

2020 春学期季期中练习

一、 单项选择题

1. 表达式 $(1 > 0 ? 3 : 2)$ 的值是 ()
A. 1 B. 0 C. 3 D. 2
2. 以下基本数据类型的字节数由大到小排序正确的是 ()
A. double , int , short , bool
B. double , char , int , bool
C. int , double , char , short
D. bool , char , int , short
3. C++中的标识符只能有字母、数字和下划线三种字符组成 ,且第一个字符()
A. 必须为字母
B. 必须为字母或下划线
C. 必须不是字母或下划线
D. 必须为下划线
4. 以下哪些项语法是正确的 ()
① `int a(3);`
② `char c = "car";`
③ `const float PI; PI = 3.1415926;`
④ `float f += 3.5;`
A. ① B. ①和② C. ③ D. ③和④
5. 以下表达式的值为假的是 ()
A. $(3 == 4) || (4 > 2)$
B. $(6 <= 3 + 3) || (2 >= 3)$
C. $(3 + 3 <= 6) \&\& (-6 > -4)$
D. $!(3 + 5 <= 2 + 4)$
6. 以下程序的输出为 ()

```
#include <iostream>
using namespace std;
int main(){
    int a = 1;
```

```

int b = int(1.5);
int c = int(0.5);
int out = 0;
if(a == b)
    cout << --out << endl;
else if(a == c)
    cout << out++ << endl;
else
    cout << out+2 << endl;
return 0;
}

```

A. -1 B. 0 C. 1 D. 2

7. 关于以下程序段说法正确的是 ()

```

#include <iostream>
using namespace std;
int main(){
    int a = 3;
    do{
        a -= 5;
    }while(a<0);
    return 0;
}

```

- A. 是死循环 B. 循环执行多次，但最终会退出循环
C. 循环共执行一次 D. 循环没有被执行

8. 以下语句存在语法错误的是 ()

- ① for(;;) int a = 1;
② const int i = 1; for(; i < 10; i++) int a = 1;
③ for(int i = 0; int j = 100; i <= j; i++, j--) int a = 1;
④ for(const int i = 0; i < 100;) int a = 1;

- A. ①和② B. ②和③ C. ③和④ D. ①和④

9. 以下程序的输出是 ()

```

#include <iostream>
using namespace std;
int main(){
    enum WEEKDAY {Monday, Tuesday, Wednesday=3, Thursday=1, Friday};
    cout<< Monday+Tuesday+Wednesday+Thursday+Friday <<endl;
    return 0;
}

```

A. 7 B. 5 C. 6 D. 4

10. 在下面的程序段中语句“ $s = s + 1$;”执行的总次数为 ()

```
int s = 0
for(int i = 0; i < n ; i++)
    for(int j = i ; j < n; j++)
        s = s + 1;
```

A. n 的平方 B. $n(n+1)/2$ C. $(n+1)/2$ D. $n(n-1)/2$

11. 下列哪些为错误的重载形式 ()

- ① `float func1();`
`float func2();`
- ② `float func();`
`int func();`
- ③ `string func(string c);`
`string func(string d);`
- ④ `int func(float a);`
`int func(float a, float & b);`

A. ①、②和③ B. ②、③和④ C. ①、③和④ D. ①、②和④

12. 以下函数原型错误的是 ()

- ① `int add(int x, int y = 5, int z = 6);`
- ② `int add(int x = 1, int y = 5, int z);`
- ③ `int add(int x = 1, int y, int z = 6);`

A. ①和② B. ②和③ C. ①和③ D. ①、②和③

13. 以下程序的输出为 ()

```
#include <iostream>
using namespace std;
int func1(int m) {
    return m*m;
}
int func2(int x, int y) {
    return func1(x) + func1(y);
}
int main() {
    cout << func2(3, 4) << endl;
    return 0;
}
```

A. 16 B. 9 C. 25 D. 7

14. 以下程序的输出为 ()

```
#include <iostream>
```

```
using namespace std;
int func(int x, int& y){
    int tmp = x;
    x = y;
    y = tmp;
    return x + y;
}
int main(){
    int x = 0, y = 1;
    cout << func(x,y) << " " << x << " " << y << endl;
    return 0;
}
```

