

编号： 第 1 章第 1 次



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：	朱会琛
学号：	202311000202
班级：	计工本 2302
教师：	张庆科
时间：	2024 年 10 月 26 日



《面向对象程序设计》实验报告

报告要求：实验报告包含实验目的、实验内容、实验过程（详细操作流程）、实验结果（程序运行结果高清截图）、实验分析总结五个部分。报告中若涉及代码程序，请在附录部分提供完整程序源码及源码托管地址(基于 Highlight 软件导入源码)。报告撰写完毕后请将 PDF 格式版本上传到坚果云作业提交系统。

一、实验目的

1. 实验目的

- 理解指针和引用的区别与联系
- 掌握 C++ 指针和引用的基本用法
- 理解引用作为形式参数和返回类型的用法
- 掌握 C++ 动态内存申请和释放的基本用法
- 熟练运用 Visual Studio 进行算法代码的调试

二、实验内容

实验 1：请采用 C++ 语言编写重载函数 `void swap(...)`，实现对任意两个相同数值型数据的交换。

实验 2：阅读分析代码

建立 VS 项目，调试下列代码，找出程序 Bug 或错误，并予以更正，分析程序正确执行结果。

```
#include <iostream.h>
```

```
#include <cstdlib>
```

```
#include <windows.h>
```

```
int& function(int v, int *p, int &r)
```

```
{
```

```
    v = (*p)++;
```

```
    *p = 100;
```



```
p = new int;

*p = 1000;

r = 2 + (*p)++;

cout << "*p = " << *p << endl;

cin >> r;

return v;
}

/***** 主函数 *****/

int main()
{

    cout << "-----" << endl;

    //1. 传值，传址和传引用分析

    int a,b,c;

    cout<<"请输入整型变量 a, b, c 的数值: ";
```



```
cin>>a>>b>>c>>endl;
```

```
cout << "-----" << endl;
```

```
cout << "a = " << a << ", b = " << b << ", c = " << c << endl;
```

```
function(a, b, c);
```

```
cout << "a = " << a << ", b = " << b << ", c = " << c << endl;
```

//2. 引用规律分析

```
int &r = a;
```

```
r = a + b + c;
```

```
a = r + a + b + c;
```

```
int &r = b;
```

```
cout << "r = " << endl;
```

```
cout << "a = " << a << endl;
```

```
cout << "&r = " << &r << endl;
```

```
cout << "&a = " << &a << endl;
```

```
cout << "-----" << endl;
```



//3. 局部变量存储分析

```
int i = 10, j = 20, k = 30;
```

```
cout << "变量 a 的地址: " << &a << endl;
```

```
cout << "变量 b 的地址: " << &b << endl;
```

```
cout << "变量 c 的地址: " << &c << endl;
```

//4. 动态内存申请分析

```
int *p1 = new int (10) ;
```

```
int *p2 = new int[10]{10, 20, 30};
```

```
cout << "变量 p1 地址: " << p1 << endl;
```

```
cout << "变量 p2 地址: " << p2 << endl;
```

```
cout << "变量 p1 指向的地址: " << &p1 << endl;
```

```
cout << "变量 p2 指向的地址: " << &p2 << endl;
```

```
system("pause");
```

```
}
```

实验 3: 动态内存分配

自定义含有 n 个整数 ($n > 10000$) 的数组, 数组元素数值要求随机生成 (正负数元素满足均匀分布), 试编写函数 `FindMaxSumSequence(...)`, 找出数组中具有最大数值总和的连续子序列, 并输出该子序列及其总和。



提示：生成区间[a,b]之间（如[-1000,1000]）的任意随机整数方法：rand()%(b-a+1)+a（带上头文件 cstdlib）

实验 4：综合编程

给定 m 根木棍，每根木棍的长度记为 $L_i (3 \leq i \leq m)$ ，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，或面积尽可能最大的三角形（不一定同时最大），分别输出最大的周长和最大的面积。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。
2. 给出算法的复杂度分析，且要求算法复杂度必须低于 $O(n^3)$ 。
3. 算法设计时尽可能多的使用函数重载，引用，及动态内存申请等 C++ 核心特性。

