编号：1-1

山东师范大学
SHANDONG NORMAL UNIVERSITY

信息科学与工程学院实验报告

《面向对象程序设计》

Object-Oriented Programming

姓名：	姬彬荃
学号：	201911020125
班级：	计联培 1901
时间：	2020 年 10 月 13 日

《面向对象程序设计》实验报告

基本要求：请围绕实验目的、实验内容、实验过程、实验结果（附图）、实验总结（重点阐述）五个部分进行撰写。若报告中若涉及源代码内容，请在附录部分提供完整源码及 GitHub 源码托管地址。报告撰写完毕后请提交 PDF 格式版本到云班课。

一、实验目的

- 理解 c++对 C 的各项改进和扩展基本原理
- 熟练运用 C++特色函数解决实际问题
- 理解并掌握 C++指针和引用的本质机理
- 熟练掌握 C++动态内存申请和释放方法
- 掌握 visual studio 代码调试方法

二、实验内容

【编程设计题】 给定 m 根木棍，每根木棍的长度记为 L_i,(3<= i <=m)，下面欲从这 m 根木棍中选择 3 根木棍组成周长尽可能最长的三角形、面积尽可能最大的三角形，分别输出最大的周长和面积。如果怎么选都无法构成三角形，请直接输出 0。

要求：

1. 算法具有良好的可读性、稳健性和通用性（适合整数长度，浮点数长度）。
2. 给出算法的复杂度分析，算法复杂度尽可能越低越好。
3. 算法设计时采用指针，引用，重载函数，及动态内存申请等 C++核心特性。

