Sheharyar Khan

Lab 7 SQL

CS-116

Source Code

wine.h

```
#ifndef CCC_WINE_H
#define CCC_WINE_H

using namespace std;

class Wine
{
private:
//this is data information with five items winename, vintage, score, price and winearyname,
// I am not clearly understand winerayname, so I use the initializes of words from the wine information web you give us.
        string wineName;
        int vintage;
        int score;
        double price;
        string type;
public:
//this is constructs a defalut wine and a wine with price and name and a wine consturctor with name vintage score price and winearyname.
        Wine();
        Wine(double wine_price, string wine_name);
        Wine(string wine_name, int wine_vintage, int wine_score,
        double wine_price, string type);
//this is get in this wine object
        string get_winename() const;
        int get_vintage()const;
        int get_score()const;
        double get_price()const;
//this is set of this wine object
        void set_winename(string wine_name);
        void set_vintage(int wine_vintage);
        void set_score(int wine_score);
        void set_price(double wine_price);
                void printWineInfo() const;

};

void printMeFirst(){
            cout<<endl;
            cout<<"Program written by - Sheharyar Khan"<<endl;
            cout<<"Course Info - CS-116"<<endl;
            time_t now = time(0);
            char* dt = ctime(&now);
            cout<<" Date: "<<dt<<endl;
}

#endif
```

winedb.cpp

```
/*

http://www.codingfriends.com/index.php/2010/02/17/mysql-connection-example/

To compile up this program you will need to link to the mysql libraries and headers that are used within the program, e.g. mysql.h at the top of the program. To gain access to these, there is a nice mysql_config (you may need to install it via your package manager
```

system if you do not have it already).

Here are my outputs of what is required on the command line for the g++ compiler

```
g++ -I/usr/include/mysql winedb.cpp -o winedb -L/usr/lib/mysql -lmysqlclient


./windb "select * from wineInfo where price > 100"

*/

#include <mysql.h>
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <iomanip>
#include <vector>
#include "dbconnect.h"
#include <stdlib.h>  // atoi, atof

using namespace std;

#include "wine.h"
#include <iomanip>

using namespace std;


/*
 argv[1] - put sql command in argv[1], otherwise, just
 use sql "show tables"
*/
int main(int argc, char* argv[])
{

  vector<Wine> wineList;

                 int num=0;
                 double totalPrice=0;

  MYSQL *conn;                      // the connection
  MYSQL_RES *res;// the results
  MYSQL_ROW row;                    // the results row (line by line)

  struct connection_details mysqlD;
  mysqlD.server = (char *)"localhost";  // where the mysql database is
  mysqlD.user = (char *)"root";                 // the root user of mysql
  mysqlD.password = (char *)"password"; // the password of the root user in mysql
  mysqlD.database = (char *)"mysql";      // the databse to pick

  // connect to the mysql database
  conn = mysql_connection_setup(mysqlD);
t my name and classo

  // assign the results return to the MYSQL_RES pointer
```

```
if (argc < 2)
{
  cout << "argv[0]: " << argv[0] << endl;
  printf("\nUsage: %s  \"SQL statement here\"\n", argv[0]);
  printf("if no argument given, default is:\n %s show tables\n", argv[0]);
  res = mysql_perform_query(conn, (char *)"show tables");
  printf("MySQL Tables in mysql database:\n");
}
else
{
  // use wine database
  res = mysql_perform_query(conn, (char *)"use wine");
  cout << "argv[0]: " << argv[0] << endl;
  cout << "argv[1]: " << argv[1] << endl;
  res = mysql_perform_query(conn, argv[1]);


  while ((row = mysql_fetch_row(res)) !=NULL)
  {

           num++;
   if (argc < 2) {
     printf("%s\n", row[0]);  // only print out 1st column
   }
   else
   {
                      /* print out each row of the data extracted from
                       * MySQL database
                       * Make sure the output is line up with the header
                       * Hint: use left and setw
                       * WineInfo table Fields:
                       *   Table Field 1 - row[0] - name (wine name price)
                       *   Table Field 2 - row[1] - vintage (year of the wine)
                       *   Table Field 3 - row[2] - score (rating of the wine by experts)
                       *   Table Field 4 - row[3] - price (price of wine)
                       *   Table Field 5 - row[4] - wineryID (index to winery table)********************
                       *   Table Field 6 - row[5] - type (red, white)
                       *   Table Field 7 - row[6] - location (where is the wine from)******************
                       *   Table Field 8 - row[7] - UPC (wine product code)*****************************

            * Add your code to put the wine into wineList vector
            *
            * You need to define Wine wine variable and use constructor
            * to initialize name, vintage, score, price and type
            * ????
            * atoi(string) - convert string to integer
            * atof(string) - convert string to double
            * Wine wine (row[0], atoi(row[1]), ??????????);
            * wineList.??????(?????);  put wine into the wineList vector
            */
                    Wine wine(row[0],atoi(row[1]),atoi(row[2]),atof(row[3]),row[5]);//makes a wine object with the desired properties
            /*??? put wine into wineList vector ??? */
                    wineList.push_back(wine);//puts the wine in the wineList vector

   } /* end else */
   /*
    * number of wine retrieved from mySQL and average price
```

```
     */
     totalPrice+=atof(row[3]);
  } /* end of while my_sql_fetch */

/* clean up the database result set */
mysql_free_result(res);
/* clean up the database link */
mysql_close(conn);


/*
 * Using while loop (or do-while) to implement user menu to
 * print out information from wine vector
 */


 cout <<"\nPrint out wine vector\n\n";
                     cout << setw(30) << "Wine Name"
         << setw(16)          << "Vintage"
                                  << setw(12)                              << "Score"
                                  << setw(12)                              << "Price"
                                  << setw(10)                              << "Type"
              << endl;

   cout << setw(30) << "---------"
         << setw(16)          << "-------"
                                  << setw(12)                              << "------"
                                  << setw(12)                              << "-----"
                                  << setw(10)                              << "----"
              << endl;



 num=0;
 //totalPrice=0;

 /*
  * Add code below to print out all the wine info from
  * wineList using for loop to traverse the wineList vector
  *

  * for (int unsigned i=0; i < ?????? ; i++)
  *
  * As you print out each wineInfo from the weneList using
  * printWineInfo(), you should
  * also add the totalPrice for all the wines in the vector so you
  * can calculate average wine price
  */
             for(unsigned int i=0;i<wineList.size();i++){//iterates over the vector list
             wineList[i].printWineInfo();//prints out the wine info
             }

 cout <<"Total number of wine found: " << wineList.size() << endl;
 cout <<"Average price: $"<< fixed << setprecision(2)
                            << totalPrice/wineList.size() << endl;
}
```

```
  return 0;
}
```

