

## C++前四章小测试

### ● 基本信息

姓名：

学号：

1.下列字符串可以用作 c++标识符的是(多选)： (AE)

A.\_123 B.2009var C.goto D.test - 2009 E.Good F.3St

解析：标识符用来表示函数、类型及变量的名称，是字母、下划线和数字的排列。

1、必须用字母或下划线开头。B.F 错误

2、只能是字母、下划线、数字的组合，不能出现其他符号。D 错误

3、标识符的名称不能是 C++语言中的关键字，关键字是具有特定含义的标识符。

C 是关键字，错误。

2.以下属于 c++基本数据类型的有（多选）：（BCD）

A.wide

B.float

C.bool

D.int

E.str

3. 有以下输出语句：

```
cout<<"abc\\ ";  
cout<<"I say \\\"Thank you!\\\"";
```

则会输出：(B)

A.abc\  
I say "Thank you!"

B.abc\ I say "Thank you!"

C.abc\\ I say "Thank you!"

D.abc\ I say \"Thank you!\"

4.有关 C++常量、变量，以下定义语句中正确的是（D）。

A、char a='A' b='B';

B、float a=b=10.0;

C、#define PRICE 30;

D、const float PI=3.1415926;

解析：A 选项 b 之前没有逗号；B 中 b 这个变量没有定义，变量在使用前需要定义；C 末尾不要加分号

5.以下程序的输出结果是：(D)

```
1  #include <iostream.h>
2  void main() {
3      int x=3,y=3;
4      switch(x%2) {
5          case 1:
6              switch (y) {
7                  case 0:cout<<"first";
8                  case 1:cout<<"second";break;
9                  default: cout<<"hello";
10             }
11         case 2:cout<<"third";
12     }
13 }
```

A. second third   B.hello   C.first second   **D.hellothird**

解析：switch 语句从哪进直到 break 退出。 x=3，switch(x%2)语句，x%2 取余是 1，逻辑语句为真，进入 case1:switch(y)语句，此时 y=3，进入 default 语句，输出 hello，但此没有 break 程序没结束，再次进入 switch(x%2)语句中的 case 2： 输出 third 程序才结束。

6.下列程序的输出结果为\_\_0 1\_\_

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int x,y;
7      for (x=0,y=0;x+y<2;++x){
8          y++;
9          cout<<x<<" ";
10         cout<<y<<" ";
11     }
12     return 0;
13 }
```

7.执行完下列语句后，i 的值为\_\_1010\_\_

```
int i,j;
j = 10;
i = (j++,j+100,999+j);
```

8.下列程序的输出结果是：            (C)

```

1  # include <iostream>
2  using namespace std;
3  int main(){
4      int n=1,m,a,b=0;
5      if ((m=a>=b) || (n=a<b)){
6          cout<<"m="<<m<<" n="<<n;
7      }
8      else{
9          cout<<"m="<<m<<" n="<<n;
10     }
11     return 0;
12 }
13

```

A.m=0 n=0   B.m=1 n=1   C.m=1 n=0   D.m=0 n=1

9.运行下列语句后，var 的值为： 30

```

int var,y=5;
var = (y < 10) ? 30 : 40;

```

10.下列程序的输出结果是： (B)

```

1  # include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n=256; // 二进制为 0000 0001 0000 0000
6      char a=(char)n;
7      cout<<n<<" ";
8      n=a;
9      cout<<n<<endl;
10     return 0;
11 }

```

A.256 256   B.256 0   C.256 128   D.256 1

11.有如下程序段：

