C++ Programming | (4th Edition)

Chapter 10, Problem 17PE   Bookmark   Show all steps: ON

Problem

Write a program to keep track of a hardware store inventory. The store sells various items. For each item in the store, the following information is kept: item ID, item name, number of pieces ordered, number of pieces currently in the store, number of pieces sold, manufacturer's price for the item, and the store's selling price. At the end of each week, the store manager would like to see a report in the following form:

```
                    Friendly Hardware Store

itemID itemName        pOrdered pInStore pSold manufPrice sellingPrice
4444   Circular Saw       150      150     40      45.00       125.00
3333   Cooking Range       50       50     20     450.00       850.00
.
.
.

Total Inventory: $#########.##
Total number of items in the store: __________
```

The total inventory is the total selling value of all of the items currently in the store. The total number of items is the sum of the number of pieces of all of the items in the store.

Your program must be menu driven, giving the user various choices, such as checking whether an item is in the store, selling an item, and printing the report. After inputting the data, sort it according to the items' names. Also, after an item is sold, update the appropriate counts.

Initially, the number of pieces (of an item) in the store is the same as the number of pieces ordered, and the number of pieces of an item sold is zero. Input to the program is a file consisting of data in the following form:

itemID

itemName

pOrdered manufPrice sellingPrice

Use seven parallel vectors to store the information. The program must contain at least the following functions: one to input data into the vectors, one to display the menu, one to sell an item, and one to print the report for the manager.

Step-by-step solution

Step 1 of 3

```
#include<iostream>
#include<fstream>
#include<string>
#include<iomanip>
#include<vector>
using namespace std;

//   function prototypes
void sortInput( int count, vector<int>& itemID,
     vector<string>& itemName, vector<int>& ordered,
     vector<int>& inStore, vector<int>& sold,
     vector<double>& mPrice, vector<double>& sPrice );
int getInput( istream& inp, vector<int>& itemID,
     vector<string>& itemName, vector<int>& ordered,
     vector<int>& inStore, vector<int>& sold,
     vector<double>& mPrice, vector<double>& sPrice );
char displayMenu();
void sellAnItem(vector<int>& itemID, vector<int>& inStore,
                         vector<int>& sold, int count );
void printReport( vector<int> itemID,
     vector<string> itemName, vector<int> ordered,
     vector<int> inStore, vector<int> sold,
     vector<double> mPrice, vector<double> sPrice,
     int count );


int main()     //   function main begins program execution
{
     //   Declare variables
     char fileName[ 15 ];
     vector<int> itemID;
     vector<string> itemName;
     vector<int> ordered;
     vector<int> inStore;
     vector<int> sold;
     vector<double> mPrice;
     vector<double> sPrice;
     int count;
     char choice;

     //   let the user know about the program
     cout << "\n\n\tProgram that works with functions "
          << "\n\tVectors and files.";

     cout << "\n\n\tEnter the name of input file :  ";
     cin >> fileName;

     //   declare file stream & initialize with input file
     ifstream inFile( fileName );

     //   Display error when the file was not opened
     if ( !inFile )
     {
          cout << "\n\n\tError in opening input file";
          return 1;
     }    //   end if

     //   read the data from th einput file
     count = getInput( inFile, itemID, itemName, ordered,
                         inStore, sold, mPrice, sPrice );

     do
     {
          choice = displayMenu();

          if ( choice == '1' )
               //   sell an item
               sellAnItem( itemID, inStore, sold, count );

          if ( choice == '2' )
               //   print the report
               printReport( itemID, itemName, ordered,
                    inStore, sold, mPrice, sPrice, count );

     } while ( choice != '3' );

     //   close the file stream
     inFile.close();
     inFile.clear();

     return 0; // indicate program executed successfully
}              // end of function, main

//   returns number of items
int getInput( istream& inp, vector<int>& itemID,
     vector<string>& itemName, vector<int>& ordered,
     vector<int>& inStore, vector<int>& sold,
     vector<double>& mPrice, vector<double>& sPrice )

{
     //   declare variables
     int myInt, count = 0;
     string str = "";
     char ch;
     double myDouble;

     //   read the input from the file
     do
```

