

上海交通大学试卷(B卷)

(2020 至 2021 学年 第1 学期)

班级号 _____ 学号 _____

姓名 _____

课程名称 _____ 程序设计思想与方法 (C++) _____

成绩 _____

我承诺, 我将严格遵守考试纪律。

承诺人: _____

题号				
得分				
批阅人(流水阅卷教师签名处)				

一. 单项选择题 (每题 1 分, 共 15 分)

() 1. 在同一工程文件中, 一个文件想要共享使用在另外一个文件中声明的一个全局变量, 则需要在本文件声明该全局变量前加限定词 _____
A. register B. auto C. extern D. static

() 2. 有如下的函数定义:

```
int Xfun(int x)
```

```
{
```

```
int y=x;
```

```
{int x=10; y+=x;}
```

```
return x+y;
```

```
}
```

通过表达式 Xfun(5)调用该函数, 则得到的返回值是 _____

A. 10 B. 15 C. 20 D. 25

() 3. 关于 switch 语句, 下列说法正确的是 _____

- A. 可以没有 default 分支。
- B. case 分支中的语句可以是多个语句, 此时必须要有大括号。
- C. case 分支中必须要有 break 语句
- D. case 语句后的表达式可以是变量。

() 4. 关于类的构造和析构函数, 以下哪个说法是正确的?

- A. 类的一个对象可以多次运行构造函数
- B. 类的构造和析构函数都支持重载
- C. 声明一个类的指针变量, 系统会调用不带参数的构造函数
- D. 用户不定义类的构造函数和析构函数系统也会提供

() 5. 每个 C++ 程序必须有且仅有一个 _____

- A. 主函数 B. 语句 C. 函数 D. 预处理指令

B 卷总 14 页 第 1 页

() 6. 若有以下定义: char str1[8], str2[8] = "sjtu"; 则以下选项中 _____ 不能使 cout << str1 的输出为 sjtu

- A. strcpy(str1, str2);
- B. strncpy(str1, str2, 6);
- C. str1 = str2;
- D. str1[0] = 0; strcat(str1, str2);

() 7. 若函数的参数是数组, 则传递给这个参数的值是 _____

- A. 数组第一个元素的值
- B. 数组元素的个数
- C. 整个数组被拷贝过去
- D. 数组的首地址

() 8. 如何检查字符变量 a 的内容是否为字母?

- A. a>='a' || a<='z' || a>='A' || a<='Z'
- B. a>='a' && a<='z' || a>='A' && a<='Z'
- C. a>='a' && a<='z' && a>='A' && a<='Z'
- D. a>='a' || a<='z' && a>='A' || a<='Z'

() 9. 给出下面定义:

```
char a[] = "abcd";
```

```
char b[] = {'a', 'b', 'c', 'd'};
```

下列说法正确的是 _____

- A. 数组 a 和数组 b 占用空间大小相同
- B. 数组 a 与数组 b 等价
- C. 数组 a 占用空间小于数组 b
- D. 数组 a 占用空间大于数组 b

() 10. 已定义数组 float c[10] 和数组 double d[10], 且已定义并初始化指针 float *pc = &c[1] 和指针 double *pd = &d[1], 那么指向 c[5] 和 d[5] 的指针分别应为 _____

- A. pc+16 和 pd+32
- B. pc+16 和 pd+16
- C. pc+5 和 pd+5
- D. pc+4 和 pd+4

() 11. 以下有关模块化开发说法正确的是 _____

- A. 各个模块可以单独编译, 最后再链接在一起构成可执行程序
- B. 在不同的模块中不能定义同名的静态全局变量

B 卷总 14 页 第 2 页





D. 除了系统自动生成的成员函数, B 类有 2 个访问特性不相同的成员函数

二. 给出下列程序段的运行结果 (请将结果写在程序右侧空白处, 每题 4 分, 共 32 分)

```
1.
#include <iostream>
using namespace std;
int main() {
    int i;
    double d;
    i = d = 3.5;
    cout << "i = " << i << " " << "d = " << d << endl;
    d = i = 3.5;
    cout << "d = " << d << " " << "i = " << i << endl;
    return 0;
}
```

```
2
#include <iostream>
using namespace std;
void swap(int arr[], int i, int j){
    int tmp=arr[i];
    arr[i]=arr[j];
    arr[j]=tmp;
}
int main(){
    int a[6]={8,4,10,12,6,15};
    int b[6]={1,4,2,0,5,3};
    for (int i=0;i<6;i++) {
        for (int j=0;j<6;j++) {
            if (b[j]==i) {
                swap(a,i,j);
                swap(b,i,j);
                break;
            }
        }
    }
    for(int i=0;i<6;i++) cout<<a[i]<<" ";
    return 0;
}
```