```

1  int i = 1;
2  while(1)
3  {
4      i++;
5      if(i == 10)
6      {
7          break;
8      }
9      if(i%2 == 0)
10     {
11         cout<<"*";
12     }
13 }

```

执行这个程序段输出字符\*的个数是 (C)

A.10   B.3   C.4   D.5

12.已知枚举类型声明语句为：enum COLOR{WHITE,YELLOW,GREEN=5,RED,BLACK=10}; 则下列说法中错误的是 (D)

- A. 枚举常量 YELLOW 的值为 1
- B. 枚举常量 RED 的值为 6
- C. 枚举常量 BLACK 的值为 10
- D. 枚举常量 WHITE 的值为 1

解析：枚举值是常量，不是变量。不能在程序中用赋值语句再对它赋值；枚举元素本身由系统定义了一个表示序号的数值，从 0 开始顺序定义为 0, 1, 2…。所以枚举函数中对象的值应从 0 开始，依次加 1，如果有的对象已赋值，则后面也按照前面的值依次加 1 的原则。

13.在函数声明中，下面哪项不是必要的：（D）

- A. 函数参数的类型和参数名
- B. 函数名
- C. 函数的类型
- D. 函数体

14.如果一个函数无返回值，定义时它的函数类型应是（C）

- A.任意
- B.int
- C.void
- D.无

15.在 c++语言中，对函数参数默认值描述正确的是（D）

- A. 函数参数的默认值只能设定一个
- B. 一个函数的参数若有多个，则参数默认值的设定可以不连续
- C. 函数参数必须设定默认值
- D. 在设定了参数的默认值后，该参数后面定义的所有参数都必须设定默认值

16.下列程序的输出结果是（25）

```
1  #include <iostream>
2  using namespace std;
3
4  int fun2(int n){
5      return n*n;
6  }
7
8  int fun1(int x,int y){
9      return fun2(x) + fun2(y);
10 }
11
12 int main(){
13     int a=3,b=4,c;
14     c = fun1(3,4);
15     cout<<c<<endl;
16     return 0;
17 }
```

17.用递归的方法编写函数求 Fibonacci 级数，公式为  $F_n = F_{n-1} + F_{n-2}$  ( $n > 2$ ) ,  $F_1 = F_2 = 1$ 。

```

1  int Fibonacci(int n)
2  {
3      if(n==1||n==2)
4      {
5          return 1;
6      }
7      else
8      {
9          return Fibonacci(n-1)+Fibonacci(n-2);
10     }
11 }

```

18.若有下面的函数调用：

fun(a+b,3,max(n-1,b));

其中实参的个数是 (A)

A.3 B.4 C.5 D.6

19.给出下列程序中的输出结果

```

1  #include <iostream>
2  using namespace std;
3
4  void swap(int& a,int& b){
5      int t = a;
6      a = b;
7      b = t;
8  }
9
10 int main(){
11     int x = 10,y = 5;
12     cout<<"x="<<x<<" y="<<y<<endl;
13     swap(x,y);
14     cout<<"x="<<x<<" y="<<y<<endl;
15     return 0;
16 }

```

结果：

x=10 y=5

x=5 y=10

20.下列对函数 double add(int a , int b)进行重载，正确的是(多选) (ABC)

A.int add(int a ,int b ,int c)

B.int add(double a , double b)

C.double add(double a , double b)

D.int add(int a , int b)

A. 解析：在 C++中，函数的重载取决于函数的特征标即函数的参数列表。同名的函数，特征标不一样则函数不一样，A、B、C 的参数列表与题目给出的特征标数量或类型不一样，因此正确答案为 A、B、C。

21.以下不属于面向对象的程序设计的特征的是 (C)

A.抽象

- B.封装
- C.派生
- D.多态

22.下列有关类的说法不正确的是 (C)

- 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

23.关于内联函数正确的是 (B)

- A.类的私有成员函数不能作为内联函数
- B.在所有类说明中内部定义的成员函数都是内联函数
- C.类的保护成员函数不能作为内联函数
- D.使用内联函数的地方会在运行阶段用内联函数体替换掉

24.定义并实现一个矩形类 (rectangle)，有长 (high)，宽 (wide) 两个属性，默认长和宽分别为 3.0 和 4.0 (要求用带默认形参值的成员函数 set()实现)，有成员函数 area()用来计算矩形的面积。 (考察类的实现，带默认形参值的成员函数)

```
class rectangle{
public:
    void set(float x=3,float y=4);
    void area();
private:
    float high,wide;
};

void rectangle::set(float x,float y){
    high = x;
    wide = y;
}

void rectangle::area(){
    cout<<high*wide<<endl;
}
```

25.请指出下列程序中的错误，并改正。（考察类成员的访问）

