山东大学 软件 学院

c++程序设计 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301108 | 姓名： 牛翔宇 | | 班级： 18.4 |
| 实验题目：实验1 | | | |
| 实验学时：4 | | 实验日期：2019.3.14 | |
| 实验目的：  熟悉C++的表达式、运算符、简单程序设计  熟悉C++的函数，包括函数调用、函数递归 | | | |
| 硬件环境：  PC机 | | | |
| 软件环境：  Dev C++ | | | |
| 实验步骤与内容：  1：请写出下列表达式的值，并编写一个程序验证。  (1) 3.5\*3+2\*7-'a' //字符’a’对应的ASCII码为97  (2) 26/3+34%3+2.5  (3) 45/2+(int)3.14159/2 //注意强制类型转换，（int）3.1415926 == 3  (4) a=b=(c=a+=6) 设a的初值为3  (5) a=3\*5,a=b=3\*2  (6) (int)(a+6.5)%2+(a=b=5) 设a的初值为3 //注意强制类型转换（int）（a+6.5）== 9  (7) x+a%3\*(int)(x+y)%2/4 设x=2.5,a=7,y=4.7 //注意强制类型转换（int）（x+y）== 7  (8)(float)(a+b)/2+(int)x%(int)y设a=2,b=3,x=3.5,y=2.5 //注意强制类型转换（float）（a+b）==5.0f (int)x == 3 (int)y == 2  源代码：  --begin--cpp--code--  #include<iostream>  using namespace std;  int main(){  //q1  cout << 3.5 \* 3 + 2 \* 7 - 'a' << "\n";  //q2  cout << 26 / 3 + 34 % 3 + 2.5 << "\n";  //q3  cout << 45/2+(int)3.14159/2 << "\n";  //q4  int a = 3, b, c;  a = b = (c = a += 6);  cout << a << "\n";  //q5  a = 3 \* 5;  a = b = 3 \* 2;  cout << a << "\n";  //q6  a = 3;  cout << (int)(a+6.5)%2+(a=b=5) << "\n";  //q7  int x = 2.5, y = 4.7;  a = 7;  cout << x+a%3\*(int)(x+y)%2/4 << "\n";  //q8  a = 2;  b = 3;  x = 3.5;  y = 2.5;  cout << (float)(a+b)/2+(int)x%(int)y << "\n";    return 0 ;  }  --end--cpp--code--  运行结果：    2：要将"China"译成密码,密码规律是: 用原来的字母后面第4个字母代替原来的字母。例如字母A后面第4个字母是E,用E代替A。因此,"China"应译为"Glmre"。请编写一程序,用赋初值的方法使c1,c2,c3，c4，c5这5个变量的值分别为'C'， 'h'， 'i'， 'n'， 'a',经过运算,使c1，c2，c3，c4，c5分别变为'G'， 'l'， 'm'， 'r'， 'e',并输出。  源代码：  --begin--cpp--code--  #include<iostream>  using namespace std;  int main(){  //赋初值，使c1,c2,c3,c4,c5这5个变量的值分别为'C' 'h' 'i' 'n' 'a'  char c1 = 'C', c2 = 'h', c3 = 'i', c4 = 'n', c5 = 'a';  //用原来的字母后面第4个字母代替原来的字母  c1 += 4;  c2 += 4;  c3 += 4;  c4 += 4;  c5 += 4;  //输出密码  cout << c1 << c2 << c3 << c4 << c5;  }  --end--cpp--code--  运行结果：    3：用迭代法求x=。要求前后两次求出的x的差的绝对值小于10-5时结束，并输出每次迭代的结果和最后结果。能连续求多个正数的平方根，输入负数时给出提示，并结束程序。  用迭代法求平方根需要多次迭代，因此需要循环语句，循环条件应为不误差的要求，当误差不符合要求时，进入循环迭代。而要连续求多个正数的平方根，也需要循环语句，循环条件为输入的数是正数。因此程序需要双重循环。  源代码：  --begin--cpp--code--  #include <iostream>  using namespace std;  int main() {  double x1 = 0, x2 = 0, a = 0, num = 1;  cin >> a;    if(a < 0){  cout << "输入错误！";  return 0;  }  while(a > 0){  x1 = a / 2;  x2 = (x1 + a / x1) / 2;  cout << "第1次迭代结果为" << x2 << "\n";  while(x2 - x1 > 0.00001 || x2 - x1 < -0.00001){  x1 = x2;  x2 = (x1 + a / x1) / 2;  num++;  cout << "第" << num << "次迭代结果为" << x2 << "\n";  }  cout << "最终迭代结果为" << x2 << "\n";  cout << "请输入下一个数（输入负数退出）：";  cin >> a;  }  return 0;  }  --end--cpp--code--  运行结果：以36为例。    4：用递归方法求n阶勒让德多项式的值。方法传递两个参数，一个是变量x，另一个是阶数n，当n小于0时，报错并退出程序。利用当n=1或n=0时跳出递归。  源代码：  --begin--cpp--code--  #include <iostream>  using namespace std;  double lp(double x, int n){  double p = 0;  if(n < 0){  cout << "ERROR!";  return 0;  }  if(n == 0)  p = 1;  if(n == 1)  p = x;  if(n > 1){  p = ((2 \* n - 1) \* x \* lp(x, n - 1) - (n - 1) \* lp(x, n - 2)) / n;  }  return p;  }  int main(){  cout << lp(9,2);  return 0;  }  --end--cpp--code--  运行结果：以9的2阶勒让德多项式为例。    5. 用函数重载实现函数sum，分别求int、float、double型的三个数的和，在main函数中分别调用这3个函数。要注意方法的返回值类型和参数类型。  源代码：  --begin--cpp--code--  #include <iostream>  using namespace std;  int sum(int a, int b, int c){  return a + b + c;  }  double sum(double a, double b, double c){  return a + b + c;  }  float sum(float a, float b, float c){  return a + b + c;  }  int main(int argc, char\*\* argv) {  cout << sum(1, 2, 3) << "\n";  cout << sum(1.0, 2.0, 3.0) << "\n";  cout << sum(1.5f, 2.5f, 3.5f);    return 0;  }  --end--cpp--code--  运行结果：    6. 用函数模板实现上题中的功能。  使用函数模板时，要先声明模板，T为类型参数，之后定义通用函数，用T做虚拟类型名。  源代码：  --begin--cpp--code--  #include<iostream>  using namespace std;  template<typename T> //声明模板，T为类型参数  T sum(T a, T b, T c){ //定义通用函数，用T做虚拟类型名  return a + b +c;  }  int main() {  cout << sum(1, 2, 3) << "\n"; //T被int取代  cout << sum(1.0, 2.0, 3.0) << "\n";//T被double取代  cout << sum(1.5f, 2.5f, 3.5f); //T被float取代    return 0;  }  --end--cpp--code--  运行结果： | | | |
| 结论分析与体会：  1、初次接触C++，感觉相当部分内容与Java相似或相同，因此在学过Java一部分基础内容后对于C++的掌握很有帮助  2、C++面向过程，Java面向对象，编写代码时很多语句有不同之处，应注意区分，不要混淆。比如main（）方法、#include等与Java不同。  3、相比函数重载来说，函数模板更为简洁，但也有缺点例如参数的个数固定，参数的类型必须一致等。 | | | |