输入样例：

请输入木棍的数目 $m = 5$

请输入 5 根木棍长度：2 3 4 5 10

程序输出：

构成的最大三角形周长为：12 （选择 3,4,5）

构成的最大三角形面积为：6 （选择 3,4,5）

三、实验过程

实验一：

源码：#include <iostream>

```
// 使用 std 命名空间，这样就可以直接使用其中的 cout 等，而无需写 std::cout
using namespace std;
```

```
// 交换两个整数
```

```
void swap(int& a, int& b) {
    int temp = a;
    a = b;
    b = temp;
}
```

```
// 交换两个浮点数
```

```
void swap(float& a, float& b) {
```



```
float temp = a;
a = b;
b = temp;
}

// 交换两个双精度浮点数
void swap(double& a, double& b) {
    double temp = a;
    a = b;
    b = temp;
}

int main() {
    int num1 = 5, num2 = 10;
    cout << "交换前（整数类型）: num1 = " << num1 << ", num2 = " << num2
<< endl;
    swap(num1, num2);
    cout << "交换后（整数类型）: num1 = " << num1 << ", num2 = " << num2
<< endl;

    float float1 = 3.14f, float2 = 6.28f;
    cout << "交换前（单精度浮点数类型）: float1 = " << float1 << ", float2
= " << float2 << endl;
    swap(float1, float2);
    cout << "交换后（单精度浮点数类型）: float1 = " << float1 << ", float2
= " << float2 << endl;

    double double1 = 2.71828, double2 = 3.14159;
    cout << "交换前（双精度浮点数类型）: double1 = " << double1 << ",
double2 = " << double2 << endl;
    swap(double1, double2);
    cout << "交换后（双精度浮点数类型）: double1 = " << double1 << ",
double2 = " << double2 << endl;
    return 0;
}
```

实验二:

```
源码: #include <iostream>
#include <cstdlib>
//#include <windows.h>

// 缺少 using namespace std; 命名空间
using namespace std;
```



```
int& function(int v, int *p, int &r)
{
    v = (*p)++;
    *p = 100;
    p = new int;
    *p = 1000;
    r = 2 + (*p)++;

    cout << "*p = " << *p << endl;
    cin >> r;

// 返回值与函数类型不符合, 直接返回 r
    // return v;
    delete p;
    return r;
}

/***** 主函数 *****/

int main()
{
    cout << "-----" << endl;

    //1. 传值, 传址和传引用分析
    int a,b,c;
    //类型不对, 新增一个指针*p 来保存 b 的地址
    int *p = &b;
    cout << "请输入整型变量 a, b, c 的数值: ";

// 输入不能使用 endl 输出流换行符
    // cin >> a >> b >> c >> endl;
    cin >> a >> b >> c;

    cout << "-----" << endl;
    cout << "a = " << a << ", b = " << b << ", c = " << c <<
endl;
    //将传入的形参改为指针类型 b -> p

    // 没有接受、使用 function 返回值
    // function(a, p, c);
    int &ans = function(a, p, c);
```




```
cout << "a = " << a << ", b = " << b << ", c = " << c << endl;
```

//2. 引用规律分析

```
int &r = a;  
r = a + b + c;  
a = r + a + b + c;
```

// 引用不允许重定义

```
//int &r = b;  
int &l = b;
```

```
cout << "r = " << r << endl;  
cout << "a = " << a << endl;
```

```
cout << "&r = " << &r << endl;  
cout << "&a = " << &a << endl;
```

```
cout << "-----" << endl;
```

//3. 局部变量存储分析

```
int i = 10, j = 20, k = 30;
```

```
cout << "变量 a 的地址: " << &a << endl;  
cout << "变量 b 的地址: " << &b << endl;  
cout << "变量 c 的地址: " << &c << endl;
```

//4. 动态内存申请分析

// 申请动态内存错误

```
// int *p1 = new int (10);  
int *p1 = new int[10];  
int *p2 = new int[10]{10, 20, 30};
```

