调用和递归调用

1、单选题：

在参数传递过程中，对形参和实参的要求是：

选项：

A: 函数定义时，形参一直占用存储空间

B: 实参可以是常量、变量或表达式

C: 形参可以是常量、变量或表达式

D: 形参和实参类型和个数都可以不同

答案：【 实参可以是常量、变量或表达式 】

2、单选题：

在函数的引用调用时，实参和形参应该是使用：

选项：

A: 变量值和变量

B: 地址值和指针

C: 地址值和引用

D: 变量值和引用

答案：【 变量值和引用 】

3、单选题：

使用值传递方式将实参传给形参，下列说法正确的是：

选项：

- A: 形参是实参的备份
- B: 实参是形参的备份
- C: 形参和实参是同一对象
- D: 形参和实参无联系

答案: 【形参是实参的备份】

4、单选题:

下列函数调用中对调用它的函数没有起到任何作用的是:

选项:

- A: `void f1(double &x){--x;}`
- B: `double f2(double x){return x-1.5;}`
- C: `void f3(double x){--x;}`
- D: `double f4(double *x){--*x;return *x;}`

答案: 【`void f3(double x){--x;}`】

5、单选题:

对于某个函数调用, 不给出调用函数的声明的情况是:

选项:

- A: 被调用函数是无参函数
- B: 被调用函数是无返回值的函数
- C: 函数的定义在调用处之前
- D: 函数的定义有别的程序文件中

答案: 【函数的定义在调用处之前】

6、单选题:

在哪种情况下适宜采用 `inline` 定义内联函数?

选项:

- A: 函数体含有循环语句
- B: 函数体含有递归语句
- C: 函数代码少、频繁调用
- D: 函数代码多、不常调用

答案: 【函数代码少、频繁调用】

7、单选题:

下列哪个类型函数不适合声明为内联函数:

选项:

- A: 函数体语句较多
- B: 函数体语句较少
- C: 函数执行时间较短
- D: 函数执行时间过长

答案: 【函数体语句较多】

8、单选题:

在 C++ 中, 编写一个内联函数 `Fsqu`, 使用 `double` 类型的参数, 求其平方并返回, 返回值为 `double` 类型, 下列定义正确的是:

选项:

A: `double Fsqu (double x){return x*x;}`

B: `inline double Fsqu (double x){return x*x;}`

C: `double inline Fsqu (double x){return x*x;}`

D: `double Fsqu (double x){inline return x*x;}`

答案: 【`inline double Fsqu (double x){return x*x;}`】

9、单选题:

已知递归函数f的定义如下:`int f(int n){ if (n<=1) return 1; else return n*f(n-2);}`则函数调用语句f(5)的返回值是:

选项:

A: 14

B: 15

C: 16

D: 17

答案: 【15】

10、单选题:

在一个函数体中直接或间接地调用函数本身, 这种调用称为( )调用。

选项:

A: 嵌套

B: 递归

C: 直接

D: 间接

答案: 【递归】

## 第 12 讲作用域、生命期和程序的组织结构

### 第 12 讲理解作用域、生命期和程序的组织结构

1、单选题:

不能实现函数之间数据传递的是:

选项:

A: 全局变量

B: 局部变量

C: 函数接口

D: 函数返回值

答案: 【局部变量】

2、单选题:

所有在函数中定义的变量, 连同形式参数, 都是:

选项:

A: 全局变量

- B: 局部变量
- C: 静态变量
- D: 寄存器变量

答案: 【局部变量】

3、单选题:

内部静态变量的作用域是:

选项:

- A: 定义该静态变量的函数外部
- B: 定义该静态变量的函数内部
- C: 定义该静态变量的文件外部
- D: 定义该静态变量的文件内部

答案: 【定义该静态变量的函数内部】

4、单选题:

进行初始化即可自动获取初值为 0 的变量是:

选项:

- A: 任何用 static 修饰的变量
- B: 任何在函数外定义的变量
- C: 局部变量和用 static 修饰的全局变量
- D: 全局变量和用 static 修饰的局部变量

答案: 【全局变量和用 static 修饰的局部变量】

5、单选题:

在函数中, 可以用 auto, extern, register 和 static 四个关键字中的一个来说明变量的存储类型, 如果不说明存储类型, 则默认的存储类型是:

选项:

- A: auto
- B: extern
- C: register
- D: static

答案: 【auto】

6、单选题:

在一个 C++源程序文件中定义的全局变量的有效范围是:

选项:

- A: 该 C++程序的所有源程序文件
- B: 本源程序文件的全部范围
- C: 从定义变量的位置开始到本源程序文件结束
- D: 函数内部全部范围

答案: 【从定义变量的位置开始到本源程序文件结束】

7、单选题:

在 C++中有以下 4 条语句: static int hot=200;int

&rad=hot;hot=hot+100;cout<<rad<<endl;执行这 4 条语句后输出为:

选项:

A: 0

B: 100

C: 200

D: 300

答案: 【 300 】

8、单选题:

下面关于 C++语言变量的叙述错误的是:

选项:

A: C++语言中变量为 auto, static, extern 和 register 四种存储类型

B: 自动变量和外部变量的作用域为整个程序

C: 内部静态变量的作用域是定义它的函数

D: 外部静态变量的作用域是定义它的文件

答案: 【 自动变量和外部变量的作用域为整个程序 】

9、单选题:

如果在一个源文件中定义的函数,只能被本文件中的函数调用,而不能被同一程序其它文件中的函数调用,则说明这个函数为:

选项:

A: 私有函数

B: 内部函数

C: 外部函数

D: 库函数

答案: 【 内部函数 】

10、单选题:

重新定义标识符的作用域规定是外层变量被隐藏, ( ) 变量是可见的。

选项:

A: 外层

B: 内层

C: 外层和内层

D: 内部

答案: 【 内层 】

## 第 13 讲数组的定义和使用

### 第 13 讲掌握数组的定义和使用

1、单选题:

下列关于数组的描述正确是:

选项:

A: 数组的长度是固定的, 而其中元素的数据类型可以不同

B: 数组的长度是固定的，而其中元素的数据类型必须相同

C: 数组的长度是可变的，而其中元素的数据类型可以不同

D: 数组的长度是可变的，而其中元素的数据类型必须相同

答案: 【数组的长度是固定的，而其中元素的数据类型必须相同】

2、单选题:

在 C++语言中引用数组元素时，下面关于数组下标数据类型的说法错误的是:

选项:

A: 整型常量

B: 整型表达式

C: 整型常量或整型表达式

D: 任何类型的表达式

答案: 【任何类型的表达式】

3、单选题:

要定义数组 A，使得其中每个元素的数据分别依次为: 3、9、4、8、0、0、0，错误的定义语句是:

选项:

A: `int A[]={3,9,4,8,0,0,0};`

B: `int A[9]={3,9,4,8,0,0,0};`

C: `int A[]={3,9,4,8};`

D: `int A[7]={3,9,4,8};`

答案: 【`int A[]={3,9,4,8};`】

4、单选题:

有如下数组声明: `int value[30];`，下标值引用错误的是:

选项:

A: `value[30]`

B: `value[0]`

C: `value[10]`

D: `value[20]`

答案: 【`value[30]`】

5、单选题:

以下叙述中错误的是:

选项:

A: 对于 `double` 类型数组，不可以直接用数组名对数组进行整体输入或输出

B: 数组名代表的是数组所占存储区的首地址，其值不可改变

C: 当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息

D: 可以通过赋初值的方式确定数组元素的个数

答案: 【当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出“下标越界”的出错信息】



6、单选题：

以下正确的二维数组定义是：

选项：

A: `int a[][]={1,2,3,4,5,6};`

B: `int a[2][]={1,2,3,4,5,6};`

C: `int a[][3]={1,2,3,4,5,6};`

D: `int a[2,3]={1,2,3,4,5,6};`

答案：【 `int a[][3]={1,2,3,4,5,6};` 】

7、单选题：

以下对二维数组 a 进行初始化正确的是：

选项：

A: `int a[2][]={{1,0,1},{5,2,3}};`

B: `int a[][3]={{1,2,3},{4,5,6}};`

C: `int a[2][4]={{1,2,3},{4,5},{6}};`

D: `int a[][3]={{1,0,1},{},{1,1}};`

答案：【 `int a[][3]={{1,2,3},{4,5,6}};` 】

8、单选题：

若有定义：`int a[3][4];`则正确引用数组 a 元素的是：

选项：

A: `a[2][4]`

B: `a[3][3]`

C: `a[0][0]`

D: `a[3][4]`

答案：【 `a[0][0]` 】

9、单选题：

若有定义：`int a[][4]={0,0};`以下叙述中错误的是：

选项：

A: 数组 a 的每个元素都可得到初值 0

B: 二维数组 a 的第一维大小为 1

C: 因为初值个数除以 a 中第二维大小的值的商为 0，故数组 a 的行数为 1

D: 只有元素 `a[0][0]`和 `a[0][1]`可得到初值 0，其余元素均得不到初值 0

答案：【 只有元素 `a[0][0]`和 `a[0][1]`可得到初值 0，其余元素均得不到初值 0 】

10、单选题：

下面选项中等价的是：

选项：

A: `int a[2][3]={1,0,2,2,4,5}`与 `int a[2][]= {1,0,2,2,4,5};`

B: `int a[][3]= {1,0,2,2,4,5}`与 `int a[2][3]= {1,0,2,2,4,5};`

C: `int a[2][3]={3,4,5}`与 `int a[][3]={3,4,5};`

D: `int a[2][3]={0,1}`与 `int a[2][3]={0},{1};`

答案：【 `int a[][3]= {1,0,2,2,4,5}`与 `int a[2][3]= {1,0,2,2,4,5};` 】

## 第 14 讲数组与函数

### 第 14 讲掌握数组作为函数参数的使用方法

#### 1、单选题：

若用数组名作为函数调用的实参，传递给形参的是：

选项：

- A: 数组的首地址
- B: 数组中第一个元素的值
- C: 数组中的全部元素的值
- D: 数组元素的个数

答案：【 **数组的首地址** 】

#### 2、单选题：

对数组名作函数的参数，下面描述正确的是：

选项：

- A: 数组名作函数的参数，调用时将实参数组复制给形参数组
- B: 数组名作函数的参数，主调函数和被调函数共用一段存储单元
- C: 数组名作参数时，形参定义的数组长度不能省略
- D: 数组名作参数，不能改变主调函数中的数据