- C. 在一个模块中定义的函数不能被其他模块调用
- D. 无法访问在其他模块中定义的全局变量

() 12. 有以下函数定义:

```
void Fun(int n, double x)
{ ..... }
```

若以下选项中的变量都已正确定义并赋值如下:

```
int a,k;
double b;
a=12;
b=0.45;
```

则对函数 Fun 的正确调用语句是_____

- A. Fun(int y,double m);
- B. k=void Fun(a,b);
- C. k=Fun(10,12.5);
- D. Fun(a,b);

() 13. 下面哪个说法不正确?

- A. 静态数据成员物理上不属于任何对象
- B. 类的成员函数都有一个隐含的 this 指针
- C. 一个类的静态常量数据成员是类一级的常量
- D. 非静态的常量数据成员在类的构造函数的初始化表中进行初始化

() 14. 关于友元, 以下哪个说法不正确?

- A. 类的友元可以有多个
- B. 类的友元也可以访问其 protected 成员
- C. 类的友元可以访问类中私有成员
- D. 类的友元必须是一个全局函数

() 15. 有如下类的定义:

```
class A
{
    int x;
public:
    void set(int k);
}
```

```
class B:public A
{
    int y;
public:
    void sety(int k);
}
```

则_____是正确的

- A. B 类有 2 个访问特性不相同的数据成员
- B. B 类只有 1 个数据成员, 其访问特性为 private
- C. 除了系统自动生成的成员函数, B 类只有 1 个公有的成员函数

3

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

```
int main(){
    int a=5,b,n=6;
    do
    {
        b=1;
        while(b < a)
        {
            cout << a << "a";
            b=b+1;
        }
        cout << endl;
        a=a+1;
    } while(a<=n);
    return 0;
}
```

4

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

```
void myfun(int *q, int &n) {
    cout << *q << endl;
    q = &n;
    cout << *q << endl;
}
```

```
int main() {
    int x = 0, y = 1, z = 2;
    int *p = &x;
    myfun(p, y);
    cout << *p << endl;
    myfun(&z, *p);
    return 0;
}
```

5

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

```
void Bin(int x)
{
    if (x/2 > 0)
        Bin(x/2);
    cout << x%2;
}
```

```
int main()
{
    Bin(12);
    return 0;
}
```

6

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

```
int i = 0;
void fun()
{
    static int i = 1;
    cout << i++ << ' ';
}
cout << i << ' ';
```

```
int main()
{
    fun();
    fun();
    return 0;
}
```





```

7
#include <iostream>
using namespace std;
class Dummy
{
    int objectID;
public:
    Dummy(int ID)
    {
        objectID = ID;
        cout << "Constructor " << objectID << endl;
    }
    Dummy(const Dummy &source)
    {
        objectID = source.objectID;
        cout << "Copy constructor " << objectID << endl;
    }
    ~Dummy()
    {
        cout << "Destructor " << objectID << endl;
    }
};

```

```

Dummy obj0(0);
void f1(Dummy aDummy)
{
    static Dummy obj1(aDummy);
}

```

```

void f2(Dummy& aDummy)
{
    Dummy obj1(aDummy);
}

```

```

int main()
{
    Dummy obj2(2),obj3(3);
    f1(obj2);
    f2(obj3);
    obj2 = obj3;
    return 0;
}

```

```

8
#include <iostream>
#include <cstring>
using namespace std;
class Person
{
    private:
        char name[10];
        int age;
        static int count;
    public:
        Person(char *str="sjtu", int n=104);
        ~Person(){count--;cout<<"name="<<name<<" count="<<count<<endl;}
};
int Person::count=10;

Person::Person(char *str, int n)
{
    strcpy(name, str);
    age = n;
    count++;
}

int main()
{
    Person personArr[2], *p;
    p= new Person("pku", 102);
    Person &q=*p;
    delete p;
    return 0;
}