// 指针意义理解错误, 上下交换

// 变量本身的地址

```
cout << "变量 p1 地址: " << &p1 << endl;  
cout << "变量 p2 地址: " << &p2 << endl;
```

// 变量所指向的数组的首地址

```
cout << "变量 p1 指向的地址: " << p1 << endl;
```



```
cout << "变量 p2 指向的地址: " << p2 << endl;

delete[] p1;
delete[] p2;

//cin.get();
system("pause");

}
```

实验三:

```
源码: #include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

// 函数用于找出具有最大数值总和的连续子序列
void FindMaxSumSequence(int* array, int n, int& startIndex, int&
endIndex, int& maxSum) {
    maxSum = array[0];
    int currentSum = array[0];
    startIndex = 0;
    endIndex = 0;
    int tempStart = 0;
    for (int i = 1; i < n; i++) {
        currentSum += array[i];
        if (currentSum > maxSum) {
            maxSum = currentSum;
            startIndex = tempStart;
            endIndex = i;
        }
        if (currentSum < 0) {
            currentSum = 0;
            tempStart = i + 1;
        }
    }
}

int main() {
    const int n = 10001; // 数组大小大于 10000
    int array[n];
    srand(static_cast<unsigned int>(time(nullptr)));
    for (int i = 0; i < n; i++) {
        // 生成[-1000, 1000]之间的随机数
    }
}
```



```
array[i] = rand() % 2001 - 1000;
}
int startIndex, endIndex, maxSum;
FindMaxSumSequence(array, n, startIndex, endIndex, maxSum);
cout << "具有最大和的连续子序列: ";
for (int i = startIndex; i <= endIndex; i++) {
    cout << array[i] << " ";
}
cout << "\n最大总和: " << maxSum << endl;
return 0;
}
```

实验四:

源码:

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

using namespace std;
// 判断是否能形成三角形
bool isTriangle(int a, int b, int c) {
    // 已经排序, 只需要考虑 两条最小边之和是否大于第三条边就行, 其实 a + b > c
    == c - b < a;
    return (a + b > c);
}

// 计算三角形面积
double triangleArea(int a, int b, int c) {
    double p = (a + b + c) / 2.0;
    return sqrt(p * (p - a) * (p - b) * (p - c));
}

int main() {

    int m;
    cout << "请输入木棍的数目: " << endl;
    cin >> m;
    vector<int> ans(m);
    cout << "请输入" << m << "根木棍长度" << endl;
    for (int i = 0; i < m; i++) {
        cin >> ans[i];
    }
}
```



```
int maxPerimeter = 0;
double maxArea = 0;

sort(ans.begin(), ans.end());
for (int i = m - 1; i >= 2; i --){
    int c = ans[i];
    int l = 0, r = i - 1;

    while (l < r){
        int a = ans[l];
        int b = ans[r];

        if (isTriangle(a,b,c)){
            int Primeter = a + b + c;
            maxPerimeter = max(Primeter, maxPerimeter);
            double Area = triangleArea(a, b, c);
            maxArea = max(maxArea, Area);
            l++;
        } else {
            r--;
        }
    }
}

cout << "输出三角形周长最大值: " << maxPerimeter << endl;
cout << "输出三角形面积的最大值: " << maxArea << endl;
return 0;
}
```

四、实验结果

实验一:



```
问题 输出 调试控制台 终端 端口
PS D:\Log\code04> & 'c:\Users\Zhu HuiChen\.vscode\extensions\ms-vscode.cpptools-1.22.10-win32-x64\debugadapters\bin\WinDBG\x64\WinDBG.exe' --stdout=Microsoft-MIEngine-Out-rjhzq2ad.0s3' '--stderr=Microsoft-MIEngine-Error-fixe3d-preter=mi'
交换前 (整数类型): num1 = 5, num2 = 10
交换后 (整数类型): num1 = 10, num2 = 5
交换前 (单精度浮点数类型): float1 = 3.14, float2 = 6.28
交换后 (单精度浮点数类型): float1 = 6.28, float2 = 3.14
交换前 (双精度浮点数类型): double1 = 2.71828, double2 = 3.14159
交换后 (双精度浮点数类型): double1 = 3.14159, double2 = 2.71828
PS D:\Log\code04>
```

实验二:

```
问题 输出 调试控制台 终端 端口
PS D:\Log\code04> & 'c:\Users\Zhu HuiChen\.vscode\extensions\ms-vscode.cpptools-1.22.10-win32-x64\debugadapters\bin\WinDBG\x64\WinDBG.exe' --stdout=Microsoft-MIEngine-Out-5iau5qjtj.zsi' '--stderr=Microsoft-MIEngine-Error-ylo3qcgj.jmy' '--pid=Microsoft-MIEngine-preter=mi'
-----
请输入整型变量a, b, c的数值: 12 34 45
-----
a = 12, b = 34, c = 45
*p = 1001
5
a = 12, b = 100, c = 5
r = 339
a = 339
&r = 0x5ffe70
&a = 0x5ffe70
-----
变量a的地址: 0x5ffe70
变量b的地址: 0x5ffe6c
变量c的地址: 0x5ffe68
变量p1地址: 0x5ffe60
变量p2地址: 0x5ffe58
变量p1指向的地址: 0x99fd0
变量p2指向的地址: 0x95350
Press any key to continue . . .
PS D:\Log\code04>
```

实验三:

```
具有最大和的连续子序列: 988 -852 839 724 -28 654 -766 889 252 -952 843 883 -111 -592 856 287 256 -61 876 -556 211 -686 354 -783 -817 480 -283 -12 956 267 -851 331 -177 -873 958 489 529 -12
2 464 246 -176 -446 649 481 -918 -388 87 346 -458 334 534 939 -352 -425 785 -432 937 -935 871 -722 591 832 -246 912 -889 -949 858 466 780 -120 -39 982 99 447 43 825 -919 -782 -416 286 757
-18 439 880 91 -335 -968 941 -281 887 721 394 650 162 334 -580 -362 -609 769 -581 920 520 -407 450 -320 -398 -682 880 146 194 306 -51 -569 -913 935 214 988 397 519 -392 838 -568 -987 -245
-483 595 819 994 690 -976 97 229 -586 -398 933 -650 -283 285 -152 -839 183 478 -86 294 -989 648 458 311 593 772 674 965 -453 -23 -339 312 -253 646 -405 912 53 -520 442 290 504 -659 767 68
2 -784 575 -656 -60 -255 674 -972 -104 -506 -27 998 75 382 973 391 -977 250 996 177 981 253 510 520 -864 9 -406 -517 -3 782 -382 -157 -861 528 848 356 522 818 -936 981 -52 987 176 678 -104
-612 -138 -480 -91 1000 -228 -19 -811 -529 -688 150 -578 495 757 490 -269 -954 519 122 -716 406 -960 264 -305 548 370 62 11 499 924 -824 -346 413 730 -122 623 997 -606 -251 -415 361 358 1
1 385 588 -339 -436 -631 596 -201 519 46 749 142 537 242 987 -18 -725 763 684 -62 -984 468 527 296 489 773 88 252 -756 -762 909 133 585 764 -482 -444 480 -472 -814 644 -582 222 -648 -53 -2
67 -319 -472 -276 67 -3 -985 745 385 454 -176 463 -411 843 -481 45 -934 795 -629 -952 319 160 -835 -251 19 321 675 668 159 224 -334 54 998 -972 61 -896 -829 -875 687 927 -764 -282 176 -191
-876 179 -560 270 306 -365 532 -548 634 543 -697 94 -216 -986 690 -832 43 -165 283 -633 537 40 -497 -448 415 928 388 -133 -411 357 -277 848 -998 493 360 -764 -497 646 -57 365 -4157 4 -64
1 257 -106 -380 638 118 -416 -297 444 -657 687 -36 -457 -818 663 -200 288 -241 -771 -539 -368 -737 -409 -379 316 -947 248 -62 -316 -760 -169 -757 -846 -45 132 969 948 -78 588 345 294 745 5
87 794 -851 -460 976 505 -718 599 -899 868 843 -571 -27 760 -697 -863 -934 -624 105 -289 366 588 663 962 -384 -520 -433 666 247 -112 -879 392 223 -741 -443 346 -629 423 -670 934 767 -354 5
68 394 684 476 95 -545 241 67 580 -989 -305 972 502 105 555 -192 974 -138 199 587 365 906 -83 -412 280 -998 -169 -696 -207 618 74 -885 -205 -364 -771 952 830 35 633 605 -990 -377 -467 204
691 -487 -263 -738 -839 703 218 56 112 -539 -676 -4 -648 592 -708 510 -235 -438 -259 -666 -969 -688 70 989 38 705 -557 -422 67 -272 -201 959 -103 -624 -960 651 -594 65 876 31 -904 -709 -31
3 -386 332 -871 911 775 988 233 -687 686 -689 -489 880 254 -591 -653 707 -183 945 259 375 274 186 435 430 270 582 999 949 361 -547 935 -917 384 552 -700 46 448 -901 896 -829 474 -810 -248
-208 -127 275 -348 545 896 -888 -585 355 561 -480 736 702 409 -485 -513 677 266 715 730 -70 759 368 482 775 244 -111 -125 -717 -28 502 -767 150 -523 868 115 -586 -121 310 -602 42 -988 765