输入：

```
m  = 5
L  = 2 3 4 5 10
```

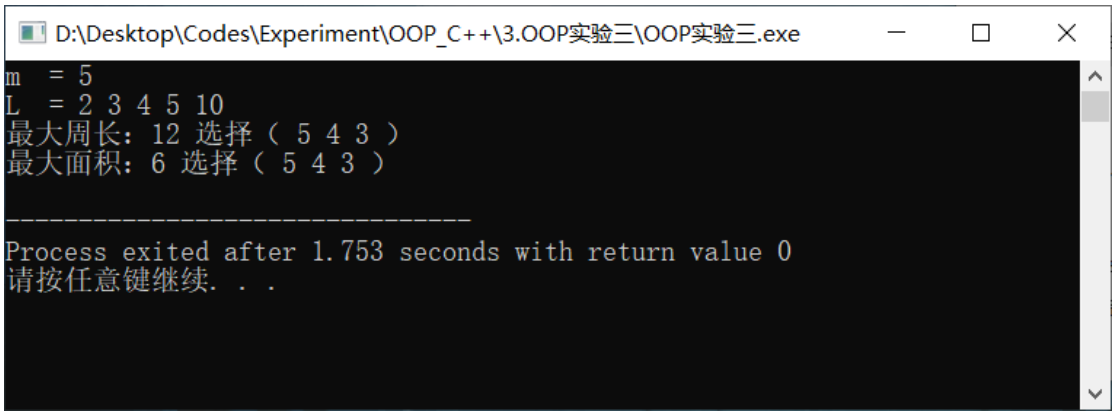
输出：最大周长 12，最大面积 6（选择 3，4，5）

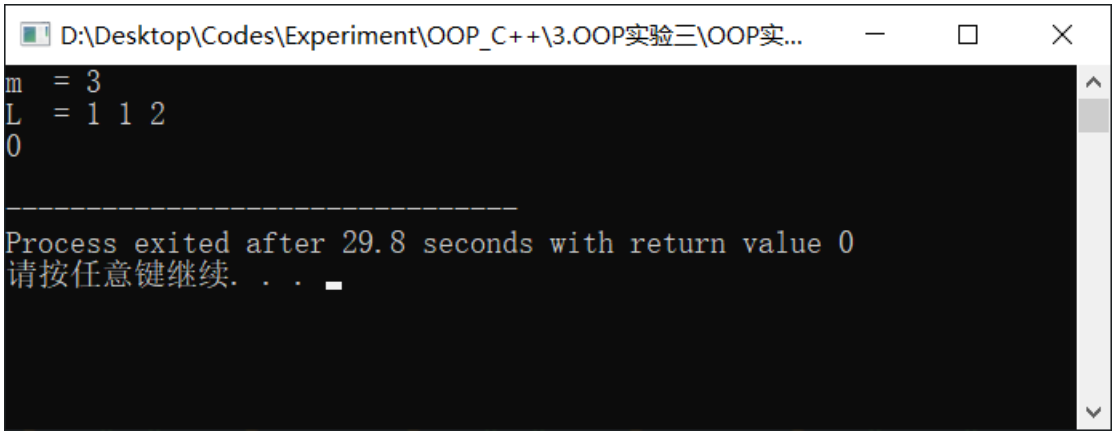
三、实验过程

实验环境：Dev C++

在按照题目要求格式读入数据后先对所有木棍长度从大到小进行排序，从前到后三个三个遍历木棍，易知周长最大的三角形是第一个合法构成的三角形。又可证明，在所有木棍中，面积最大的三角形其三边长度一定是长度连续的三个木棍组成，故再次遍历木棍，计算每三个木棍组成的合法三角形的面积。找到最大面积，记录最大面积的木棍编号。最后输出即可。

四、实验结果





（1）例 2.1

```
A 25 109.356
c=A   a=90   f=289.4

--------------------------------
Process exited after 7.067 seconds with return value 0
请按任意键继续. . .
```

分析：将输出改为如下格式

```
cout<<"c="<<c<<"  a="<<a<<"  f="<<fixed<<setprecision(1)<<f<<endl ;      //输出变量c、a、f的值
```

（2）例 2.4

报错

```
17      x=-100 ;                    //直接访问，相当于two::x=-100;
18      cout<<inf<<endl;            //using声明使用了整个one，其所有
19      cout<<M<<endl;
20      two::inf*=2;                //方法2: 使用名字空间名::局部内
21      cout<<two::inf<<endl;     //同样是two中的内容，但是访问方式不
22      cout<<x<<endl ;           //已用using声明了two中内容x，可以直
23      return  0;
24  }
```

编译器 (4) 资源 编译日志 调试 搜索结果 关闭

行	列	单元	信息
		D:\Desktop\tmp\未命名2.cpp	In function 'int main()':
18	11	D:\Desktop\tmp\未命名2.cpp	[Error] reference to 'inf' is ambiguous

分析：inf 被重复定义

解决：

```
using two::x ;                  //方法3: using声明仅
x=-100 ;                        //直接访问，相当于tw
cout<<one::inf<<endl;             //using声明使用
cout<<M<<endl;
```

（3）例 2.5

①没有变化

②Dev C++未报错（）

分析：一、变量可以随用随定义，二、IDE 太智能了。

（4）例 2.6

```
sum=60
全局sum=5060
sum=400
sum=400
全局sum=5460

--------------------------------
Process exited after 0.1307 seconds with return value 0
请按任意键继续. . .
```

分析：int sum = 200 提前，第三个输出的 sum 没有使用域选择符，故输出局部变量 200 * 2 的值，因为局部变量 200 自乘了 2，故变为 400，第四句输出 400，因为自乘操作用在了局部变量上，则全局变量不变仍为 5460.

（5）例 2.7

①报错

```
7   8   D:\Desktop\tmp\未命名2.cpp   [Error] too few arguments to function 'void Fun(int, int, int)'
```

分析：对于第一句 `Fun(20) ;` 传入参数过少（第二个参数没有设置默认值）

②

```
20  5  10
20  30  10
20  30  40

--------------------------------
Process exited after 0.117 seconds with return value 0
请按任意键继续. . .
```

分析：与原函数输出无变化，i 的默认参数并没有用到

③报错

```
4   6   D:\Desktop\tmp\未命名2.cpp   [Error] default argument missing for parameter 3 of 'void Fun(int, int, int)'
```

分析：默认形参要放在最后

④同上

（6）例 2.8

```
D:\Desktop\tmp\未命名2.exe
a=35

--------------------------------
Process exited after 0.1311 seconds with return value 0
请按任意键继续. . .
```