A. 1 0 1 B. 1 0 0 C. 1 1 0 D. 0 1 1

15. 关于构造函数，下列说法错误的是 ()

- A. 构造函数可以重载
- B. 构造函数没有返回值
- C. 如果没有定义构造函数，则程序无法通过编译
- D. 构造函数可以设置默认参数值

16. 关于析构函数，下列说法正确的是 ()

- A. 析构函数可以有一个或多个参数 B. 一个类只能定义一个析构函数
- C. 析构函数必须有返回值 D. 如果没有定义析构函数，程序无法通过编译

17. 定义了如下的类

```
#include <string>
using namespace std;
struct Contact {
    long long tel;
    string mail;
    Contact(long long _tel, string _mail) {
        tel = _tel, mail = _mail;
    }
};
class Student {
    int id;
    string name;
    int grade;
    Contact info;
public:
    Student(int _id, string _name, int _grade, long long _tel,
string _mail) : info(_tel, _mail) {
        id = _id, name = _name, grade = _grade;
    }
};
```

```
int main() {
    Student a(1, "Alice", 95, 10010, "alice@xxx.com");
    return 0;
}
```

那么想要在 main 函数中修改学生的成绩(grade)和电话(tel)的信息，下列做法正确的是 ()

A. 直接在 main 函数中使用如下语句即可

```
a.grade = 100, a.info.tel = 15500001111;
```

B. 可以在 Student 类中增加公有的 setGrade 函数

```
void setGrade(int newGrade) {
    grade = newGrade;
}
```

然后在 main 函数中使用如下语句可以实现功能

```
a.setGrade(100), a.info.tel = 15500001111
```

C. 可以在 Student 类中增加公有的 setGrade 和 setTel 函数

```
void setGrade(int newGrade) {
    grade = newGrade;
}
void setTel(long long newTel) {
    info.tel = newTel;
}
```

然后在 main 函数中使用如下语句可以实现功能

```
a.setGrade(100), a.setTel(15500001111)
```

D. 以上做法都不行，必须在 Contact 结构体中增加公有的 setTel 函数

```
void setTel(long long newTel) {
    tel = newTel;
}
```

然后在 Student 类中增加公有的 setGrade 和 setTel 函数

```
void setGrade(int newGrade) {
    grade = newGrade;
}
void setTel(long long newTel) {
    info.setTel(newTel);
}
```

然后在 main 函数中使用如下语句才可以实现功能

```
a.setGrade(100), a.setTel(15500001111)
```

18-20 题总体要求：观察以下程序中对整数封装成的 Integer 类，并回答 18-20 题

```
#include<iostream>
using namespace std;
```

```

class Integer {
private:
    int val;
public:
    Integer(int _val = 0): val(_val) {
        cout << "Initialize with value " << _val << endl;
    }
    Integer(const Integer& c) {
        val = c.val;
        cout << "Initialize with object " << c.val << endl;
    }
    ~Integer() {
        cout << "Delete an object." << endl;
    }
    void increase() {
        val ++;
    }
    int getDigit() {
        int n = val;
        int ret = 0;
        while(n > 0) {
            n /= 10;
            ret ++;
        }
        return ret;
    }
};

```

18. 若 main 函数的实现如下

```

int main() {
    Integer a(3), b(5);
    a.increase(), b.increase();
    Integer c = a;
    Integer& d = b;
    cout << "End of program" << endl;
    return 0;
}

```

请问程序的输出应为 ()

- A. Initialize with value 3
Initialize with value 5
Initialize with object 4
Initialize with object 6
End of program
Delete an object.
Delete an object.
Delete an object.
Delete an object.
- B. Initialize with value 3
Initialize with value 5

