**《C++程序设计》**

**课程实验报告（第 2 次）**

**学 院：数学与计算机学院**

**姓 名： 张宇**

**学 号： 201810804017**

**专 业：软件工程**

**班 级：1班**

**实验地点：分析测试中心7A-1**

**实验时间： 2019.4.3**

**指导教师：张会**

|  |
| --- |
| 实验名称：实验2 C++对C 的扩充 |
| 所使用的开发工具及环境：PC机一套 Visual Studio 2010 |
| 实验要求：  1.硬件基本配置：Intel PentiumIII以上级别的CPU，大于64MB的内存。  2.软件要求：Window 7操作系统，Visual Studio 2010或更高版本开发环境。  3.实验学时：2学时  4.实现实验内容中的题目。  5.写实验报告 |
| 实验目的：  1．进一步了解和熟悉Visual Studio 2010开发环境，学会在Visual Studio 2010环境下调试程序；  2．熟悉C++中简单的标准输入输出函数的实用；  3．理解const修饰符的作用，并学会应用const修饰符；  4．理解内置（内联）函数的优缺点并学会使用内置函数；  5．理解和使用函数重载以及带默认参数的函数；  6．使用new和delete进行动态内存管理；  7．理解和使用引用。   |  | | --- | | 实验内容：  **（一） 程序阅读**  **1．理解下面的程序，并在**Visual Studio 2010**下运行查看结果，回答程序后面的问题。**  #include <iostream >  using namespace std;  int max\_def(int x, int y)  {  return (x>y?x:y);  }  int max\_def(int x, int y, int z)  {  int temp = 0;  return (temp=(x>y?x:y))>z?temp:z;  }  double max\_def(double x, double y)  {  return (x>y?x:y);  }  int main()  {  int x1 = 0;  int x2 = 0;  double d1 = 0.0;  double d2 = 0.0;  x1 = max\_def(5,6);  x2 = max\_def(2,3,4);  d1 = max\_def(2.1,5.6);  d2 = max\_def(12.3,3.4,7.8);//-----------------------------------------------------①  cout<<"x1="<<x1<<endl;  cout<<"x2="<<x2<<endl;  cout<<"d1="<<d1<<endl;  cout<<"d2="<<d2<<endl;//--------------------------------------------------------②  return 1;  }    **问题一**：上述程序的输出结果是什么？  **答：如图**  **问题二**：哪些情况可以参与函数的重载？  **答：**参数个数不同，参数类型不同  **问题三**：①处调用的是哪个函数？  **答：**max\_def(int x, int y, int z)  **问题四：**②处的输出结果为什么是d2=12，而不是d2=12.3？  **答：**max\_def(int x, int y, int z) 从“double”转换到“int”，数据丢失  **2．理解下面的程序，并在Visual Studio2010下运行查看结果，回答程序后面的问题。**  #include <iostream>  using namespace std;  int main()  {  int \*p1 = new int; // -----------------------------------------------------①  int \*p2 = new int(0); //-----------------------------------------------------②  char \*p3 = new char[64]; //-----------------------------------------------------③  return 1;  }  问题一：①、②、③处动态申请内存分别代表什么不一样的意思？  **答：：**1是一个int型的变量 2是一个int型的数组 3是一个char型的数组  问题二：该程序存在什么隐患？改正该程序使隐患消除。  **答：：**动态申请的内存没有删除掉  3．理解下面的程序，并在**Visual Studio2010**下运行查看结果，回答程序后面的问题。  #include <iostream>  using namespace std;  void swap(int a, int b)  {  int temp = a;  a = b;  b = temp;  }  void swap(int \*a, int \*b)  {  int temp = \*a;  \*a = \*b;  \*b = temp;  }  int main()  {  int i = 5;  int j = 10;  cout<<"Before swap: i="<<i<<",j="<<j<<endl;  swap(i,j); //-----------------------------------------------------①  cout<<"After the first swap: i="<<i<<",j="<<j<<endl;  swap(&i,&j); // -----------------------------------------------------②  cout<<"After the second swap: i="<<i<<",j="<<j<<endl;    return 1;  }  问题一：上述程序的输出结果是什么？  **答：：**Before swap: i=5,j=10  After the first swap: i=5,j=10  After the second swap: i=10,j=5  问题二：①处函数调用后并不能实现两个数的交换，而②处却可以，为什么？  **答：：**2交换的是地址  问题三：②处调用的是哪个重载函数？  **答：** void swap(int \*a, int \*b)  （二）程序填空  1.按照带默认参数函数的机制，将下面的程序填写完整，并运行调试结果。  #include <iostream>  using namespace std;  int max(int x, int y=2, int z=5) //求x,y,z三个数的最大值  { int m=x;  if( y>m) m=y;  if( z>m )m=z ;  return (m);  }  void main( )  {  cout<<max(1,2,3)<<endl;  cout<<max(1,8)<<endl;  cout<<max(9)<<endl;  }    2.根据引用的原理，运行调试如下程序并分析结果：  #include <iostream>  using namespace std;  int & max( int & x, int &y)  {  if(x>y)x=y ;  else return(y);  }  void main( )  {  int a,b;  a=3; b=-5;  int &c=a;  max(c,b)++;  cout<<”a=”<<a<<endl;  cout<<”c=”<<c<<endl;  cout<<”b=”<<b<<endl;  }    （三） 程序设计  1．使用函数重载的方法定义两个重名函数，分别求出整形数平面间两点间距离和双精度平面间两点间距离，如果没有输入第二点的坐标则默认为圆点（0,0）。  #include<iostream>  #include<cmath>  using namespace std;  double distance(int x1,int x2,int y1=0,int y2=0)  {  double s;  s=sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1));  return s;  }  double distance(double x1,double x2,double y1=0,double y2=0)  {  double s;  s=sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1));  return s;  }  void main()  {  double s1,s2;  s1=distance(2,3,4,5);  s2=distance(2,5,3,5);  cout<<"s1="<<s1<<endl;  cout<<"s2="<<s2<<endl;  }  2．设计一个函数：exchange(float x, float y, float z)，当调用exchange(a,b,c)时，将a的内容  赋值给b，b的内容赋值给c，c的内容赋值给a，要求采用引用的方式来实现。  #include<iostream>  using namespace std;  float exchange(float &x,float &y, float &z)  {  float t;  t=x;  x=y;  y=t;  y=z;  z=t;  return x,y,z;  }  int main()  {  float a,b,c;  cout<<" please inputa,b,c"<<endl;  cin>>a>>b>>c;  float &i=a;  float &j=b;  float &k=c;  exchange(i,j,k);  cout<<"交换后的值"<<a<<" "<<b<<" "<<c<<" "<<endl;  return 0;  }  3. 编写一个函数，实现两个字符串变量的交换，要求参数用引用。  #include <iostream>  #include <string>  using namespace std;  void swap(string &x,string &y);  int main()  {  string str1,str2;  cout<<"输入str1:";  cin>>str1;  cout<<"输入str2:";  cin>>str2;  swap(str1,str2);  cout<<"str1="<<str1<<endl;  cout<<"str2="<<str2<<endl;  return 0;  }  void swap(string &x ,string &y)  {  string temp;  temp=x;x=y;y=temp;  }  4.编写几个计算面积的函数，分别计算圆、矩形、梯形和三角形的面积。  要求：采用重载函数实现。   【提示】：函数原型如下：  double area(double radius=0);  // 圆面积，参数为半径，缺省参数0，表示点面积  double area(double a, double b);  // 计算矩形面积，参数为长和宽   double area(double a, double b, double h);  // 计算梯形面积，参数为两底和高   double area(double a, double b, double c, int);  // 三角形，参数为三边长，int型参数起表示作用，以区别于梯形，不参加计算。  #include<iostream>  #include<cmath>  using namespace std;  double area(double radius=0)  // 圆面积，参数为半径，缺省参数0，表示点面积  {  double s1;  s1 = 3.14\*radius\*radius;  return s1;  }  double area(double a, double b)  // 计算矩形面积，参数为长和宽  {  double s2;  s2 = a \* b;  return s2;  }  double area(double a, double b, double h)  // 计算梯形面积，参数为两底和高  {  double s3;  s3 = ((a + b)\*h)/2;  return s3;  }  double area(double a, double b, double c, int)  // 三角形，参数为三边长，int型参数起表示作用，以区别于梯形，不参加计算。  {  double p,s4;  p = (a+b+c)/2;  s4 = sqrt(p\*(p-a)\*(p-b)\*(p-c));  return s4;  }  int main()  {  double area(double radius=0);  double area(double a, double b);  double area(double a, double b, double h);  double area(double a, double b, double c, int);  double a,b,h,c,radius;  double s1,s2,s3,s4;  cin>>a>>b>>c>>h>>radius;  s1 = area(radius);  s2 = area(a,b);  s3 = area(a,b,h);  s4 = area(a,b,c);  cout<<s1<<endl;  cout<<s2<<endl;  cout<<s3<<endl;  cout<<s4<<endl;  return 0;  }  5.建立一个头文件area.h，在其中声明第4题中的所有area()函数，然后再在area.cpp文件中  实现4题中的4个面积函数。另外建立一个实现文件mainTest.cpp，在其中定义主函数。通  过包含area.h，输入数据并输出各种图形面积。  area.h  #include <iostream>  #include<cmath>  #define PI 314159  double area(double radius=0);  double area(double a,double b);  double area(double a,double b,double h);  double area(double a,double b,double c,int);  area.cpp  #include <iostream>  #include "area.h"  double area(double radius){  return PI\*radius\*radius;  }  double area(double a,double b){  return a\*b;  }  double area(double a,double b,double h){  return (0.5\*(a+b)\*h);  }  double area(double a,double b,double c,int){  double s=0.5\*(a+b+c);  return sqrt(s\*(s-a)\*(s-b)\*(s-c));  }  mainTest.cpp  #include <iostream>  #include "area.h"  using namespace std;  int main(){  cout<<"Area of point is:"<<area()<<endl;  cout<<"Area of square is:"<<area(1,1)<<endl;  cout<<"Area of circle is:"<<area(0.5)<<endl;  cout<<"Area of trapezium is:"<<area(1,0.5,1)<<endl;  cout<<"Area of triangle is:"<<area(1,sqrt(1+0.5\*0.5),sqrt(1+0.5\*0.5),0)<<endl;  return 0;  }    6.编写一个程序，用同一个函数名对n个数据进行求平均值，数据类型可以是整形、单精  度型、双精度型。  （1）用重载函数实现。  （2）用函数模板实现。  比较这两种方法各有什么特点，什么情况下可以用函数模板代替重载函数？什么情况下不  可以用函数模板代替重载函数？  （1）用重载函数实现。  #include<iostream>  using namespace std;  int ave(int \*a, int n);  float ave(float \*a, int n);  double ave(double \*a, int n);  int main()  {  int n, num;  double \*a;  cout<<"请输入参数个数并依次输入各参数"<<endl;  cin>>n;  a = new double[n];  for (int i=0;i<n;i++)cin>>a[i];  cout<<ave(a,n)<<endl;  system("pause");  return 0;  }  int ave(int \*a, int n)  {  int total = 0;  for (int i = 0; i < n; i++)  total += a[i];  return(total / n);  }  float ave(float \*a, int n)  {  float total = 0;  for (int i = 0; i < n; i++)  total += a[i];  return(total / n);  }  double ave(double \*a, int n)  {  double total = 0;  for (int i = 0; i < n; i++)  total += a[i];  return(total / n);  }    （2）用函数模板实现。  #include<iostream>  using namespace std;  template<class M>  M ave(M \*a, int n)  {  M total = 0;  for (int i = 0; i < n; i++)  total += a[i];  return(total / n);  }  int main()  {  int n;  double \*a;  cout<<"请输入参数个数并依次输入各参数"<<endl;  cin >> n;  a = new double[n];  for (int i = 0; i < n; i++) cin >> a[i];  cout << ave(a, n) << endl;  system("pause");  return 0;  } | | **结果与分析 (** 收获、问题 **)**  **1．进一步了解和熟悉Visual Studio 2010开发环境，学会在Visual Studio 2010环境下调试程序；**  **2．熟悉C++中简单的标准输入输出函数的实用；**  **3．对函数的重载不够了解，头函数的定义不会；**  **4．理解内置（内联）函数的优缺点并学会使用内置函数；**  **5．理解和使用函数重载以及带默认参数的函数；** | |