分析：如此更改 `#define Multiply(x,y) (x)*(y)` 替换时加上括号就没有问题了

（7）例 2.9

报错

```
11   30   D:\Desktop\tmp\未命名2.cpp   [Error] call of overloaded 'square()' is ambiguous
```

分析：多个带默认参数的函数产生了二义性

（8）例 2.10

报错：

```
5   6   D:\Desktop\tmp\未命名2.cpp   [Error] 'r' declared as reference but not initialized
```

分析：引用被声明时必须被初始化

（9）例 2.11

```
选择D:\Desktop\tmp\未命名2.exe
a=3   b=5
a=3   b=3
c=10   d=20
c=10   d=10

--------------------------------
Process exited after 0.1307 seconds with return value 0
请按任意键继续. . .
```

分析：因为没有对 x 进行传引用操作，函数不会对第一个参数即主函数中的 a、c 变量产生影响，但是会改变第二个参数的值，就导致第二个变量的值被改为第一个变量的值，但第一个变量的值并没有发生改变。

五、实验总结

scanf()不能读入空格，要用 getchar()吃掉再进行下面的读入。

Int 型函数可以返回一些不可能存在的数值以起到 bool 型的功能，同理，int 型变量也可以起到存数据和 bool 型两个功能。

① Cout 精度控制可以使用 setprecision 实现

② 使用多个域时要小心重复声明，使用域选择符区分

③ IDE 可以帮你忽略掉一些错误

④ 注意全局变量和局部变量的区分

⑤ 注意函数重载时必须要规避掉二义性，默认参数要放到最后

⑥ 宏定义是简单替换具有风险，可以加一些括号规避掉部分风险

⑦ 引用被声明时必须被初始化

■ 附录：程序源码（建议基于 Highlight 软件导入）

OOP实验三.cpp

```cpp
#include<bits/stdc++.h>
#define ios ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);
#define debug(a) cout << #a << " " << a << endl
using namespace std;
typedef long long ll;
```

```
const double pi=acos(-1);
const double eps = 1e-8;
const int inf = 0x3f3f3f3f;
const int maxn = 100007;//1e5+7
const ll mod = 1000000007;//1e9+7

double a[maxn];

double Area(double a,double b,double c)
{
    if (a >= b + c) {
        return 0;
    }
    double p = (a + b + c) * 1.0 / 2;
    double ar = sqrt(p * (p - a) * (p - b) * (p - c));
    return ar;
}


bool cmp(double a,double b)
{
    return a > b;
}

int main()
{
    int m;
    scanf("m  = %d",&m);
    getchar();
    scanf("L  = ");
    for(int i = 1;i<=m;i++){
        cin>>a[i];
    }
/*  for(int i = 1;i<=m;i++){
        cout<<a[i]<<" \n"[i == m];
    }*/
    sort(a+1,a+m+1,cmp);
    int posc = -1,poss = -1;
    double mxs = 0,mxc = 0;
    for(int i = 1;i+2<=m;i++){
        if(a[i] < (a[i+1] + a[i+2])){
            posc = i;
            mxc = a[i] +a[i+1] + a[i+2];
            break;
```

```
        }
    }
    if(posc != -1){
        for(int i = 2;i+1<=m;i++){
            double areaa = Area(a[i-1],a[i],a[i+1]);
            if(areaa > 0){
                if(areaa > mxs){
                    mxs = areaa;
                    poss = i;
                }
            }
        }
        cout<<"最大周长：" << mxc<<" 选择
    （ "<<a[posc]<<" "<<a[posc+1]<<" "<<a[posc+2]<<" ）"<<endl;
        cout<<"最大面积：" << mxs<<" 选择（ "<<a[poss-
    1]<<" "<<a[poss]<<" "<<a[poss+1]<<" ）"<<endl;
    }
    else{
        cout<<0<<endl;
    }
    return 0;
}
```

Github 地址：
https://github.com/JiBinquan/OOP_Homework_CS2020/tree/master/3.OOP 实验三