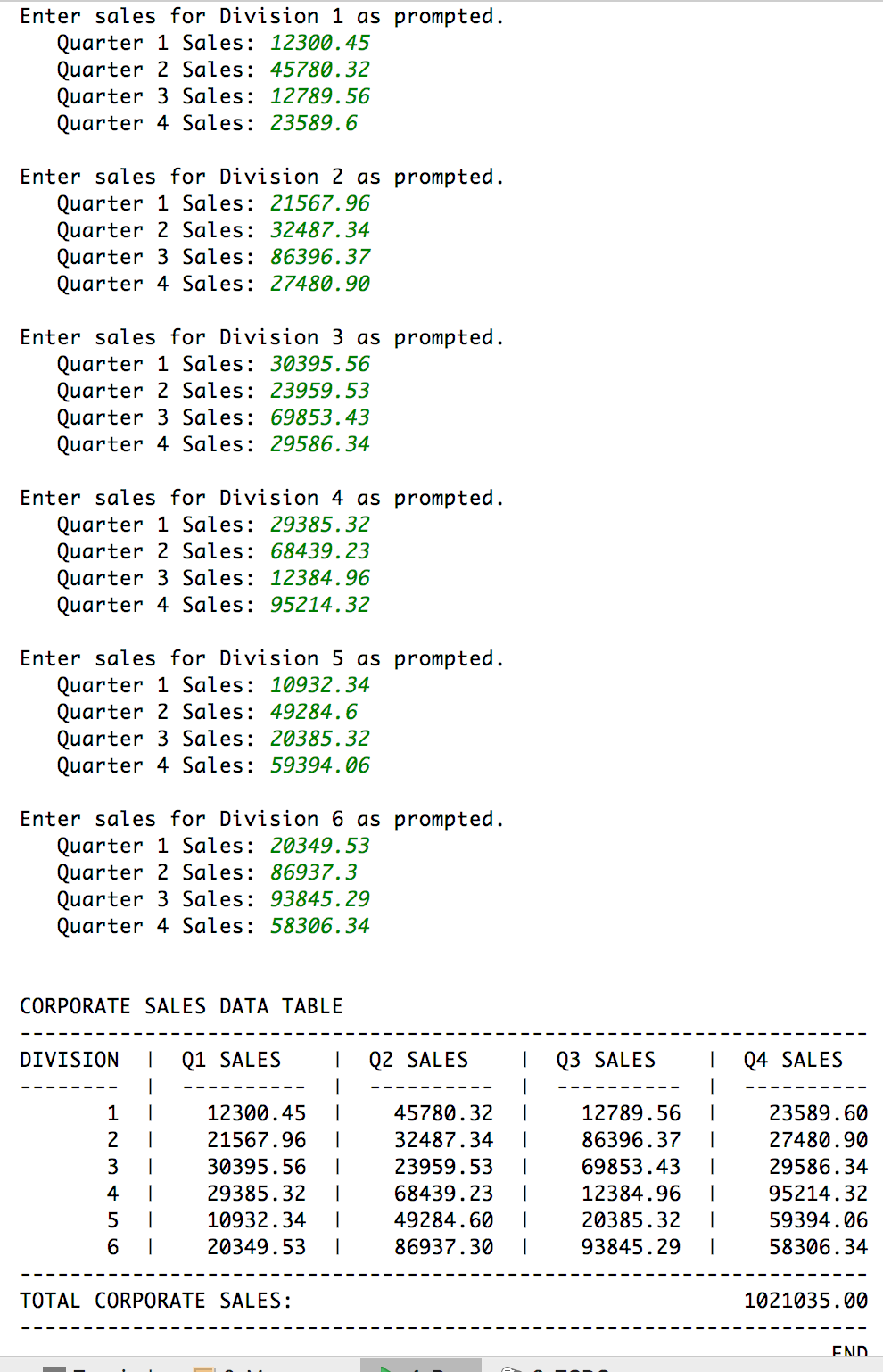
CSC 122 001 Computer Science II

Julius Ranoa

Chapter 11 Programming Challenge 7 Corporate Sales

See book for complete requirements.