答案：【 **数组名作函数的参数，主调函数和被调函数共用一段存储单元** 】

#### 3、单选题：

已知某函数的一个形式参数被说明为 MAT[3][10]，在下列说明中，与此等效的形参说明是：

选项：

- A: int MAT[][10]
- B: int MAT[3][]
- C: int MAT[10][3]
- D: int MAT[][]

答案：【 **int MAT[][10]** 】

#### 4、单选题：

设主调用函数为如下程序段，则函数 f 中对形参数组定义错误的是：int a[3][4];f(a);

选项：

- A: f(int array[3][4])
- B: f(int array[][4])
- C: f(int array[3][])
- D: f(int array[4][3])

答案：【 **f(int array[3][])** 】

#### 5、单选题：

下面程序执行后的输出结果是：#include <iostream>using namespace std;int f(int

```
b[],int m,int n){ int i,s=0; for(i=m;i<n;i++) s=s+b[i-1]; return s;}int main(){ int x,a[]={1,2,3,4,5,6,7,8,9}; x=f(a,3,7); cout<<x; return 0;}
```

选项:

A: 10

B: 18

C: 8

D: 15

答案: 【 18 】

6、单选题:

下面程序执行后的输出结果是: #include <iostream>using namespace std;#define N 20void fun(int a[],int n,int m){ int i; for(i=m;i>=n;i--) a[i+1]=a[i];}int main(){ int i; int a[N]={1,2,3,4,5,6,7,8,9,10}; fun(a,2,9); for(i=0;i<5;i++) cout<<a[i]; return 0;}

选项:

A: 10234

B: 12344

C: 12334

D: 12234

答案: 【 12334 】

7、单选题:

下面程序执行后的输出结果是: #include <iostream>using namespace std;void swap1(int c[]){ int t; t=c[0];c[0]=c[1];c[1]=t;}void swap2(int c0,int c1){ int t; t=c0;c0=c1;c1=t;}int main( ){ int a[2]={3,5},b[2]={3,5}; swap1(a) ; swap2(b[0],b[1]); cout<<a[0] <<a[1] <<b[0] <<b[1]; return 0;}

选项:

A: 5353

B: 5335

C: 3535

D: 3553

答案: 【 5335 】

8、单选题:

下面程序执行后的输出结果是: #include <iostream>using namespace std;int fun(int t[],int n){ int m; if(n>=2) { m=fun(t,n-1); return m; } return t[0];}int main(){ int a[]={11,4,6,3,8,2,3,5,9,2}; cout<<fun(a,10); return 0;}

选项:

A: 10

B: 11

C: 12

D: 13

答案: 【 11 】

## 第 15 讲字符串的处理

## 第 15 讲掌握 C 和 C++的字符串处理方式

### 1、单选题：

字符串的长度是：

选项：

- A: 串中不同字符的个数
- B: 串中不同字母的个数
- C: 串中所含字符的个数且字符个数大于 0
- D: 串中所含字符的个数

答案：【串中所含字符的个数】

### 2、单选题：

下列说法正确的是：

选项：

- A: 字符型数组与整型数组可通用
- B: 字符型数组与字符串其实没什么区别
- C: 当字符串放在字符数组中，这时要求字符数组长度比字符串长 1 个单元，因为要放字符串终结符”
- D: 字符串的输出可以用它所存储的数组来输出，也可以字符串的形式整体输出，结果没区别

答案：【当字符串放在字符数组中，这时要求字符数组长度比字符串长 1 个单元，因为要放字符串终结符”】

### 3、单选题：

下面有关字符数组的描述中错误的是：

选项：

- A: 字符数组可以存放字符串
- B: 字符串可以整体输入、输出
- C: 可以在赋值语句中通过赋值运算对字符数组整体赋值
- D: 不可以用关系运算符对字符数组中的字符串进行比较

答案：【可以在赋值语句中通过赋值运算对字符数组整体赋值】

### 4、单选题：

给出下面定义：char a[]="abcd";char b[]={‘a’,‘b’,‘c’,‘d’};则下列说法正确的是：

选项：

- A: 数组 a 与数组 b 等价
- B: 数组 a 和数组 b 的长度相同
- C: 数组 a 的长度大于数组 b 的长度
- D: 数组 a 的长度小于数组 b 的长度

答案：【数组 a 的长度大于数组 b 的长度】

### 5、单选题：

下面程序输出结果为：`#include<iostream>using namespace std;#include<string.h>  
int main( ){ char st[20]="hellot\"; cout<<strlen(st); cout<<sizeof(st)<<endl;`

```
cout<<st;    return 0;}
```

选项:

A: 520hello

B: 1220hellot

C: 520hellot

D: 1120hello □ □

答案: 【 520hello 】

6、单选题:

要使字符串变量 str 具有初值"Lucky", 正确的定义语句是:

选项:

A: char str[]={ 'L','u','c','k','y' };

B: char str[5]={ 'L','u','c','k','y' };

C: char str []="Lucky";

D: char str [5]="Lucky";

答案: 【 char str []="Lucky"; 】

7、单选题:

下列是为字符数组赋字符串的语句组, 其中错误的是:

选项:

A: char s[10]; s="program";

B: char s[]="program";

C: char s[10]="Hello!";

D: char s[10];strcpy(s,"hello!");

答案: 【 char s[10]; s="program"; 】

8、单选题:

字符数组 s 不能作为字符串使用的是:

选项:

A: char s[]="happy";

B: char s[6]={ 'h','a','p','p','y',' ' };

C: char s[]={ "happy" };

D: char s[5]={ 'h','a','p','p','y' };

答案: 【 char s[5]={ 'h','a','p','p','y' }; 】

9、单选题:

下面程序段执行后的输出结果是: char c[5]={ 'a','b',' ','c',' ' };cout<<c;

选项:

A: 'a' 'b'

B: ab

C: ab c

D: abc

答案: 【 ab 】

10、单选题：

下面程序执行后的输出结果是：#include <iostream>using namespace std;  
#include<string.h>int main(){ char str[]=" SSWLIA" , c; int k; for (k=2;(c=str[k])!="";k++)  
{ switch (c) { case 'l': ++k; break ; case 'L': continue; default : cout<<c; continue ; }  
cout<<'\*'; } return 0;}

选项：

A: SSW

B: SW\*

C: SW\*A

D: SW

答案：【 SW\* 】

## 第 17 讲指针的定义与使用

### 第 17 讲理解指针的定义掌握指针的使用方法

1、单选题：

下面对于指针的描述不正确的是：

选项：

A: 指针是地址变量

B: 指针不能用除 0 以外的常量赋值

C: 两个指针变量的加法无意义

D: 指针指向不同基类型的变量长度不同

答案：【 指针指向不同基类型的变量长度不同 】

2、单选题：

在 int a=3,int \*p=&a; 中， \*p 的值是：

选项：

A: 变量 a 的地址值

B: 无意义

C: 变量 p 的地址值

D: 3

答案：【 3 】

3、单选题：

已知 int a, \*pa=&a;输出指针 pa 十进制的地址值的方法是：

选项：

A: cout<<pa;

B: cout<<\*pa;

C: cout<<&pa;

D: cout<<(long)&pa;

答案：【 cout<<(long)&pa; 】

4、单选题:

变量的指针，其含义是指该变量的:

选项:

A: 值

B: 地址

C: 名

D: 一个标志

答案: 【地址】

5、单选题:

已有定义 `int k=2;int *ptr1,*ptr2;`且 `ptr1` 和 `ptr2` 均已指向变量 `k`，下面不能正确执行的赋值是:

选项:

A: `k=*ptr1+*ptr2`

B: `ptr2=k`

C: `ptr1=ptr2`

D: `k=*ptr1*( *ptr2)`

答案: 【`ptr2=k`】

6、单选题:

若有 `int *p,a=4;`和 `p=&a;`下面那个选项均代表地址?