Comment

Step 2 of 3

```
          //   sentinal condition
          if ( ( inp >> myInt ) == NULL )
               break;
          itemID.push_back( myInt );

          //   read and store the itemName from the file
          inp.get( ch );
          getline( inp, str );
          itemName.push_back( str );

          //   read and store the order and stock
          inp >> myInt;
          ordered.push_back( myInt );
          inStore.push_back( myInt );
          sold.push_back( 0 );

          //   read and store manufacturer's price
          inp >> myDouble;
          mPrice.push_back( myDouble );

          //   read and store sales-proce
          inp >> myDouble;
          sPrice.push_back( myDouble );

          count++;
     } while ( true );    //   end do-while

     //   sort the entries in alphabetical order
     sortInput( count, itemID, itemName, ordered,
                    inStore, sold, mPrice, sPrice );

     return count;  //   return the count to the user
}    //   end function getInput

//   returns the sorted data
void sortInput( int count, vector<int>& itemID,
     vector<string>& itemName, vector<int>& ordered,
     vector<int>& inStore, vector<int>& sold,
     vector<double>& mPrice, vector<double>& sPrice )

{
     //   declare variables
     int myInt, min;
     string str = "";
     double myDouble;

     //   process for sort the entries
     for ( int i = 0; i < count - 1; i++ )
     {
          min = i;

          //   find entry which was need to be sorted
          for ( int j = i+1; j < count; j++ )
               if ( itemName[j].compare(itemName[min]) < 0)
                    min = j;

          //   sort all the entries
          if ( min != i )
          {
               myInt = itemID[ i ];
               itemID[ i ] = itemID[ min ];
               itemID[ min ] = myInt;

               itemName[ i ].swap( itemName[ min ] );

               myInt = ordered[ i ];
               ordered[ i ] = ordered[ min ];
               ordered[ min ] = myInt;

               myInt = inStore[ i ];
               inStore[ i ] = inStore[ min ];
               inStore[ min ] = myInt;

               myInt = sold[ i ];
               sold[ i ] = sold[ min ];
               sold[ min ] = myInt;

               myDouble = mPrice[ i ];
               mPrice[ i ] = mPrice[ min ];
               mPrice[ min ] = myDouble;

               myDouble = sPrice[ i ];
               sPrice[ i ] = sPrice[ min ];
               sPrice[ min ] = myDouble;
          }    //   end if
     }    //   end for
}    //   end function sortInput

//   displays a menu
char displayMenu()
{
     char choice;

     cout << "\n\n\t\t\tM  E  N  U  ";
     cout << "\n\t\t-----------------------";
     cout << "\n\n\tSell an item       :    1";
     cout << "\n\tPrint a report    :    2";
     cout << "\n\tQuit              :    3";
     cout << "\n\n\tEnter your choice :   ";
```

Comment

Step 3 of 3

```
     cin >> choice;

     return tolower( choice );
}    //   end function displayMenu

void sellAnItem(vector<int>& itemID, vector<int>& inStore,
                         vector<int>& sold, int count )
{
     int i, n, quantity;

     //   prompt and read an item number to sell
     cout << "\n\n\tEnter item-ID number :  ";
     cin >> n;

     //   search for the item
     for ( i = 0; i < count; i++ )
          if ( itemID[ i ] == n )
               break;

     //   sell the item
     if ( i != count )
     {
          //   prompt and read the quantity
          cout << "\tEnter quantity :  ";
          cin >> quantity;

          //   check for valid quantity
          if ( quantity > inStore[ i ] )
               cout << "\n\tStock does not available.";
          else
          {
               //   sell the quantity
               inStore[ i ] -= quantity;
               sold[ i ] += quantity;
               cout << "\n\t" << quantity
                         << "pieces sold out.";
          }    //   end inner else-if
     }
     else //   print error message
          cout << "\tSpecified item was not found in the store";

}    //   end function sellAnItem

//   print the report to the manager
void printReport( vector<int> itemID,
     vector<string> itemName, vector<int> ordered,
     vector<int> inStore, vector<int> sold,
     vector<double> mPrice, vector<double> sPrice,
     int count )
{
     //   declare variables
     double inventory = 0;
     int items = 0;

     //   print the captions in the report
     cout << "\n\n\t\t\tFriendly Hardware Store";
     cout << "\n\nItemID   Item Name          "
               << "pOrdered  pInStore  pSold  "
               << "manufPrice  sellingPrice";

     //   set the floating point to
     //   2 decimal places fixed decimal point notation
     cout << fixed;
     cout.precision( 2 );

     //   print the data from the vectors
     for ( int i = 0; i < count; i++ )
     {
          inventory += inStore[ i ] * sPrice[ i ];
          items += inStore[ i ];
          cout << "\n"
               << setw( 6 ) << right << itemID[ i ] <<"    "
               << setw( 15 ) << left << itemName[ i ]
               << setw( 10 ) << right << ordered[ i ]
               << setw( 10 ) << right << inStore[ i ]
               << setw( 7 ) << right << sold[ i ]
               << setw( 12 ) << right << mPrice[ i ]
               << setw( 13 ) << right << sPrice[ i ];
     }    //   end for

     //   print the summary results
     cout << "\n\nTotal Inventory:  $ " << inventory;
     cout << "\nTotal number of items in the store: "
               << items;
}    //   end function printReport
```

Comment

Was this solution helpful? 1 0