Screenshot of runtime.



Files included: (1) DivSales.h, (2) DivSales.cpp, (3) main.cpp

**DivSales.h**

#ifndef **CH11\_PR7\_CORPORATE\_SALES\_DIVSALES\_H**#define **CH11\_PR7\_CORPORATE\_SALES\_DIVSALES\_H  
  
  
class** DivSales {  
  
**private**:  
 **static double** totalCorporateSales;  
 **double** quarterSales[4];  
  
**public**:  
 **static double** getTotalCorporateSales();  
 **void** setSales(**double**, **double**, **double**, **double**);  
 **double** getSales(**int**) **const**;  
  
};  
  
  
#endif *//CH11\_PR7\_CORPORATE\_SALES\_DIVSALES\_H*

**DivSales.cpp**

#include **<numeric>**#include **<algorithm>**#include **<iostream>**#include **"DivSales.h"  
  
double** DivSales::totalCorporateSales = 0;  
  
**double** DivSales::getTotalCorporateSales() {  
 **return** totalCorporateSales;  
}  
  
**void** DivSales::setSales(  
 **double** salesQ1, **double** salesQ2, **double** salesQ3, **double** salesQ4  
) {  
 quarterSales[0] = salesQ1;  
 quarterSales[1] = salesQ2;  
 quarterSales[2] = salesQ3;  
 quarterSales[3] = salesQ4;  
 totalCorporateSales += std::accumulate(quarterSales, quarterSales + 4, 0);  
}  
  
**double** DivSales::getSales(**int** idx) **const** {  
 **if** (idx < 0 || idx >= 4) {  
 std::cout << **"Error: Invalid index for quarter. \n"**;  
 std::cout << **"Index must be from 0 to 3 inclusive. \n"**;  
 exit(-404);  
 }  
 **return** quarterSales[idx];  
}

**main.cpp**

#include **<iostream>**#include **<iomanip>**#include **"DivSales.h"  
  
int** main() {  
 DivSales div[6];  
 **int** NUM\_DIV = **sizeof**(div) / **sizeof**(div[0]);  
  
 *// Get sales for all four divs* **for** (**int** i = 0; i < NUM\_DIV; i++) {  
 **static double** temp\_q[4];  
 std::cout << **"Enter sales for Division "** << i + 1 << **" as prompted. \n"**;  
 **for** (**int** q = 0; q < 4; q++) {  
 std::cout << **" Quarter "** << q + 1 << **" Sales: "**;  
 std::cin >> temp\_q[q];  
 }  
 div[i].setSales(temp\_q[0], temp\_q[1], temp\_q[2], temp\_q[3]);  
 std::cout << **"\n"**;  
 }  
 std::cout << **"\n"**;  
  
 *// Display information in table.* std::cout << std::fixed << std::showpoint;  
 std::cout << **"CORPORATE SALES DATA TABLE \n"**;  
 std::cout << **"-------------------------------------------------------------------- \n"**;  
 std::cout << **"DIVISION | Q1 SALES | Q2 SALES | Q3 SALES | Q4 SALES \n"**;  
 std::cout << **"-------- | ---------- | ---------- | ---------- | ---------- \n"**;  
 **for** (**int** i = 0; i < NUM\_DIV; i++) {  
 std::cout << std::setw(8) << std::setprecision(0) << std::right  
 << i + 1 << **" "**;  
 **for** (**int** q = 0; q < 4; q++) {  
 std::cout << **"| "** << std::setw(10) << std::setprecision(2)  
 << div[i].getSales(q) << **" "**;  
 }  
 std::cout << **"\n"**;  
 }  
 std::cout << **"-------------------------------------------------------------------- \n"**;  
 std::cout << **"TOTAL CORPORATE SALES: "**;  
 std::cout << std::setw(45) << std::setprecision(2) << std::right  
 << DivSales::getTotalCorporateSales() << **"\n"**;  
 std::cout << **"-------------------------------------------------------------------- \n"**;  
 std::cout << **" END \n"**;  
  
 **return** 0;  
}