- Initialize with object 4
End of program
Delete an object.
Delete an object.
Delete an object.
- C. Initialize with value 3
Initialize with value 5
Initialize with object 4
Initialize with object 6
Delete an object.
Delete an object.
Delete an object.
Delete an object.
End of program
- D. Initialize with value 3
Initialize with value 5
Initialize with object 4
Delete an object.
Delete an object.
Delete an object.
End of program
19. 若 main 函数的实现同“18”题，请问对象的析构顺序为 ()
- A. d-c-b-a
B. a-b-c-d
C. c-b-a
D. a-b-c

20. 若 main 函数实现如下

```
int main() {
    Integer a(9), b;
    a.increase();
    b = a;
    cout << b.getDigit() << endl;
}
```

则程序的输出应为 ()

- A.
Initialize with value 9
Initialize with value 0
2
Delete an object.
Delete an object.
- B.
Initialize with value 9
Initialize with object 9
1
Delete an object.

Delete an object.

C.

Initialize with value 9

Initialize with object 10

2

Delete an object.

Delete an object.

D.

Initialize with value 9

Initialize with value 0

Initialize with object 10

2

Delete an object.

Delete an object.

Delete an object.

21. 定义如下客户类 Client

```
#include<iostream>
```

```
using namespace std;
```

```
class Client {
```

```
    public:
```

```
        static int count; //客户总数
```

```
        int id; //客户编号
```

```
        Client(int _id): id(_id) {
            count ++;
```

```
        }
```

```
        ~Client() {
            count --;
```

```
        }
```

```
};
```

```
int Client::count = 0;
```

```
int main() {
```

```
    Client a(1);
```

```
    cout << a.id << endl; //①
```

```
    cout << Client::id << endl; //②
```

```
    cout << a.count << endl; //③
```

```
    cout << Client::count << endl; //④
```



```
    return 0;
}
```

那么在 main 函数中的语句不能通过编译的是 ()

- A. ① B. ② C. ③ D. ④

22. 以下关于静态成员的描述中，错误的是 ()

- A. 类的所有对象共享静态成员
- B. 非 const 静态数据成员要在类外初始化
- C. 只有静态成员函数可以访问静态数据成员
- D. 在没有定义任何对象时，可以通过类名调用静态成员函数

23. 已知二维平面上的直线可以由点和方向的线性组合表示，于是在如下程序中定义了二维平面上的 Point 类和 Direction 类，以及 Line 类

```
#include <iostream>
using namespace std;

class Point {
    double x;
    double y;
public:
    Point (double _x, double _y): x(_x), y(_y) {
        cout << "New point: (" << x << "," << y << ")" << endl;
    }
};

class Direction {
    double x;
    double y;
public:
    Direction (double _x, double _y): x(_x), y(_y) {
        cout << "New direction: (" << x << "," << y << ")" <<
endl;
    }
};

class Line {
    Point p;
    Direction d;
public:
    Line (double px, double py, double dx, double dy): d(dx, dy),
p(px, py) {
        cout << "New line: (" << px << "," << py << ")";
        cout << " + k(" << dx << "," << dy << ")" << endl;
    }
}
```

```
};

int main() {
    Line line(0.0, 0.0, 1.0, 1.0);
    return 0;
}
```

程序的输出为 ()

A.

New line: $(0,0) + k(1,1)$

New point: $(0,0)$

New direction: $(1,1)$

B.

New line: $(0,0) + k(1,1)$

New direction: $(1,1)$

New point: $(0,0)$

C.

New point: $(0,0)$

New direction: $(1,1)$

New line: $(0,0) + k(1,1)$

D.

New direction: $(1,1)$

New point: $(0,0)$

New line: $(0,0) + k(1,1)$

24. 对于以下 Circle 类 (省略部分成员函数和成员变量)

```
const double pi = 3.14;
class Circle {
    double r;
    //其他数据成员略...
public:
    double biggerSquare (double R) const;
    double preciseSquare () const;
    double approximateSquare() const;
    //其他函数成员略...
};
```

若要在类外实现常成员函数, 下列函数定义中, 没有语法错误的是 ()