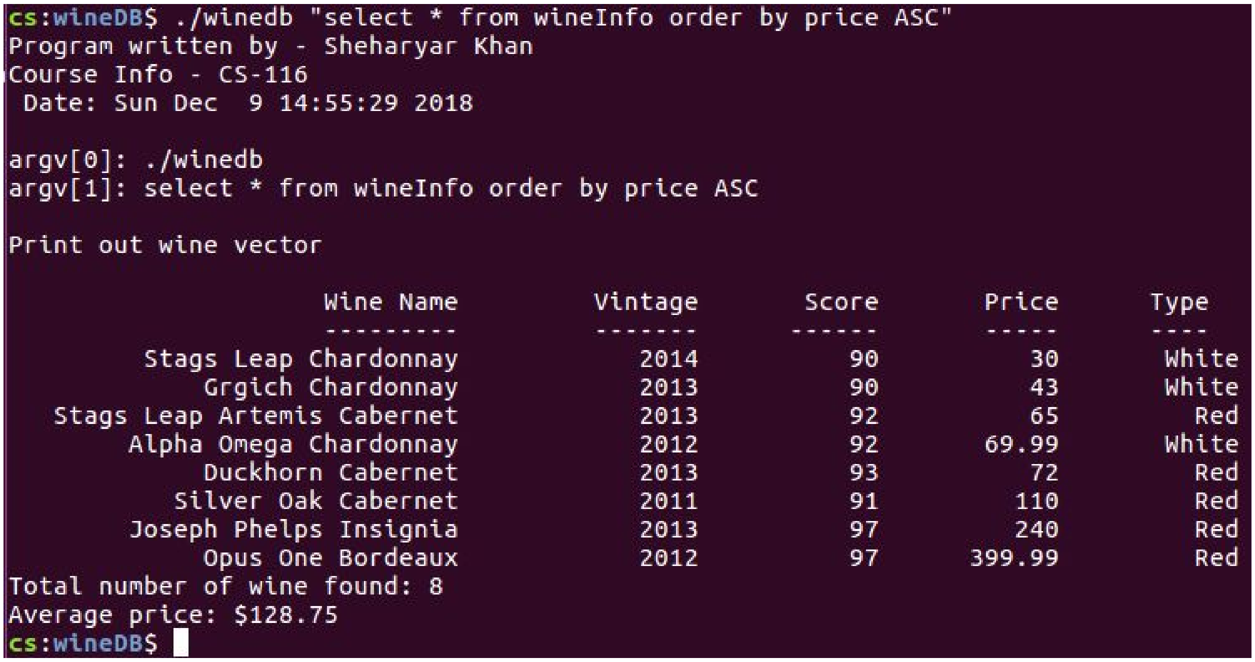
wineInfo.cpp

```
#include <stdio.h>
#include <iomanip>
#include "wine.h"
#include <iostream>

using namespace std;


Wine::Wine()
{
        price = 0;
}
//constructs a wine
Wine::Wine(double wine_price, string wine_name)
{
        price = wine_price;
        wineName = wine_name;
}
Wine::Wine(string wine_name, int wine_vintage, int wine_score,
  double wine_price, string wine_type)
{
        /* set class wine attributes to the parameters
         * ????????
         * wineName = wine_name;
         * vintage = ????
         * */
                wineName=wine_name;//assigns the value of wine_name to wineName
                vintage=wine_vintage;//assigns the value of wine_vintage to vintage
                score=wine_score;//assigns the value of wine_score to score
                price=wine_price;//assigns the value of wine_price to price
                type=wine_type;//assigns the value of wine_type to type
}
string Wine::get_winename() const
{
        return wineName;
}
void Wine::set_winename(string wine_name)
{
        wineName = wine_name;
}
int Wine::get_vintage()const
{
        return vintage;
}
void Wine::set_vintage(int wine_vintage)
{
        vintage = wine_vintage;
}
int Wine::get_score()const
{
        return score;
}
void Wine::set_score(int wine_score)
```

```
{
        score = wine_score;
}

/*
 * return wine price
 * */

double Wine::get_price()const
{
        return price;
}

/*
 * set wine price
 * */
void Wine::set_price(double wine_price)
{
        price = wine_price;
}

/*
 * print out wineInfo wineName, vintage, score, price and type
 * */

void Wine::printWineInfo() const
{
        /* implement this function.
         * Use setw to format the output
         * ????????
         * cout << setw(30) << wineName ??????
         * ????? */
                cout<<setw(30)<< wineName //couts the wineName and sets the width of the column to 30 characters
                <<setw(16)<<vintage//couts the vintage and sets the width of the column to 16 characters
                <<setw(12)<<score//couts the score and sets the width of the column to 12 characters
                <<setw(12)<<price//couts the price and sets the width of the column to 12 characters
                <<setw(12)<<type<<endl;//couts the type and sets the width of the column to 12 characters
}
```