选项:

A: `a,p,*&a`

B: `&*a,&a,*p`

C: `*&p,*p,&a`

D: `&a,&*p,p`

答案: 【`&a,&*p,p`】

7、单选题:

若变量已正确定义并且指针 `p` 已经指向变量 `x`,则 `*&x` 相当于:

选项:

A: `x`

B: `p`

C: `&x`

D: `&*p`

答案: 【`x`】

8、单选题:

若定义了 `int m,n=0,*p1=&m;`则下列哪个表达式与 `m=n` 等价。

选项:

A: `m=*p1`

B: `*p1=&*n`

C: `*&p1=&*n`

D: `*p1=*&n`

答案: 【`*p1=*&n`】

9、单选题：

假如指针 p 已经指向整型变量 x，则(\*p)++相当于：

选项：

A: x++

B: p++

C: \*(p++)

D: &x++

答案：【 x++ 】

10、单选题：

对于基类型相同的两个指针变量之间，不合理的运算是：

选项：

A: <

B: =

C: +

D: -

答案：【 + 】

## 第 18 讲指针与数组

### 第 18 讲掌握指针对一维数组的操作

1、单选题：

以下四种说法中，正确的一项是：

选项：

A: C++允许在字符串上进行整体操作

B: 语句：char vn[]="Brown";将 vn 定义成一个有 5 个元素的数组，因为"Brown"中含有 5 个字符

C: 对指针只要赋给一个地址值就可以了

D: 一维的指针数组实际上是一个二维数组

答案：【 C++允许在字符串上进行整体操作 】

2、单选题：

定义如下二维数组：int a[5],\*p=a;则下面描述错误的是：

选项：

A: 表达式 p=p+1 是合法的

B: 表达式 a=a+1 是合法的

C: 表达式 p-a 是合法的

D: 表达式 a+2 是合法的

答案：【 表达式 a=a+1 是合法的 】

3、单选题：

若有语句 int a[10]={0,1,2,3,4,5,6,7,8,9},\*p=a; 则哪个不是对 a 数组元素的正确引



用（其中  $0 \leq i < 10$ ）。

选项：

A: `p[i]`

B: `*(*a+i)`

C: `a[p-a]`

D: `*(&a[i])`

答案：【 `*(*a+i)` 】

4、单选题：

有如下程序段：`int *p, a[6]={-1,0,1,2,3,4}; p=a+3;`执行该程序段后，`*p` 的值为：

选项：

A: 4

B: 2

C: 1

D: NULL

答案：【 2 】

5、单选题：

若有定义 `char s[10];`则在下面表达式中不表示 `s[1]`的地址的是：

选项：

A: `s+1`

B: `s++`

C: `&s[0]+1`

D: `&s[1]`

答案：【 `s++` 】

6、单选题：

若要对 `a` 进行合法的自减运算，则之前应有下面哪个说明？

选项：

A: `int p[3]; int *a=p;`

B: `int k; int *a=&k;`

C: `char *a[3];`

D: `int b[10]; int *a=b+1;`

答案：【 `int b[10]; int *a=b+1;` 】

7、单选题：

若有定义 `int x[10]={0,1,2,3,4,5,6,7,8,9}, *p1;`则数值不为 3 的表达式是：

选项：

A: `x[3]`

B: `p1=x+3, *p1++`

C: `p1=x+2, *(p1++)`

D: `p1=x+2, *++p1`

答案：【 `p1=x+2, *(p1++)` 】

8、单选题：

若有定义 `int x[6]={2,4,6,8,5,7}, *p=x,i;` 要求依次输出 `x` 数组 6 个元素中的值，不能完成此操作的语句是：

选项：

A: `for(i=0;i<6;i++) cout<<setw(2)<<*(p++);`

B: `for(i=0;i<6;i++) cout<<setw(2)<<*(p+i);`

C: `for(i=0;i<6;i++) cout<<setw(2)<<*p++;`

D: `for(i=0;i<6;i++) cout<<setw(2)<<(*p)++;`

答案：【 `for(i=0;i<6;i++) cout<<setw(2)<<(*p)++;` 】

9、单选题：

下面程序执行后的输出结果是：  
`#include <iostream>using namespace std;void sum(int *a) { a[0]=a[1];}int main(){ int aa[10]={1,2,3,4,5,6,7,8,9,10},i;for(i=2;i>=0;i--) sum(&aa[i]);cout<<aa[0];return 0;}`

选项：

A: 1

B: 2

C: 3

D: 4

答案：【 4 】

10、单选题：

下面程序执行后的输出结果是：  
`#include <iostream>using namespace std;int main() { int a[10]={1,2,3,4,5,6,7,8,9,10}, *p=&a[3], *q=p+2; cout<<*p+*q; return 0;}`

选项：

A: 16

B: 10

C: 8

D: 6

答案：【 10 】

## 第 19 讲指针与函数

### 第 19 讲掌握指针作为函数参数的用法以及指向函数的指针

1、单选题：

下面程序执行后的输出结果是：  
`#include <iostream>using namespace std;void fun(char *c,char d){ *c=*c+1; d=d+1; cout<<*c<<","<<d<<",";}int main(){ char a='A', b='a'; fun(&b,a); cout<<a<<","<<b<<endl; return 0;}`

选项：

A: B,a,B,a

B: a,B,a,B

C: A,b,A,b

D: b,B,A,b

答案: 【 b,B,A,b 】

2、单选题:

下面程序执行后的输出结果是:  
`#include <iostream>using namespace std;void ss(char *s,char t){ while(*s) { if(*s==t) *s=t-'a'+'A'; s++; }}int main(){ char str1[100]="abcddfefdbd",c='d'; ss(str1,c); cout<<str1; return 0;}`

选项:

A: ABCDDEFEDBD

B: abcDDfefDbD

C: abcAAfefAbA

D: Abcddfefdbd

答案: 【 abcDDfefDbD 】

3、单选题:

以下函数的功能是:  
`fun(char *a,char *b) {while((*a!="")&&(*b!="")&&(*a==*b)) { a++;b++;} return(*a-*b); }`

选项:

A: 计算 a 和 b 所指字符串的长度之差

B: 将 b 所指字符串连接到 a 所指字符串中

C: 将 a 所指字符串连接到 b 所指字符串后面

D: 比较 a 和 b 所指字符串的大小

答案: 【 比较 a 和 b 所指字符串的大小 】

4、单选题:

若有定义 `char *st="how are you";` 下列程序段中正确的是:

选项:

A: `char a[11],*p; strcpy(p=a+1,&st[4]);`

B: `char a[11]; strcpy(++a, st);`

C: `char a[11]; strcpy(a, st);`

D: `char a[],*p; strcpy(p=&a[1],st+2);`

答案: 【 `char a[11],*p; strcpy(p=a+1,&st[4]);` 】

5、单选题:

以下与库函数 `strcpy(char *p1,char *p2)` 功能不相等的程序段是:

选项:

A: `strcpy1(char *p1,char *p2){ while ((*p1++=*p2++)!="") ; }`

B: `strcpy2(char *p1,char *p2){ while ((*p1=*p2)!="") { p1++; p2++ } }`

C: `strcpy3(char *p1,char *p2){ while (*p1++=*p2++) ; }`

D: `strcpy4(char *p1,char *p2){ while (*p2) *p1++=*p2++ ; }`

答案: 【 `strcpy4(char *p1,char *p2){ while (*p2) *p1++=*p2++ ; }` 】

6、单选题:

函数 `char* fun(char *p){ return p; }` 的返回值是:

选项:

- A: 无确切的值
  - B: 形参 p 中存放的地址值
  - C: 一个临时存储单元的地址
  - D: 形参 p 自身的地址值
- 答案: 【形参 p 中存放的地址值】

7、单选题:

若有定义 `int *f();` 中, 标识符 f 代表的是:

选项:

- A: 一个用于指向整型数据的指针变量
- B: 一个用于指向一维数组的行指针
- C: 一个用于指向函数的指针变量
- D: 一个返回值为指针型的函数名

答案: 【一个返回值为指针型的函数名】

8、单选题:

若有定义 `int(*p)();` 标识符 p 可以:

选项:

- A: 表示函数的返回值
- B: 指向函数的入口地址
- C: 表示函数的返回类型
- D: 表示函数名

答案: 【指向函数的入口地址】

9、单选题:

若有函数 `max(a,b)`, 为了让函数指针变量 p 指向函数 max, 正确的赋值方法是:

选项:

- A: `p=max;`
- B: `p=max(a,b);`
- C: `*p=max;`
- D: `*p=max(a,b);`

答案: 【`p=max;`】

10、单选题:

若有以下说明和定义: `int fun(int *c) {.....} int main(){ int (*a)(int`

`*)=fun,*b,w[10],c;}......` 在必要的赋值之后, 对 fun 函数的正确调用语句是:

选项:

- A: `a=a(w);`
- B: `a(&c);`
- C: `b=*b(w);`
- D: `fun(*b);`

答案: 【`a(&c);`】

## 第 20 讲动态分配内存

## 第 20 讲掌握动态分配内存的方法

### 1、单选题：

对使用关键字 `new` 所开辟的动态存储空间，释放时必须使用：

选项：

A: `free`

B: `create`

C: `delete`

D: `realse`

答案：【 `delete` 】

### 2、单选题：

关于动态存储分配，下列说法正确的是：

选项：

A: `new` 和 `delete` 是 C++ 语言中专门用于动态内存分配和释放的函数

B: 动态分配的内存空间也可以被初始化

C: 当系统内存不够时，会自动回收不再使用的内存单元，因此程序中不必使用 `delete` 释放内存空间

D: 当动态分配内存失败时，系统会立刻崩溃，因此一定要慎用 `new`

答案：【 动态分配的内存空间也可以被初始化 】

### 3、单选题：

对 `new` 运算符的下列描述中，哪个是错误的？

选项：

A: 它可以动态创建对象和对象数组

B: 用它创建对象数组时必须指定初始值

C: 用它创建对象时要调用构造函数

D: 用它创建的对象可以使用运算符 `delete` 来释放

答案：【 用它创建对象数组时必须指定初始值 】

### 4、单选题：

若有以下定义，则释放指针所指内存空间的操作是：`float *r=new float[10];`

选项：

A: `delete r ;`

B: `delete *r;`

C: `delete []r;`

D: `delete r[];`

答案：【 `delete []r;` 】

### 5、单选题：

已知 `n` 是一个 `int` 型变量，下列语句中错误的是：

选项：

A: `long *p=new long[n];`

B: `long p[n];`

C: long \*p=new long(n);

D: long p[10];

答案: 【 long p[n]; 】

## 第 21 讲自定义数据类型

### 第 21 讲掌握三种自定义数据类型

#### 1、单选题:

设有结构体说明 struct ex { int x; float y; char z;} example;以下叙述中错误的是:  
选项:

A: struct 是结构体类型的关键字

B: example 是结构体类型名

C: x,y,z 都是结构体成员名

D: struct ex 是结构体类型

答案: 【 example 是结构体类型名 】

#### 2、单选题:

以下哪个定义不会分配实际的存储空间?

选项:

A: struct { char name[10]; int age ;} student ;

B: struct STUDENT { char name[10]; int age ;} student ;

C: struct STUDENT { char name[10]; int age ;};struct STUDENT student;

D: struct STUDENT { char name[10]; int age ;};

答案: 【 struct STUDENT { char name[10]; int age ;}; 】

#### 3、单选题:

在说明一个结构体变量时, 系统分配给它的存储空间是:

选项:

A: 该结构体中的第一个成员所需的存储空间

B: 该结构体中的最后一个成员所需的存储空间

C: 该结构体中占用最大存储空间的成员所需的存储空间

D: 该结构体中所有成员所需存储空间的总和

答案: 【 该结构体中所有成员所需存储空间的总和 】

#### 4、单选题:

已知学生记录描述为: struct student { int no ; char name[20]; char sex; struct { int year; int month ; int day ; } birth;} s; 设结构变量 s 中的“birth”应是“1985 年 10 月 1 日”, 则下面正确的赋值是:

选项:

A: year=1985;month=10;day=1;

B: birth.year=1985;birth.month=10;birth.day=1;

C: s.year=1985;s.month=10;s.day=1;

D: s.birth.year=1985;s.birth.month=10;s.birth.day=1;

答案: 【 s.birth.year=1985;s.birth.month=10;s.birth.day=1; 】

5、单选题:

设有定义 struct ru{long x;float y;} time,\*timep=&time;则对 time 中成员 x 的正确引用是:

选项:

A: ru.time.x

B: timep.x

C: (\*timep).x

D: time->x

答案: 【 (\*timep).x 】

6、单选题:

有以下说明和定义: union dt { int a;char b;double c;} data;以下叙述中错误的是:

选项:

A: data 的每个成员起始地址都相同

B: 变量 data 所占的内存字节数与成员 c 所占字节数相等

C: 程序段: data.a=5;cout<<data.c;输出结果为 5.000000

D: data 可以作为函数的实参

答案: 【 程序段: data.a=5;cout<<data.c;输出结果为 5.000000 】

7、单选题:

若有定义 union data {char ch;int x;} a;下列语句中哪个是不正确的?

选项:

A: a={'x',10}

B: a.x=10;a.x++;

C: a.ch='x';a.ch++;

D: a.x=10;a.ch='x';

答案: 【 a={'x',10} 】

8、单选题:

若有定义 enum color {red,yellow=2,blue,white,black} r=white;执行 cout<<r;后的输出结果是:

选项:

A: 1

B: 2

C: 3

D: 4

答案: 【 4 】

9、单选题:

下面对 typedef 的叙述中错误的是:

选项:

- A: 用 typedef 可以定义各种类型名, 但不能用来定义变量  
B: 用 typedef 可以增加新类型  
C: 用 typedef 只是将已存在的类型用一个新的标识符来代表  
D: 使用 typedef 有利于程序的通用和移植

答案: 【用 typedef 可以增加新类型】

10、单选题:

若有定义 typedef struct {int n;char ch[8];} PER;以下叙述中正确的是:

选项:

- A: PER 是结构体变量名  
B: PER 是结构体类型名  
C: typedef struct 是结构体类型  
D: struct 是结构体类型名

答案: 【PER 是结构体类型名】

## 第 22 讲自定义数据类型的应用——链表

### 第 22 讲掌握链表的结构和使用

1、单选题:

链表不具有的特点是:

选项:

- A: 可随机访问任一元素  
B: 插入、删除不需要移动元素  
C: 不必事先估计存储空间  
D: 所需空间与结点个数成正比

答案: 【可随机访问任一元素】

2、单选题:

链表结点所占存储空间:

选项:

- A: 分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针  
B: 只有一部分, 存放结点值  
C: 只有一部分, 存储表示结点间关系的指针  
D: 分两部分, 一部分存放结点值, 另一部分存放结点所占单元数

答案: 【分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针】

3、单选题:

有以下结构体说明和变量的定义, 且指针 p 指向变量 a, 指针 q 指向变量 b, 则不能把结点 b 连接到结点 a 之后的语句是: struct node {char data; struct node \*next; } a, b, \*p=&a, \*q=&b;

选项:

- A: a.next=q;



B: p.next=&b;

C: p->next=&b;

D: (\*p).next=q;

答案: 【 p.next=&b;】

4、单选题:

在单向链表中, 存储每个结点需有两个域, 一个是数据域, 另一个是指针域, 它指向该结点的:

选项:

A: 直接前趋

B: 直接后继

C: 开始结点

D: 终端结点

答案: 【 直接后继】

5、单选题:

对于一个头指针为 head 的带头结点的单向链表, 判定该表为空表的条件是:

选项:

A: head==NULL

B: head→next==NULL

C: head→next==head

D: head!=NULL

答案: 【 head→next==NULL】

6、单选题:

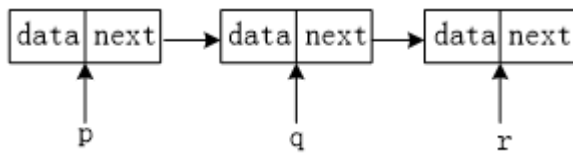
有以下结构体说明和变量定义, 如图所示:

```
struct node {
```

```
int data;
```

```
struct node *next;
```

```
} *p, *q, *r;
```



现要将 q 所指结点从链表中删除，同时要保持链表的连续，以下不能完成指定操作的语句是：

选项：

A: `p->next=q->next;`

B: `p->next=p->next->next;`

C: `p->next=r;`

D: `p=q->next;`

答案：【 `p=q->next;` 】

7、单选题：

有以下定义：

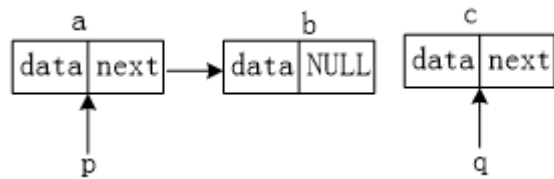
```
struct link {
```

```
int data;
```

```
struct link *next;
```

```
} a,b,c,*p,*q;
```

且变量 a 和 b 之间已有如下图所示的链表结构：



指针 p 指向变量 a，q 指向变量 c。则能够把 c 插入到 a 和 b 之间并形成新的链表的语句组是：

选项：

- A: a.next=c; c.next=b;
- B: p.next=q; q.next=p.next;
- C: p->next=&c; q->next=p->next;
- D: (\*p).next=q; (\*q).next=&b;

答案：【 (\*p).next=q; (\*q).next=&b;】

8、单选题：

有关双向链表的说法正确的是：

选项：

- A: 双向链表实现了对结点的随机访问，是一种随机存储结构
- B: 双向链表的结点含有两个指针域，分别存放其直接前趋和直接后继结点的地址
- C: 双向链表所需空间与单向链表相同
- D: 在双向链表中插入或删除结点时，需要移动结点

答案：【 双向链表的结点含有两个指针域，分别存放其直接前趋和直接后继结点的地址】

9、单选题：

链表是一种采用（ ）存储结构存储的线性表。

选项：

- A: 顺序
- B: 链式
- C: 星式
- D: 网状

答案：【 链式】

10、单选题：

下面程序执行后的输出结果是：  

```
#include <iostream>#include <cmath>using namespace std;struct NODE { int num; struct NODE *next;};int main(){ struct NODE *p,*q,*r; p=new NODE; q=new NODE; r=new NODE; p->num=10; q->num=20;
```

```
r->num=30; p->next=q;q->next=r; cout<<p->num+q->next->num<<endl;    return 0;
}
```

选项:

A: 10

B: 20

C: 30

D: 40

答案: 【 40 】

## 第 23 讲预处理命令

### 第 23 讲掌握预处理命令的使用

1、单选题:

以下叙述中错误的是:

选项:

A: 预处理命令行都必须以#开始

B: 在程序中凡是以#开始的语句行都是预处理命令行

C: C++程序在执行过程中对预处理命令行进行处理

D: 预处理命令行可以出现在 C++程序中任意一行上

答案: 【 C++程序在执行过程中对预处理命令行进行处理 】

2、单选题:

设#define P(x) x/x 执行语句 cout <<P(3\*5);后的输出结果是:

选项:

A: 1

B: 0

C: 25

D: 15

答案: 【 25 】

3、单选题:

在任何情况下计算平方都不会引起二义性的宏定义是:

选项:

A: #define POWER(x) x\*x

B: #define POWER(x) (x)\*(x)

C: #define POWER(x) (x\*x)

D: #define POWER(x) ((x)\*(x))

答案: 【 #define POWER(x) ((x)\*(x)) 】

4、单选题:

下面程序执行后的输出结果是: #include <iostream>using namespace std;#define A 3#define B 2\*A#define C B+Aint main(){ int a=B; cout<<C<<" \ " <<--a<<endl; return

0;}

选项:

A:  $9 \sqcup 5$

B:  $2 \sqcup 3$

C:  $9 \sqcup 3$

D:  $7 \sqcup 5$

答案: 【 $9 \sqcup 5$ 】

5、单选题:

若有宏定义: `#define N 3` `#define Y(n) ((N+1)*n)` 则表达式 `2*(N+Y(5+1))` 的值是:

选项:

A: 出错

B: 42

C: 48

D: 54

答案: 【48】

6、单选题:

已知宏定义 `#define p(x,y,z) x=y*z`; 则宏替换 `p(a,x+5,y-3.1)` 应为:

选项:

A: `a=x+5*y-3.1;`

B: `a=(x+5)*(y-3.1);`

C: `a=x+5*y-3.1`

D: `a=(x+5)*(y-3.1)`

答案: 【`a=x+5*y-3.1;`】

7、单选题:

下面程序执行后的输出结果是: `#include <iostream> using namespace std; #define MA(x) x*x-1`  
`int main(){ int a=1,b=2; cout<<MA(1+a+b); return 0;}`

选项:

A: 6

B: 8

C: 10

D: 12

答案: 【6】

8、单选题:

下面程序执行后的输出结果是: `#include <iostream> using namespace std; #define R 0.5 #define AREA(x) R*x*x`  
`int main(){ int a=1, b=2; cout<<AREA(a+b); return 0;}`

选项:

A: 0.0

B: 0.5

C: 3.5

D: 4.5

答案: 【 4.5 】

9、单选题:

在“文件包含”预处理命令形式中, 当#include 后面的文件名用" " (双引号) 括起时, 寻找被包含文件的方式是:

选项:

- A: 直接按系统设定的标准方式搜索目录
- B: 先在源程序所在目录中搜索, 再按系统设定的标准方式搜索
- C: 仅仅搜索源程序所在目录
- D: 仅仅搜索当前目录

答案: 【 先在源程序所在目录中搜索, 再按系统设定的标准方式搜索 】

10、单选题:

在“文件包含”预处理命令形式中, 当#include 后面的文件名用<> (尖括号) 括起时, 寻找被包含文件的方式是:

选项:

- A: 直接按系统设定的标准方式搜索目录
- B: 先在源程序所在目录中搜索, 再按系统设定的标准方式搜索
- C: 仅仅搜索源程序所在目录
- D: 仅仅搜索当前目录

答案: 【 直接按系统设定的标准方式搜索目录 】

## 第 25 讲类的定义

### 第 25 讲掌握类的基本概念

1、单选题:

下面关于类中概念的描述中错误的是:

选项:

- A: 类是抽象数据类型的实现
- B: 类是具有共同行为的若干对象的统一描述体
- C: 类是创建对象的样板
- D: 类就是 C 语言中的结构体类型

答案: 【 类就是 C 语言中的结构体类型 】

2、单选题:

下列关于 C++ 语言类的描述中错误的是:

选项:

- A: 类用于描述事物的属性和对事物的操作
- B: 类与类之间通过封装而具有明确的独立性
- C: 类与类之间必须是平等的关系, 而不能组成层次结构

D: 类与类之间可以通过一些方法进行通信和联络

答案: 【类与类之间必须是平等的关系，而不能组成层次结构】

3、单选题:

在面向对象设计中，对象有很多基本特点，其中“一个系统中通常包含很多类，这些类之间呈树形结构”这一性质指的是对象的:

选项:

A: 分类性

B: 标识唯一性

C: 继承性

D: 封装性

答案: 【继承性】

4、单选题:

什么是成员:

选项:

A: 类的特征

B: 类的方法

C: 类的事件

D: 以上全是

答案: 【以上全是】

5、单选题:

作用域运算符的功能是:

选项:

A: 标识作用域的级别

B: 指出的大小

C: 标识某个成员是属于哪一类的

D: 给出的范围

答案: 【标识某个成员是属于哪一类的】

## 第 26 讲对象的定义和使用

### 第 26 讲掌握对象的定义和使用

1、单选题:

假定 AA 为一个类，a()为该类公有的函数成员，x 为该类的一个对象，则访问 x 对象中函数成员 a()的格式为:

选项:

A: x.a

B: x.a()

C: x->a

D: x->a()

答案: 【 x.a() 】

2、单选题:

为了使类中的某个成员不能被类的对象通过成员操作符访问,则不能把该成员的访问权限定义为:

选项:

A: public

B: protected

C: private

D: static

答案: 【 public 】

3、单选题:

在用关键字 class 定义的类中, 以下叙述正确的是:

选项:

A: 在类中, 不作特别说明的数据成员均为私有类型

B: 在类中, 不作特别说明的数据成员均为公有类型

C: 类成员的定义必须是成员变量定义在前, 成员函数定义在后

D: 类的成员定义必须放在类定义体内部

答案: 【 在类中, 不作特别说明的数据成员均为私有类型 】

4、单选题:

下列关于成员函数特征的描述中, 错误的是:

选项:

A: 成员函数一定是内联函数

B: 成员函数可以重载

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

答案: 【 成员函数一定是内联函数 】

5、单选题:

下列有关类的说法不正确的是:

选项:

A: 对象是类的一个实例

B: 任何一个对象只能属于一个具体的类

C: 一个类只能有一个对象

D: 类与对象的关系和数据类型与变量的关系相似

答案: 【 一个类只能有一个对象 】

## 第 27 讲构造函数

### 第 27 讲掌握构造函数的功能和使用方法



1、单选题：

类的构造函数被自动调用执行的情况是在定义该类的：

选项：

- A: 成员函数时
- B: 数据成员时
- C: 对象时
- D: 友元函数时

答案：【对象时】

2、单选题：

设类 A 将其它类对象作为成员，则建立 A 类对象时，下列描述正确的是：

选项：

- A: 类构造函数先执行
- B: 成员构造函数先执行
- C: 两者并行执行
- D: 不能确定

答案：【成员构造函数先执行】

3、单选题：

假定一个类的构造函数为“A(int aa=1, int bb=0) {a=aa; b=bb;}”，则执行“A x (4);”语句后，x.a 和 x.b 的值分别为：

选项：

- A: 1 和 0
- B: 1 和 4
- C: 4 和 0
- D: 4 和 1

答案：【4 和 0】

4、单选题：

下列是构造函数的特点，是错误的：

选项：

- A: 构造函数是一种成员函数，它具有一般成员函数的特点
- B: 构造函数的名称与其类名相同
- C: 构造函数必须指明其类型
- D: 一个类中可定义一个或多个构造函数

答案：【构造函数必须指明其类型】

5、单选题：

构造函数不具备的特征的是：

选项：

- A: 构造函数的函数名与类名相同
- B: 构造函数可以重载
- C: 构造函数可以设置默认参数
- D: 构造函数必须指定类型说明

答案：【构造函数必须指定类型说明】

## 第 28 讲默认构造函数和复制构造函数

### 第 28 讲掌握默认构造函数和复制构造函数的使用法则

#### 1、单选题：

假定 AB 为一个类，则执行 AB x(y); 语句时将自动调用该类的：

选项：

A: 有参构造函数

B: 无参构造函数

C: 复制构造函数

D: 赋值构造函数

答案：【复制构造函数】

#### 2、单选题：

通常，复制构造函数的参数是：

选项：

A: 某个对象名

B: 某个对象的成员名

C: 某个对象的引用名

D: 某个对象的指针名

答案：【某个对象的引用名】

#### 3、单选题：

类的默认的复制构造函数：

选项：

A: 在任何情况下都存在

B: 仅当未定义复制构造函数时存在

C: 仅当未定义有参构造函数时存在

D: 仅当未定义任何构造函数时存在

答案：【仅当未定义复制构造函数时存在】

#### 4、单选题：

下列情况中，不会调用复制构造函数的是：

选项：

A: 用一个对象去初始化同一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: 函数的形参是类的对象，调用函数进行形参和实参相结合

D: 函数的返回值是类的对象，函数执行返回调用时

答案：【将类的一个对象赋值给该类的另一个对象时】

#### 5、单选题：

假设 OneClass 为一个类，则该类的复制初始化构造函数的声明语句为：

选项：

- A: OneClass(OneClass p);
- B: OneClass& (OneClass p);
- C: OneClass(OneClass & p);
- D: OneClass (OneClass \*p);

答案: 【 OneClass(OneClass & p); 】

## 第 29 讲析构函数

### 第 29 讲掌握析构函数的用途和使用方法

#### 1、单选题:

下面有关类说法不正确的是:

选项:

- A: 一个类可以有多个构造函数
- B: 一个类只有一个析构函数
- C: 析构函数需要指定参数
- D: 在一个类中可以说明具有类类型的数据成员

答案: 【 析构函数需要指定参数 】

#### 2、单选题:

对类的构造函数和析构函数描述正确的是:

选项:

- A: 构造函数可以重载, 析构函数不能重载
- B: 构造函数不能重载, 析构函数可以重载
- C: 构造函数可以重载, 析构函数也可以重载
- D: 构造函数不能重载, 析构函数也不能重载

答案: 【 构造函数可以重载, 析构函数不能重载 】

#### 3、单选题:

类的析构函数的作用是:

选项:

- A: 一般成员函数
- B: 类的初始化
- C: 对象初始化
- D: 撤销对象时要执行的操作

答案: 【 撤销对象时要执行的操作 】

#### 4、单选题:

下面哪项是对构造函数和析构函数的正确定义?