129 996 832 -713 442 -740 -668 311 323 -24 -405 487 723 -371 249 -958 48 -299 879 -908 570 766 -114 -685 -472 812 -926 -444 -389 954 -900 -72 -232 -973 121 437 -704 612 -488 -930 664 558 8
83 -928 -998 -395 663 396 349 -432 -785 98 845 -850 -944 -853 -1000 763 -219 -610 -47 -437 576 -832 -384 -392 519 -151 552 -898 -126 -28 8 -153 820 542 -17 910 -347 -320 763 -898 995 -336
772 -967 -920 540 -816 -416 -780 622 422 -192 -835 -705 687 381 444 -562 179 -433 -642 -860 -835 -396 722 -990 -747 461 -145 779 836 -138 -280 -536 -227 -145 -950 -182 -954 877 -362 588 9
53 -663 523 -236 51 322 509 -612 221 801 516 850 -500 -330 912 -546 -667 -614 873 -529 -838 18 138 194 446 -985 454 -795 602 -722 348 900 650 -993 -492 -571 -171 747 670 409 235 555 -231 7
75 432 228 536 868 -327 -475 773 -312 893 -226 -511 -172 -435 -376 647 325 420 261 -312 -565 555 -358 100 544 455 558 -843 732 -537 800 657 -267 -818 793 -264 -478 -598 -824 846 583 -15 -3
6 2 358 838 2 600 -554 -873 -365 711 -643 -864 -238 144 -743 -590 742 706 -886 827 462 -163 199 -216 -785 -187 458 -923 605 -8 645 951 487 482 -949 896 -816 -97 238 658 545 -452 -160 -706
-869 -898 -128 -19 369 -221 -166 -575 928 -918 -963 -683 -565 -120 759 -311 -220 -264 -13 855 -913 656 -549 -694 412 -236 -983 744 473 -936 -662 604 681 -446 -664 261 -480 -910 944 -714 -
-554 -970 16 921 -273 63 466 -846 -115 464 -951 -683 88 717 618 561 391 -745 -590 -71 232 -543 180 576 581 692 494 482 -71 92 760 -629 -370 442 800 287 948 080 239 822 -697 -820 -745 -728 9
85 250 357 -413 -464 560 983 -946 -272 -546 563 -683 -545 -652 -908 181 437 484 913 317 697 940 451 -386 -958 692 -47 -783 422 -740 422 920 -214 -437 388 -14 -525 264 -816 522 159 775 812
-314 -109 685 775 617 -211 -48 243 943 481 16 682 -399 -452 -421 -236 300 524 -57 771 577 315 972 486 543 -957 15 559 535 470 -357 372 -512 183 -743 -168 -238 739 177 411 498 -325 -472 -6
47 -43 -173 -958 -892 -295 399 837 164 818 -340 445 -988 719 291 614 -16 -148 199 5 66 -720 -659 -270 707 -98 874 -921 -917 -341 181 -44 225 -357 -164 428 -560 752 -217 -838 617 -209 82 -8
52 643 -843 -273 -29 -832 39 -613 -441 -265 470 579 -614 -265 -540 118 77 -795 34 -134 641 -299 -626 865 785 243 -567 -376 266 -956 691 -331 493 -158 682 319 -886 718 -288 613 -253 259 -11
0 80 -623 -96 -92 484 -177 -758 -340 383 -456 -909 -76 -466 -820 -233 927 -415 -398 6 275 -399 -617 -333 293 -249 -563 -861 795 -453 3 748 312 -518 -280 -148 281 816 -29 685 755 -970 995 9
7 -887 -938 658 -457 -661 283 499 464 847 -107 -37 872 -577 -586 201 -416 15 -56 499 -609 427 387 297 -140 -90 -455 -592 797 465 797 933 -698 826 -760 -210 -880 955 301 980 -976 -979 -107
-690 -272 -516 682 -865 -148 406 393 151 872 -840 -894 -387 -139 938 -809 22 -887 -383 -680 -375 14 995 -545 91 74 237 -570 863 -58 -285 789 737 -213 369 -154 685 647 -198 318 872 150 -80
163 285 945 -695 -133 185 948 466 482 650 76 610 -827 -404 -2 834 -74 -183 -621 659 351 -821 872 195 624 -982 566 -651 -515 -941 -239 338 252 -452 -593 517 767 -734 633 946 -726 155 54 71
5 954 580 518 568 2 886 885 187 -62 -433 -426 314 600 1000 106 -760 991 697 553 -721 -134 508 394 871 -985 -624 777 177 813 -399 645 908 684 -696 777 693 833 -596 -91 984 493 503 -732 807
-818 743 662 479 269 401 718 838 2 -868 -832 -270 806 -134 257 450 699 773 443 845 -157 278 -559 892 384 488 887 -811 -203 -415 -657 -67 777 -44 -472 620 88 570 -742 -464 -644 11 577 756 -
190 -169 -17 744 -142 315 193 -18 -8 750 -797 531 -486 -57 -926 481 466 -93 966 -368 502
最大总和: 38350
PS D:\Log\code04>
```

实验四:



```
6
7 10 9 7
输出三角形周长最大值: 26
输出三角形面积的最大值: 30.5941
PS D:\Log\code04> ^C
PS D:\Log\code04>
PS D:\Log\code04> & 'c:\Users\Zhu HuiChen\.vscode\extensions\ms-vscode.c
'--stdout=Microsoft-MIEngine-Out-mbyc5512.m41' '--stderr=Microsoft-MIEng
preter=mi'
请输入木棍的数目:
5
请输入5根木棍长度
2 5 6 8 11
输出三角形周长最大值: 13
输出三角形面积的最大值: 4.68375
PS D:\Log\code04> 
```

五、实验总结

答: 实验 1 : 通过重载 swap 函数, 掌握了函数重载的概念和用法, 能够针对不同数值类型实现数据交换, 增强了程序的灵活性和通用性。

实验 2 : 对一段给定代码进行调试分析, 涉及到传值、传址、传引用的理解, 以及局部变量存储和动态内存申请等知识点。在这个过程中, 学会了如何查找和纠正代码中的错误, 深入理解了 C++ 中各种数据传递方式和内存管理的细节。

实验 3 : 要求实现寻找具有最大数值总和的连续子序列, 锻炼了对数组操作和算法设计的能力, 涉及到随机数生成和动态内存分配, 进一步提升了对 C++ 语言特性的运用能力。

实验 4 : 综合了多种 C++ 核心特性, 通过解决选择木棍组成三角形的问题, 培养了算法设计和复杂度分析的能力, 同时强调了函数重载、引用和动态内存申请的实际应用, 提高了综合编程能力和对 C++ 语言的整体掌握程度。