

北京科技大学 2020-2021 学年 第 一 学期
程序设计基础 C++ 期末试卷（模拟）

院（系）_____ 班级_____ 学号_____ 姓名_____

一、选择题（每小题 2 分，共 20 分）

1. 下列符号中可以用作 C++ 标识符的是_____
A. _cyberpunk_2077
B. Left4~dead2
C. else
D. 4399seer
2. 下面关于构造函数和析构函数的描述中，正确的是_____
A. 构造函数可以带有返回值，而析构函数不能带返回值
B. 构造函数的名字与类名完全相同，在对象创建的时候执行。
C. 构造函数必须带有参数，而析构函数不能带有参数。
D. 构造函数和析构函数都必须定义，不能缺省。
3. 下列关于虚函数的说明中，正确的是_____
A. 虚函数实现了 C++ 的多态机制
B. 虚函数可以是静态成员函数
C. 只能通过指针和引用调用虚函数
D. 抽象类中的成员函数都是虚函数
4. 假定一条定义语句为“int a[10], x, *p=a;”，若要把数组 a 中下标为 3 的元素值赋给 x，
不正确的语句为_____
A. x=p[3];

B. `x=*(a+3);`

C. `x=a[3];`

D. `x=*p+3;`

5. 下列定义了一个指向数组的指针 p1 的是_____

A. `int *p1[3]`

B. `int (*p1)[3]`

C. `(int *)p1[2]`

D. `int *p1[]`

6. 一个类的构造函数为: `C(int x, int y){a=x--; b=a*y--;}`, 则执行 `C c(3,5);` 语句后, x.a 和 x.b 的值分别为_____

A. 3 和 5

B. 5 和 3

C. 3 和 15

D. 20 和 5

7. 下面有关 new 和 delete 操作符的说法中, 不正确的是_____

A. 使用 new 操作符, 可以动态分配全局堆中的内存资源

B. 若 P 的类型已由 A 强制转换为 void*, 那么执行语句 `delete P*` 时, 类 A 的析构函数不会被调用

C. 实现全局函数时, new 和 delete 通常成对地出现在由一对匹配的花括号限定的语句块中

D. 执行语句 `A *P=new A[100]` 时, 类 A 的构造函数只会被调用一次

8. 关于静态数据成员, 下面叙述不正确的是_____

A. 使用静态数据成员, 实际上是为了消除全局变量

B. 可以使用“对象名.静态成员”或者“类名::静态成员”来访问静态数据成员



- C.静态数据成员只能在静态成员函数中引用
- D.所有对象的静态数据成员占用同一内存单元

9. 有以下程序段

```
int main() {  
    char a[8]={1,2,3,4,5,6,7,8};  
    int *ptr=(int*)a;  
    ++ptr;  
    return 0;  
}
```

当++ptr 语句执行完毕后， ptr 的值等同于_____

- A. &a[1];
 - B. &a[2];
 - C. &a[3];
 - D. &a[4];
10. 假定 C 是一个类，那么执行语句“C a[3],b(5),*p;”调用了_____次构造函数
- A.1
 - B.2
 - C.3
 - D.4

北京科技大学学生学习与发展指导中心
Center for Student Learning and Development USTB

二、判断题（每小题 2 分，共 20 分）

1. 类中定义的成员默认为 public 访问属性。（ ）
2. 通过函数来实现一种不太复杂的功能，并且要求加快执行速度时，选用内联函数最合适。（ ）
3. 友员函数是独立于当前类的外部函数，一个友员函数不可以同时定义为两个类的友员函数。（ ）

4. 假定 AB 为一个类,则执行 AB x;语句时将自动调用该类的无参构造函数。
()
5. 派生类的对象对它的基类成员中私有继承的公有成员是可以访问的。 ()
6. 面向对象的程序设计有四大特征, 它们是抽象、封装、继承、多态。 ()
7. 静态数据成员在类外进行初始化, 且静态数据成员的一个拷贝被类的所有对象共享。 ()
8. 抽象类中可以不含有虚函数, 但抽象类只能被继承, 不能被实例化为对象。
()
9. 当创建一个对象后, this 指针就指向该对象, 此时所有的成员函数都共享这个 this 指针。 ()
10. 拷贝构造函数使用引用作为参数初始化创建中的对象。 ()