```
1  #include<iostream>
2  using namespace std;
3
4  class Point{
5      public:
6          void SetPoint(int a);
7          void display();
8      private:
9          int x;
10 };
11
12 void Point::SetPoint(int a){
13     x = a;
14 }
15
16 void Point::display(){
17     cout<<x<<endl;
18 }
19
20
21 int main()
22 {
23     Point a;
24     a.SetPoint(25);
25     cout<<a.x<<endl;
26     return 0;
27 }
```

答案：不能直接访问私有成员。将 25 行代码改为 `a.display();`

26.下列程序的输出结果是： 30 30 （考察对象的赋值）

```

1  #include<iostream>
2  using namespace std;
3
4  class Point{
5  public:
6      void SetPoint(int a);
7      void display();
8  private:
9      int x;
10 };
11
12 void Point::SetPoint(int a){
13     x = a;
14 }
15
16 void Point::display(){
17     cout<<x<<" ";
18 }
19
20
21 int main()
22 {
23     Point a,b;
24     a.SetPoint(25);
25     b.SetPoint(30);
26     a = b;
27     a.display();
28     b.display();
29     return 0;
30 }

```

27.下列程序的输出结果是： （考察类的构造函数，析构函数和复制构造函数）

```

1  #include<iostream>
2  using namespace std;
3
4  class Point{
5  public:
6      Point(int xx=0,int yy=0){x = xx;y = yy;}
7      Point(Point& p);
8      ~Point();
9      int getX(){return x;}
10     int getY(){return y;}
11 private:
12     int x,y;
13 };
14
15 Point::Point(Point& p){
16     x = p.x;
17     y = p.y;
18     cout<<"copy constructor"<<endl;
19 }
20
21 Point::~~Point(){
22     cout<<"destructing"<<endl;
23 }
24
25
26 int main()
27 {
28     Point a,b(1,2);
29     Point c(b);
30     cout<<c.getY()<<endl;
31     return 0;
32 }

```



答案:

copy constructor

2

destructing

destructing

destructing

28. 下列程序的输出结果是 (考察组合类的构造函数和复制构造函数)

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  class Point{ //定义Point类
6  public:
7      Point(int xx=0){x = xx;}
8      Point(Point& p);
9      int getX(){return x;}
10 private:
11     int x;
12 };
13
14 Point::Point(Point& p){
15     x = p.x;
16     cout<<"copy constructor of the point"<<endl;
17 }
18
19 class Line{ //定义Line组合类
20 public:
21     Line(Point xp1);
22     Line(Line& l);
23     int getLen(){return len;}
24 private:
25     Point p1;
26     int len;
27 };
28
29 Line::Line(Point xp1):p1(xp1){ //组合类的构造函数
30     cout<<"constructor of line"<<endl;
31     len = p1.getX();
32 }
33
34 Line::Line(Line& l):p1(l.p1){ //组合类的复制构造函数
35     cout<<"copy constructor of line"<<endl;
36     len = l.len;
37 }
38
39
40 int main()
41 {
42     Point a(2);
43     Line line1(a);
44     Line line2(line1);
45     cout<<"the length of line2 is "<<line2.getLen()<<endl;
46     return 0;
47 }
```

答案:

copy constructor of the point

copy constructor of the point

constructor of line

copy constructor of the point

copy constructor of line

the length of line2 is 2

29.请指出下列程序段中的错误，并改正：（考察前向引用）

```
1  class B;  
2  
3  class B {  
4      public:  
5          void g(A a);  
6  };  
7  
8  class A {  
9      public:  
10         void f(B b);  
11  };
```

答案: classA 没有定义，将第一行 class B 改为 class A。

30.下面图形标识在 UML 中是对什么关系的表示：（B）



A. 依赖关系

B. 作用关系

C. 包含关系

D. 继承关系

31.下列关于 c++中结构体的说法错误的是（D）

A. 结构体是由用户自定义的一种数据类型

B. 结构体中可设定若干个不同数据类型的成员

C. 结构体中成员的数据类型可以是另一个已定义的结构体

D. 结构体没有构造函数