```

三. 程序填空 (每空 2 分, 共 14 分)

1 以下程序定义了分数这种结构体, 并定义了一个可以对分数进行化简的函数, 程序输出为 3/8

```

#include <iostream>
using namespace std;

```

```

struct Fraction
{

```



```

int numerator;
int denominator;
};

void reduce(____ f)
{
    int min,i;
    min = f->numerator;
    if (min > f->denominator)
        min = f->denominator;
    for (i = 1; i <= _____; i++){
        if (____){
            f->numerator = f->numerator / i;
            f->denominator = f->denominator / i;
            min = min / i;
        }
    }
    cout << f->numerator<<'/'<< f->denominator << endl;
}

int main()
{
    Fraction a = { 24,64 };
    reduce(____);
    return 0;
}

```

2 以下程序定义了一个 Goods 类，期望输出
Weight: 8 Total weight: 188
Total weight is 188

```

#include <iostream>
using namespace std;

class Goods
{
    const int weight;
    static int total_weight;
public:

```

```

        {total_weight+=w;} //构造函数
~Goods(){total_weight-=weight;} //析构函数
int getWeight()const{return weight;}
static int totalWeight(){return total_weight;}
};

void display(____)
{
    cout<<"Weight: "<<g.getWeight()<<" Total weight: "<<g.totalWeight()<<endl;
}

int main()
{
    Goods g1(8),g2(80),g3(100);
    display(g1);
    cout<<"Total weight is "<<Goods::totalWeight()<<endl;
    return 0;
}

```

四. 编程题 (共 39 分)

1 编写一个程序，输入正整数，表示一段时间的长度，单位为秒，输出这段时间有多少小时多少分多秒。
例如输入 10000，则输出为 2:46:40，即 2 小时 46 分 40 秒。(4 分)



2 有 n 盏灯, 编号为 $1 \sim n$ 。第 1 个人把所有灯打开, 第 2 个人按下所有编号为 2 的倍数的开关 (这些灯将被关掉), 第 3 个人按下所有编号为 3 的倍数的开关 (其中关掉的灯将被打开, 开着的灯将被关闭), 依次类推。一共有 k 个人, 问最后有哪些灯开着? 编写一个程序, 输入 n 和 k , 输出开着的灯的编号 ($k \leq n \leq 1000$)。 (10 分)

3 设计一个程序, 先让用户输入一个整数 $length$, 然后让用户输入一行长度不超过 $length$ 的句子。已知该句子中只包含空格以及由字母组成的单词, 单词的个数不超过 10 个, 句子的首字符和末尾字符都不是空格, 相邻两个单词之间有且只有一个空格。要求该程序能将句子中的所有单词按其字母数量从少到多重新排序, 然后输出排序后的一行新句子, 相邻两个单词之间有且只有一个空格。 (10 分)

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

int main() {
    cout << "Sentence Length: ";
    int length;
    cin >> length;
    cin.get(); //remove '\n'
    char *inputs = new char[length+1];
    char *outputs = new char[length+1];
    cout << "Input Sentence: ";
    cin.getline(inputs, length+1);
    //你的代码
    cout << "Output Sentence: " << outputs;
    delete []inputs;
    delete []outputs;
    return 0;
}
```

运行示例:

Sentence Length: 19

Input Sentence: aaaaa bb cccc d eee

Output Sentence: d bb eee cccc aaaaa





4 设计一个字符串类 myString, 可以求字符串长度, 可以连接两个串 (如, s1="abcd", s2="12", s1 与 s2 连接得到 "abcd12"), 运行以下主程序, 获得 s1 和 s2 的连接结果和连接后的字符长度。不能使用 string 类 (15 分)

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

int main()
{
    myString str1("I am "), str2("a student.");
    str1.stringCat(str2);
    str1.disp();
    cout<<"Length of str1: "<<str1.getLen()<<endl;
    return 0;
}
```

运行结果为

I am a student.

Length of str1: 15