三、编程题 (共 20 分)

1. 统计字符串出现次数 (分值: 10.00)

【问题描述】

话说这个世界上有各种各样的兔子和乌龟, 但是研究发现, 所有的兔子和乌龟都有一个共同的特点——喜欢赛跑。于是世界上各个角落都不断在发生着乌龟和兔子的比赛, 小华对此很感兴趣, 于是决定研究不同兔子和乌龟的赛跑。他发现, 兔子虽然跑比乌龟快, 但它们有众所周知的毛病——骄傲且懒惰, 于是在与乌龟的比赛中, 一旦任一秒结束后兔子发现自己领先 t 米或以上, 它们就会停下来休息 s 秒。对于不同的兔子, t , s 的数值是不同的, 但是所有的乌龟却是一致——它们不到终点决不停止。

然而有些比赛相当漫长, 全程观看会耗费大量时间, 而小华发现只要在每场比赛开始后记录下兔子和乌龟的数据——兔子的速度 v_1 (表示每秒兔子能跑 v_1 米), 乌龟的速度 v_2 , 以及兔子对应的 t , s 值, 以及赛道的长度 l ——就能预测出比赛的结果。但是小华很懒, 不想通过手工计算推测出比赛的结果, 于是他找到了你——清华大学计算机系

的高才生——请求帮助，请你写一个程序，对于输入的一场比赛的数据 v_1 , v_2 , t , s , l ，预测该场比赛的结果。

【输入形式】

输入只有一行，包含用空格隔开的五个正整数 v_1 , v_2 , t , s , l ，其中($v_1, v_2 <= 100; t <= 300; s <= 10; l <= 10000$ 且为 v_1, v_2 的公倍数)

【输出形式】

输出包含两行，第一行输出比赛结果——一个大写字母“T”或“R”或“D”，分别表示乌龟获胜，兔子获胜，或者两者同时到达终点。

第二行输出一个正整数，表示获胜者(或者双方同时)到达终点所耗费的时间(秒数)。

【样例输入 1】

10 5 5 2 20

【样例输出 1】

D

4

【样例输入 2】

10 5 5 1 20

【样例输出 2】

R

3

2. 文字图形 (分值: 10.00)

【问题描述】

利用字母可以组成一些美丽的图形，下面给出了一个例子：

```
ABCDEFGG  
BABCDEF  
CBABCDE  
DCBABCD  
EDCBABC
```

这是一个 5 行 7 列的图形，请找出这个图形的规律，并输出一个 n 行 m 列的图形。

【输入形式】

输入一行，包含两个整数 n 和 m，分别表示你要输出的图形的行数的列数。

【输出形式】

输出 n 行，每个 m 个字符，为你的图形。

【样例输入 1】

5 7

【样例输出 1】

```
ABCDEFGG  
BABCDEF  
CBABCDE  
DCBABCD  
EDCBABC
```

【数据规模与约定】

$1 \leq n, m < 26$

四、程序片段编程题（共 20 分）

1. 统计字符串出现次数（分值：10.00）

【问题描述】

编程题：统计字符串中每个字符出现的次数，按照字符在字符串中的先后顺序输出结果。统计时不区分大小写，均按小写字母计。请按照要求将程序补充完整。

【样例输入】

UniversityofScienceTechnologyBeijing

【样例输出】

u1,n4,i5,V1,e5,r1,s2,t2,y2,03,f1,s2,c3,h1,11,g2,b1,i1,

【程序代码片段】

```
#include <iostream>
#include <string.h>
using namespace std;
void fun(char str[], int n);
int main(){
    char str[100]="UniversityofScienceTechnologyBeijing";
    cout<<strlen(str)<<endl;
    /*调用函数统计并输出结果*/
    fun(str,strlen(str));
    return 0;
}
void fun(char str[], int n){
    /*设计fun函数，实现题目所要求的功能*/
}
```