选项:

- A: X::X(参数), X::~~X()
- B: X::X(参数), X::~~X(参数)
- C: void X::X(参数), void X::~~X(参数)

D: void X::X(), void X::~X()

答案: 【 X::X(参数), X::~X() 】

5、单选题:

以下有关析构函数的叙述不正确的是:

选项:

A: 析构函数没有任何函数类型

B: 析构函数的作用是在对象被撤销时收回先前分配的内存空间

C: 析构函数可以有形参

D: 一个类只有一个析构函数

答案: 【 析构函数可以有形参 】

## 第 30 讲对象数组和对象指针

### 第 30 讲对象数组和对象指针

1、单选题:

有以下的类定义: class MyClass{public:MyClass(){cout<<'1';}};则执行语句 MyClass a,b[2],\*p[2];后, 程序的输出结果是 ( )。

选项:

A: 11

B: 111

C: 1111

D: 11111

答案: 【 111 】

2、单选题:

已知 p 是一个指向类 sample 数据成员 m 的指针, s 是类 sample 的一个对象。如果要给 m 赋值为 5, 哪个是正确的?

选项:

A: s.p=5

B: s->p=5

C: s.\*p=5

D: \*s.p=5

答案: 【 s.\*p=5 】

3、单选题:

假定 AA 是一个类, “AA\* abc()const;”是该类中一个成员函数的原型, 若该函数返回 this 值, 当用 x.abc()调用该成员函数后, x 的值:

选项:

A: 已经被改变

B: 可能被改变

C: 不变

D: 受到函数调用的影响

答案: 【不变】

4、单选题:

下列关于 this 指针的叙述中, 正确的是:

选项:

A: this 指针是一个隐含指针, 它隐含于类的成员函数中

B: 只有在使用 this 时, 系统才会将对象的地址赋值给 this

C: 类的友元函数也有 this 指针

D: this 指针表示了成员函数当前操作的数据所属的对象

答案: 【this 指针表示了成员函数当前操作的数据所属的对象】

5、单选题:

this 指针存在的目的是:

选项:

A: 保证基类公有成员在子类中可以被访问

B: 保证每个对象拥有自己的数据成员, 但共享处理这些数据成员的代码

C: 保证基类保护成员在子类中可以被访问

D: 保证基类私有成员在子类中可以被访问

答案: 【保证每个对象拥有自己的数据成员, 但共享处理这些数据成员的代码】

## 第 31 讲类作用域、对象生命期、const 限定

### 第 31 讲掌握 const 成员的使用

1、单选题:

定 AA 是一个类, “AA\* abc()const;”是该类中一个成员函数的原型, 若该函数返回 this 值, 当用 x.abc()调用该成员函数后, x 的值:

选项:

A: 已经被改变

B: 可能被改变

C: 不变

D: 受到函数调用的影响

答案: 【不变】

2、单选题:

对于常数据成员, 下面描述正确的是:

选项:

A: 常数据成员可以不初始化, 并且不能更新

B: 常数据成员必须被初始化, 并且不能更新

C: 常数据成员可以不初始化, 并且可以被更新

D: 常数据成员必须被初始化, 并且可以更新

答案: 【常数据成员必须被初始化, 并且不能更新】

3、单选题：

下列不能作为类的成员的是：

选项：

A: 自身类对象的指针

B: 自身类对象

C: 自身类对象的引用

D: 另一个类的对象

答案：【自身类对象】

## 第 32 讲静态成员和友元

### 第 32 讲掌握静态成员和友元的使用

1、单选题：

静态数据成员的初始化必须在：

选项：

A: 类内

B: 类外

C: 构造函数内

D: 静态成员函数内

答案：【类外】

2、单选题：

静态成员函数没有：

选项：

A: 返回值

B: this 指针

C: 指针参数

D: 返回类型

答案：【this 指针】

3、单选题：

下列不是描述类的成员函数的是：

选项：

A: 构造函数

B: 析构函数

C: 友元函数

D: 复制构造函数

答案：【友元函数】

4、单选题：

下面关于友元的描述中，错误的是：

选项：

- A: 友元函数可以访问该类的私有数据成员
- 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: 友元可以提高程序的运行效率
- D: 类与类之间的友元关系可以继承

答案: 【类与类之间的友元关系可以继承】

5、单选题:

已知类 A 是类 B 的友元, 类 B 是类 C 的友元, 则:

选项:

- A: 类 A 一定是类 C 的友元
- B: 类 C 一定是类 A 的友元
- C: 类 C 的成员函数可以访问类 B 的对象的所有成员
- D: 类 A 的成员函数可以访问类 B 的对象的所有成员

答案: 【类 A 的成员函数可以访问类 B 的对象的所有成员】

## 第 33 讲类的继承与派生

### 第 33 讲掌握派生类的定义

1、单选题:

定义新类是通过什么实现的

选项:

- A: 信息隐藏
- B: 数据封装
- C: 继承机制
- D: 数据抽象

答案: 【继承机制】

2、单选题:

继承具有 ( ), 即当基类本身也是某一个类派生类时, 底层的派生类也会自动继承间接基类的成员

选项:

- A: 规律性
- B: 传递性
- C: 重复性
- D: 多样性

答案: 【传递性】

3、单选题:

C++类体系中, 不能被派生类继承的有:

选项:

- A: 构造函数
- B: 虚函数

C: 静态成员函数  
D: 赋值操作函数  
答案: 【构造函数】

4、单选题:

C++的继承性允许派生类继承基类的:

选项:

- A: 部分特性, 并允许增加新的特性或重定义基类的特性
- B: 部分特性, 但不允许增加新的特性或重定义基类的特性
- C: 所有特性, 并允许增加新的特性或重定义基类的特性
- D: 所有特性, 但不允许增加新的特性或重定义基类的特性

答案: 【所有特性, 并允许增加新的特性或重定义基类的特性】

5、单选题:

对基类和派生类的关系描述中, 错误的是:

选项:

- A: 派生类是基类的具体化
- B: 基类继承了派生类的属性
- C: 派生类是基类定义的延续
- D: 派生类是基类的特殊化

答案: 【基类继承了派生类的属性】

## 第 34 讲派生类成员的访问

### 第 34 讲掌握派生类成员的访问方法

1、单选题:

若要用派生类的对象访问基类的保护成员, 以下观点正确的是:

选项:

- A: 可采用公有继承
- B: 可采用私有继承
- C: 可采用保护继承
- D: 不可能实现

答案: 【不可能实现】

2、单选题:

派生类的对象对它的基类成员中 ( ) 是可以访问的?

选项:

- A: 公有继承的公有成员
- B: 公有继承的私有成员
- C: 公有继承的保护成员
- D: 私有继承的公有成员

答案: 【公有继承的公有成员】



### 3、单选题：

下列说法中错误的是：

选项：

- A: 公有继承时基类中的 `public` 成员在派生类中仍是 `public` 成员
- B: 私有继承时基类中的 `protected` 成员在派生类中仍是 `protected` 成员
- C: 私有继承时基类的 `public` 成员在派生类中是 `private` 成员
- D: 保护继承时基类的 `public` 成员在派生类中是 `protected` 成员

答案：【私有继承时基类中的 `protected` 成员在派生类中仍是 `protected` 成员】

### 4、单选题：

下面叙述不正确的是：

选项：

- A: 派生类一般都用公有派生
- B: 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: 基类的公有成员在派生类中仍然是公有的

答案：【基类的公有成员在派生类中仍然是公有的】

### 5、单选题：

若派生类的成员函数不能直接访问基类中继承来的某个成员，则该成员一定是基类中的：

选项：

- A: 私有成员
- B: 公有成员
- C: 保护成员
- D: 保护成员或私有成员

答案：【私有成员】

## 第 35 讲派生类的构造和析构函数

### 第 35 讲掌握派生类的构造和析构函数

#### 1、单选题：

有如下程序：`#include <iostream>using namespace std;class Base{protected: Base(){cout<<'A';} Base(char c){cout<<c;}};class Derived:public Base{public: Derived(char c){cout<<c;}};int main(){ Derived d1('B'); return 0;}`执行这个程序屏幕上将显示输出：