A.

```
double Circle::biggerSquare (double R) const {
    r = R > r ? R : r;
    return pi * r * r;
}
```

B.

```
double Circle::biggerSquare (double R) const {
    R = R > r ? R : r;
    return pi * R * R;
}
```

C.

```
double Circle::preciseSquare () const {
    pi = 3.1415926;
    return pi * r * r;
}
```

D.

```
double Circle::approximateSquare() const {
    pi = 3.0;
    return pi * r * r;
}
```

25. 观察以下程序

```
#include <iostream>
using namespace std;

int x = 0, y = 0;

void func(int x, int y) {
    cout << "(" << x << ", " << y << ")";
    x = 1, y = 1;
}

int main() {
    cout << "(" << x << ", " << y << ")";
    func(10, 10);
    cout << "(" << x << ", " << y << ")";
    return 0;
}
```

其输出结果应为 ()

- A. (0,0)(10,10)(1,1)
- B. (0,0)(0,0)(1,1)
- C. (0,0)(10,10)(0,0)
- D. (0,0)(0,0)(10,10)

26. 关于友元函数和友元类，下列说法错误的是 ()

- A. 友元函数破坏了类的封装性
- B. 友元函数不属于类的成员函数

- C. 友元函数可以访问公有成员和私有成员，但不能访问保护成员
 D. 一个类的友元类的成员函数全部都是这个类的友元函数

二、填空题 (共 14 空)

27. 补全函数 `void func(int i ,int N);`

其中 $i \leq N$ ，功能为输出 i 递增到 N 再递减到 i 的整数，每行输出一个数。

示例：`func(1,4)`

输出：

```
1
2
3
4
3
2
1
```

```
#include <iostream>
using namespace std;

void func(int i, int n){
    if (____【27-1】____) {
        cout << i << endl;
        return;
    }
    cout << i << endl;
    ____【27-2】____;
    cout << i << endl;
}

int main(){
    func(1,4);
    return 0;
}
```

28. 编写函数求两个整数的最大公约数

```
#include <iostream>
using namespace std;
int func(int i, int j){
    int temp;
    if(i < j) { //如果 i 比 j 小，交换 i 与 j 的值。
        temp = i;
        ____【28-1】____;
    }
}
```

```

        ____【28-2】____;
    }
    while (j != 0) {
        temp = i % j;
        i = j;
        j = temp;
    }
    return ____【28-3】____;
}
int main(){
    int i,j;
    cout << "请输入一个正整数：";
    cin >> i;
    cout << "请输入另一个正整数： ";
    cin >> j;
    cout << i << "和" << j << "的最大公约数是：" << func(i,j);
    return 0;
}

```

29. 程序补全

定义如下的二维平面上的向量类，请按照注释部分的提示补全空白处的代码。

```

1  #include <iostream>
2  using namespace std;
3
4  class Vector {
5      private:
6          double x;
7          double y;
8      public:
9          static int count;
10         Vector(double _x, double _y): x(_x), y(_y) {
11             count ++;
12         }
13
14         Vector(const ____【29-1】____) { //复制构造函数
15             x = c.x, y = c.y;
16             count ++;
17         }
18
19         ~Vector() {
20             count --;
21         }
22
23         void show();
24         void add(const Vector&);

```

```
25 };
26
27 【29-2】; //初始化静态成员变量 count，初值为 0
28
29 void Vector::show() {
30     cout << "(" << x << ", " << y << ")" << endl;
31 }
32
33 【29-3】{ //类外定义 24 行 add 函数
34     x += c2.x;
35     y += c2.y;
36 }
37
38 void work(Vector c1, Vector c2) {
39     c1.add(c2);
40     c1.show();
41     cout << Vector::count << endl;
42 }
43
44 int main() {
45     Vector c1(1.5, 2.5), c2(4.5, 0);
46     work(c1, c2);
47     c1.show();
48     cout << Vector::count << endl;
49     return 0;
50 }
```

30. 程序分析

运行第 29 题中的程序，填写执行完下列行的输出结果。

- 1) 第 40 行 show 函数的输出结果为 【30-1】。
- 2) 第 41 行的输出结果为 【30-2】。
- 3) 第 47 行 show 函数输出结果为 【30-3】。
- 4) 第 48 行的输出结果为 【30-4】。