2. 圆桌问题 （分值：10.00）

【问题描述】

有一个圆类 circle 和一个桌子类 table，另设计一个圆桌类 roundtable，它是从前两个类派生的，要求输出一个圆桌的高度、面积和颜色等数据。 circle 类包含公有数据成员 radius 和求圆面积的成员函数 getarea(); table 类包含公有数据

成员 height 和返回高度的成员函数 getheight()。roundtable 类继承所有上述类的数据成员和成员函数，添加了私有数据成员 color 和相应的成员函数。

主函数中读取圆桌类的桌面面积、高度和颜色，并展示出来。

【样例输入】

```
1 5.8 red
```

【样例输出】

```
3.14
5.8
red
```

【程序代码片段】

```
#include<iostream>
#include<string.h>
const double pi=3.14; //定义常量pi
using namespace std;
class circle
{
    //写成纯虚函数
    //补全代码区1
};
class table
{
    //补全代码区2
};
class roundtable:public circle,public table
{
    //补全代码区3
};
int main()
{
    //补全代码区4
    return 0;
}
```


五、程序填空题（共 20 分）

1. 填写空白，使得输出的结果是

B::f()5 6

D::f() 10

```
#include <iostream>
using namespace std;
class B
{
public:
    /*填空 1_____*/
    {
        cout << "B::f()" << i << " " << j << endl;
    }
    B(int int1, int int2) : j(int2), i(int1)
    {
    }
private:
    int i;
    int j;
};

class D : /*填空 2_____*/
{
public:
    D(int anInt = 10, int bi = 5, int bj = 6) : /*填空
3_____*/ {}
    void f() const
    {
        B::f();
        cout << "D::f()" << i << endl;
    }
private:
    int i;
};

int main()
{
```

```

        B *p = new D;
        p->f();
    }

```

2. 填写空白，使得函数 yhtriangle 输出杨辉三角

```

#include<iostream>
using namespace std;
void yhtriangle(int* const, int);
int main()
{
    int n, *yh;
    cin >> n;
    yh = new int;           //创建动态数组
    yhtriangle(yh,n);       //调用函数，输出n次幂的系数表
    delete []yh;           //释放动态数组
    yh = NULL;
}

void yhtriangle(int* const py, int pn)
{
    int i, j, k;
    py[0] = 1;              //对第一项赋值1
    cout << py[0] << endl;  //输出0次幂系数
    for (i = 0; i < pn + 1; i++)
    {
        py[i+1] = 1;        //每行的最后一项赋值1
        for (/*填空1_____*/ /*迭代计算当前行各项值
            /*填空2_____*/
        for (k = 0; k <= i+1; k++) //输出第i次幂系数
            /*填空3_____*/
        cout << endl;
    }
}

```

3. 填写空白，使得程序能够判断输入 k 个数字是否是素数

```

#include <cmath>
using namespace std;
int main()

{

```

```

int i, j, k, m, n;
cin >> k;
for (j = 0; j < k; j++)
{
    cin >> m;
    n = sqrt(m);
    for (i = 2; i < n; i++)
        if (/*填空1_____*/
            /*填空2_____*/
            if (/*填空3_____*/
                cout << m << " is a prime number! " << endl;
            else
                cout << m << " is not a prime number!" << endl;
        }
    return 0;
}

```

4. 填写空白，使得函数将输入字符串倒序

```

#include <cstring>
#include "stdio.h"
using namespace std;
void reverseString(char* a, int l)
{
    int i;
    char temp;
    for (i = 0; i <= /*填空1_____*/; i++)
    {
        /*填空2_____*/
        a[i] = a[l - 1 - i];
        a[l - 1 - i] = temp;
    }
}

int main()
{
    char a[10];
    gets(a);
    reverseString(a, strlen(a));
    cout << a;
    return 0;
}

```