选项：

- A: B
- B: BA
- C: AB
- D: BB

答案：【AB】

2、单选题：

派生类的构造函数的成员初始化列表中，不能包含：

选项：

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

答案：【基类中子对象的初始化】

3、单选题：

假设 `ClassY:publicX`，即类 Y 是类 X 的派生类，则说明一个 Y 类的对象时和删除 Y 类对象时，调用构造函数和析构函数的次序分别为：

选项：

- A: X,Y,Y,X
- B: X,Y,X,Y
- C: Y,X,X,Y
- D: Y,X,Y,X

答案：【X,Y,Y,X】

4、单选题：

以下有关继承的叙述正确的是：

选项：

- A: 构造函数和析构函数都能被继承
- B: 派生类是基类的组合
- C: 派生类对象除了能访问自己的成员以外，不能访问基类中的所有成员
- D: 基类的公有成员一定能被派生类的对象访问

答案：【派生类是基类的组合】

5、单选题：

建立包含有类对象成员的派生类对象时，自动调用构造函数的执行顺序依次为：

选项：

- A: 自己所属类、对象成员所属类、基类的构造函数
- B: 对象成员所属类、基类、自己所属类的构造函数
- C: 类、对象成员所属类、自己所属类的构造函数
- D: 基类、自己所属类、对象成员所属类的构造函数

答案：【类、对象成员所属类、自己所属类的构造函数】

## 第 36 讲多重继承

### 第 36 讲掌握多重继承的规则

1、单选题：

下列关于多继承二义性的描述中，错误的是：

选项：

- A: 一个派生类的两个基类中都有某个同名成员，派生类中这个成员的访问可能出现二义性
- B: 解决二义性的最常用的方法是对成员名的限定法
- C: 基类和派生类中同时出现的同名函数，也存在二义性问题
- D: 一个派生类是从两个基类派生出来的，而这两个基类又有一个共同的基类，对该基类成员进行访问时，可能出现二义性

答案：【基类和派生类中同时出现的同名函数，也存在二义性问题】

2、单选题：

解决二义性问题的方法有：

选项：

- A: 只能使用作用域分辨操作符
- B: 使用作用域分辨操作符或赋值兼容规则
- C: 使用作用域分辨操作符或虚基类
- D: 使用虚基类或赋值兼容规则

答案：【使用作用域分辨操作符或虚基类】

3、单选题：

假设类X以类Y作为它的一个基类，并且X类的名字func()支配Y类的名字func()，obj为类X的对象，则obj.func()语句实现的功能为：

选项：

- A: 先执行类X的func()，再执行访问类Y的func()
- B: 先执行类Y的func()，再执行访问类X的func()
- C: 执行类X的func()
- D: 执行类Y的func()

答案：【执行类X的func()】

4、单选题：

多重继承的构造顺序可分为如下四步：（1）所有非虚基类的构造函数按照他们被继承的顺序构造（2）所有虚基类的构造函数按照他们被继承的顺序构造（3）所有子对象的构造函数按照他们被继承的顺序构造（4）派生类自己的构造函数体这四个步骤的正确顺序是：

选项：

- A: （4）（3）（2）（1）
- B: （2）（4）（3）（1）
- C: （2）（1）（3）（4）
- D: （3）（4）（1）（2）

答案：【（2）（1）（3）（4）】

5、单选题：

带有基类的多层派生类构造函数的成员初始化列表中都要排出虚基类的构造函数，这样将对虚基类的子对象初始化：

选项:

A: 与虚基类下面的派生类个数有关

B: 多次

C: 二次

D: 一次

答案: 【一次】

## 第 37 讲多态性

### 第 37 讲掌握多态性的概念和实现原理

1、单选题:

下列关于动态联编的描述中, 错误的是:

选项:

A: 动态联编是以虚函数为基础

B: 动态联编是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

答案: 【动态联编是在编译时确定操作函数的】

2、单选题:

要实现动态联编, 必须通过 ( ) 调用虚函数?

选项:

A: 对象指针

B: 成员名限定

C: 对象名

D: 派生类名

答案: 【对象指针】

3、单选题:

对虚函数的调用:

选项:

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

答案: 【不一定使用动态联编】

4、单选题:

以下叙述正确的是:

选项:

A: 构造函数调用虚函数采用动态联编

B: 构造函数可以说明为虚函数

C: 当基类的析构函数是虚函数时，它的派生类的析构函数也是虚函数

D: 析构函数必须是虚函数

答案: 【当基类的析构函数是虚函数时，它的派生类的析构函数也是虚函数】

5、单选题:

对虚函数使用对象指针或引用，系统使用（），对虚函数使用对象调用时，系统使用（）联编。

选项:

A: 静态联编、静态联编

B: 动态联编、动态联编

C: 静态联编、动态联编

D: 动态联编、静态联编

答案: 【动态联编、静态联编】

## 第 38 讲虚函数

### 第 38 讲掌握虚函数、纯虚函数、抽象类的定义和使用

1、单选题:

以下成员函数哪个表示纯虚函数?

选项:

A: virtual int vf (int)

B: void vf(int)=0

C: virtual void vf()=0

D: virtual void vf(int){}

答案: 【virtual void vf()=0】

2、单选题:

关于虚函数的描述中，哪个是正确的?

选项:

A: 虚函数是一个静态成员函数

B: 虚函数是一个非成员函数

C: 虚函数既可以在函数说明时定义，也可以在函数实现时定义

D: 派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型

答案: 【派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型】

3、单选题:

关于纯虚函数和抽象类的描述中，错误的是:

选项:

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，该基类派生类一定不再是抽象类

D: 抽象类只能作为基类来使用，其纯虚函数的实现由派生类给出

答案: 【一个基类中说明有纯虚函数，该基类派生类一定不再是抽象类】

4、单选题:

关于抽象类，下列表述中正确的是:

选项:

A: 抽象类的成员函数中至少有一个是没有实现的函数（即无函数体定义的函数）

B: 派生类必须实现作为基类的抽象类中的纯虚函数

C: 派生类不可能成为抽象型

D: 抽象类可以用来定义对象

答案: 【抽象类的成员函数中至少有一个是没有实现的函数（即无函数体定义的函数）】

5、单选题:

下列关于纯虚函数的描述中，正确的是:

选项:

A: 纯虚函数是一种特殊的虚函数，它是个空函数

B: 具有纯虚函数的类称为虚基类

C: 一个基类中说明有纯虚函数，其派生类一定要实现该纯虚函数

D: 具有纯虚函数的类不能创建类对象

答案: 【具有纯虚函数的类不能创建类对象】

## 第 39 讲运算符重载

### 第 39 讲掌握运算符重载的方法

1、单选题:

下列有关运算符重载的描述中，哪个是正确的？

选项:

A: 运算符重载可改变其优先级

B: 运算符重载不改变其语法结构

C: 运算符重载可改变其结合性

D: 运算符重载可改变其操作数的个数

答案: 【运算符重载不改变其语法结构】

2、单选题:

在重载一个运算符时，其参数表中没有任何参数，这表明该运算符是:

选项:

A: 作为友元函数重载的 1 元运算符

B: 作为成员函数重载的 1 元运算符

C: 作为友元函数重载的 2 元运算符

D: 作为成员函数重载的 2 元运算符

答案: 【作为成员函数重载的 1 元运算符】

3、单选题：

在下面的运算符重载函数的原型中，错误的是：

选项：

A: volume operator-(double,double);

B: double volume::operator-(double);

C: volume volume ::operator-(volume);

D: volume operator-(volume,volume);

答案：【 volume operator-(double,double); 】

4、单选题：

下列运算符中，哪个运算符在 C++中不能重载？

选项：

A: ?:

B: []

C: new

D: &&

答案：【 ?: 】

5、单选题：

如果表达式++a 中的“++”是作为成员函数重载的运算符，若采用运算符函数调用格式，则可表示为：

选项：

A: a.operator++

B: operator++(a)

C: operator++(a,1)

D: a.operator++()

答案：【 a.operator++() 】

## 第 40 讲典型运算符的重载

### 第 40 讲掌握多种运算符的重载

1、单选题：

关于运算符重载，下列表述中正确的是：

选项：

A: C++已有的任何运算符都可以重载

B: 运算符函数的返回类型不能声明为基本数据类型

C: 在类型转换函数的定义中不需要声明返回类型

D: 可以通过运算符重载来创建 C++中原来没有的运算符

答案：【 在类型转换函数的定义中不需要声明返回类型 】

2、单选题：

重载输入流运算符>>必须使用的原型为：

选项:

A: ostream&operator>>(ostream&,<类名>);

B: istream&operator>>(istream&,<类名>&);

C: ostream&operator>>(ostream,<类名>&);

D: <类名> operator>>(istream&,<类名>&);

答案: 【 istream&operator>>(istream&,<类名>&); 】

3、单选题:

哪个运算符既可以重载为一元运算符, 又可重载为二元运算符?

选项:

A: “\*”

B: “=”

C: “=”和“\*”

D: “\*”和“\*++”

答案: 【 “\*” 】

4、单选题:

友元运算 obj1>obj2 被 C++编译器解释为:

选项:

A: operator>(obj1,obj2)

B: >(obj1,obj2)

C: obj2.operator>(obj1)

D: obj1.operator>(obj2)

答案: 【 operator>(obj1,obj2) 】

5、单选题:

下列运算符不能重载为友元函数的是?

选项:

A: = ( ) [] ->

B: + - ++ --

C: > < >= <=

D: += -= \*= /=

答案: 【 = ( ) [] -> 】

## 第 41 讲类模板

### 第 41 讲掌握类模板的定义和使用

1、单选题:

下列关于模板的叙述中, 错误的是:

选项:

A: 模板声明中的第一个符号总是关键字 template

B: 在模板生命中用<>括起来的部分是模板的形参表



- C: 类模板不能有数据成员  
D: 在一定条件下函数模板的实参可以省略  
答案: 【类模板不能有数据成员】

2、单选题:

下列程序段中有错的是:

选项:

- A: `template <class Type>`  
B: `Type`  
C: `func(Type a,b)`  
D: `{return (a>b)?(a):(b) ;}`

答案: 【`func(Type a,b)`】

3、单选题:

如果一个模板声明列出了多个参数,则每个参数之间必须使用逗号隔开,每个参数都必须使用 ( ) 关键字来修饰

选项:

- A: `const`  
B: `static`  
C: `void`  
D: `class`

答案: 【`class`】

4、单选题:

类模板的模板参数:

选项:

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

答案: 【以上三者均可】

5、单选题:

模板对类型的参数化提供了很好的支持,因此:

选项:

- A: 类模板的主要作用是生成抽象类  
B: 类模板实例化时,编译器将根据给出的模板实参生成一个类  
C: 在类模板中的数据成员都具有同样类型  
D: 类模板中的成员函数都没有返回值

答案: 【类模板实例化时,编译器将根据给出的模板实参生成一个类】

## 第 42 讲标准库

## 第 42 讲了解标准库的内容、掌握字符串流的使用

### 1、单选题：

下列关于 C++流的说明中，正确的是：

选项：

A: 与键盘、屏幕、打印机和通信端口的交互都可以通过流类来实现

B: 从流中获取数据的操作称为插入操作，向流中添加数据的操作称为提取操作

C: cin 是一个预定义的输入流类

D: 输出流有一个名为 open 的成员函数，其作用是生成一个新的流对象

答案：【与键盘、屏幕、打印机和通信端口的交互都可以通过流类来实现】

### 2、单选题：

在 C++中使用流进行输入输出，其中用于屏幕输出的对象是：

选项：

A: cerr

B: cin

C: cout

D: cfile

答案：【cout】

### 3、单选题：

out、cerr、clog 是（ ）的对象，cout 处理标准输出，cerr 和 clog 都处理标准出错信息

选项：

A: istream

B: ostream

C: cerr

D: clog

答案：【ostream】

### 4、单选题：

在下面格式化命令的解释中，错误的是：

选项：

A: ios::skipws          跳过输入中的空白字符

B: ios::fill()          获得当前的填充字符

C: ios::hex          转换基数为八进制形式

D: ios::precision          返回当前的精度

答案：【ios::hex          转换基数为八进制形式】

### 5、单选题：

在进行完任何 C++流的操作后，都可以用 C++流的有关成员函数检测流的状态；其中只能用于检测输入流状态的操作函数名称是：

选项：

A: fail

B: eof  
C: bad  
D: good  
答案: 【 eof 】

## 第 43 讲文件

### 第 43 讲掌握文件的基本操作

#### 1、单选题:

在 C++ 中, 打开一个文件, 就是将整个文件与一个 ( ) 建立关联, 关闭一个文件, 就是取消这种关联

选项:

- A: 类
- B: 流
- C: 对象
- D: 结构

答案: 【 流 】

#### 2、单选题:

下列打开文件的表达式中, 错误的是:

选项:

- A: ofstream ofile; ofile.open("C:\\vc\\abc.txt", ios::binary);
- B: fstream iofile; iofile.open("abc.txt", ios::ate);
- C: ifstream ifile ("C:\\vc\\abc.txt");
- D: cout.open("C:\\vc\\abc.txt", ios::binary);

答案: 【 cout.open("C:\\vc\\abc.txt", ios::binary); 】

#### 3、单选题:

以下关于文件操作的叙述中, 不正确的是:

选项:

- A: 打开文件的目的是使文件对象与磁盘文件建立联系
- B: 文件读写过程中, 程序将直接与磁盘文件进行数据交换
- C: 关闭文件的目的之一是保证将输出的数据写入硬盘文件
- D: 关闭文件的目的之一是释放内存中的文件对象

答案: 【 文件读写过程中, 程序将直接与磁盘文件进行数据交换 】

#### 4、单选题:

语句 ofstream f("SALARY.DAT", ios::noreplace | ios::binary); 的功能是建立流对象 f, 并试图打开文件 SALARY.DAT 并与之连接, 而且:

选项:

- A: 若文件存在, 将文件指针定位于文件尾; 若文件不存在, 建立一个新文件
- B: 若文件存在, 将其截为空文件; 若文件不存在, 打开失败

C: 若文件存在, 将文件指针定位于文件首; 若文件不存在, 建立一个新文件

D: 若文件存在, 打开失败; 若文件不存在, 建立一个新文件

答案: 【若文件存在, 打开失败; 若文件不存在, 建立一个新文件】

5、单选题:

执行语句序列 `ofstream outf("SALARY.DAT"); if(...)cout<<"成功"; else cout<<"失败";` 后, 如文件打开成功, 显示"成功", 否则显示"失败". 由此可知, 上面 if 语句的条件表达式是:

选项:

A: `!outf` 或者 `outf.fail()`

B: `!outf` 或者 `outf.good()`

C: `outf` 或者 `outf.fail()`

D: `outf` 或者 `outf.good()`

答案: 【`outf` 或者 `outf.good()`】

## 第 47 讲异常处理

### 第 47 讲掌握异常处理的使用

1、单选题:

下列关于异常的叙述错误的是:

选项:

A: 编译错属于异常, 可以抛出

B: 运行错属于异常

C: 硬件故障也可当异常抛出

D: 只要是编程者认为是异常的都可当异常抛出

答案: 【编译错属于异常, 可以抛出】

2、单选题:

下列叙述错误的是:

选项:

A: `throw` 的操作数表示异常类型

B: `throw` 的操作数值可以区别不同的异常

C: `throw` 抛出不同异常时需要用不同的操作数类型来区分

D: `throw` 语句抛出的异常可以不被捕获

答案: 【`throw` 的操作数值可以区别不同的异常】

3、单选题:

关于函数声明 `float f()throw()`, 下列叙述正确的是:

选项:

A: 表明函数抛出 `float` 类型异常

B: 表明函数抛出任何类型异常

C: 表明函数不抛出任何类型异常

D: 表明函数实际抛出的异常

答案: 【表明函数不抛出任何类型异常】

4、单选题:

下列叙述错误的是:

选项:

A: catch (...) 语句可捕获所有类型的异常

B: 一个 try 语句可以有多个 catch 语句

C: catch (...) 语句可以放在 catch 语句组的前面

D: 程序中 try 语句与 catch 语句是一个整体, 缺一不可

答案: 【catch (...) 语句可以放在 catch 语句组的前面】

5、单选题:

下列程序运行结果为: `class A{public: ~A() {cout<<"A"<<"n"; }};char fun0() { A A1; throw('E'); return '0';}int main(){ try{ cout<<fun0()<<"n";} catch(char c) { cout<<c<<"n";} return 0;}`

选项:

A: AE

B: OAE

C: OE

D: E

答案: 【AE】

## 第 48 讲命名空间

### 第 48 讲掌握命名空间的使用

1、单选题:

下面关于命名空间的说法, 错误的是:

选项:

A: 命名空间的引入让程序员可以在不同的模块中使用相同名字表示不同事物

B: 一个命名空间中可以集合很多不同的标识符

C: 一个命名空间中的标识符命名作用域相同

D: 一个命名空间对应多个命名作用域

答案: 【一个命名空间对应多个命名作用域】

2、单选题:

要说明标识符是属于哪个命名空间时, 需要在标识符和命名空间名字之间加上:

选项:

A: ::

B: ->

C: .

D: ()

答案: 【 :: 】

### 3、单选题:

下面关于 namespace 与 class、struct、union、enum 的区别描述正确的是: (1) namespace 只能在全局范畴定义,但他们之间可以互相嵌套;(2)在 namespace 定义的末尾,右大括号的后面不必要跟一个分号;(3)一个 namespace 可以在多个头文件中定义,就好像重复定义一个类一样;多个定义中的函数或者类型合在一起构成整个 namespace;(4)一个 namespace 可以用另一个名字来作为别名。(5)不能像类那样创建一个 namespace 的实例。

选项:

A: (2) (4)

B: (1) (2) (3)

C: (2) (3) (4) (5)

D: 全部

答案: 【 全部 】

### 4、单选题:

如果程序中使用了 using 命令同时引用了多个命名空间,并且命名空间中存在相同的函数,将出现:

选项:

A: 编译错误

B: 语法错误

C: 逻辑错误

D: 无法判定错误类型

答案: 【 编译错误 】

### 5、单选题:

如果在函数中定义的局部变量与命名空间中的变量同名时, ( ) 被隐藏

选项:

A: 函数中的变量

B: 命名空间中的变量

C: 两个变量都

D: 两个变量都不

答案: 【 命名空间中的变量 】