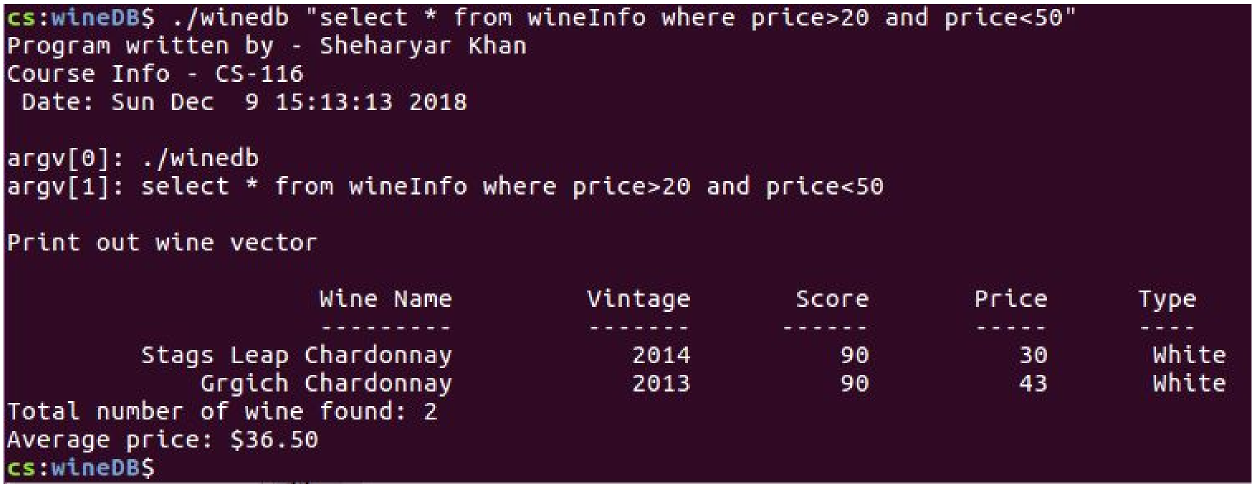
Test Cases:

All the Wines sorted by price in ascending order:



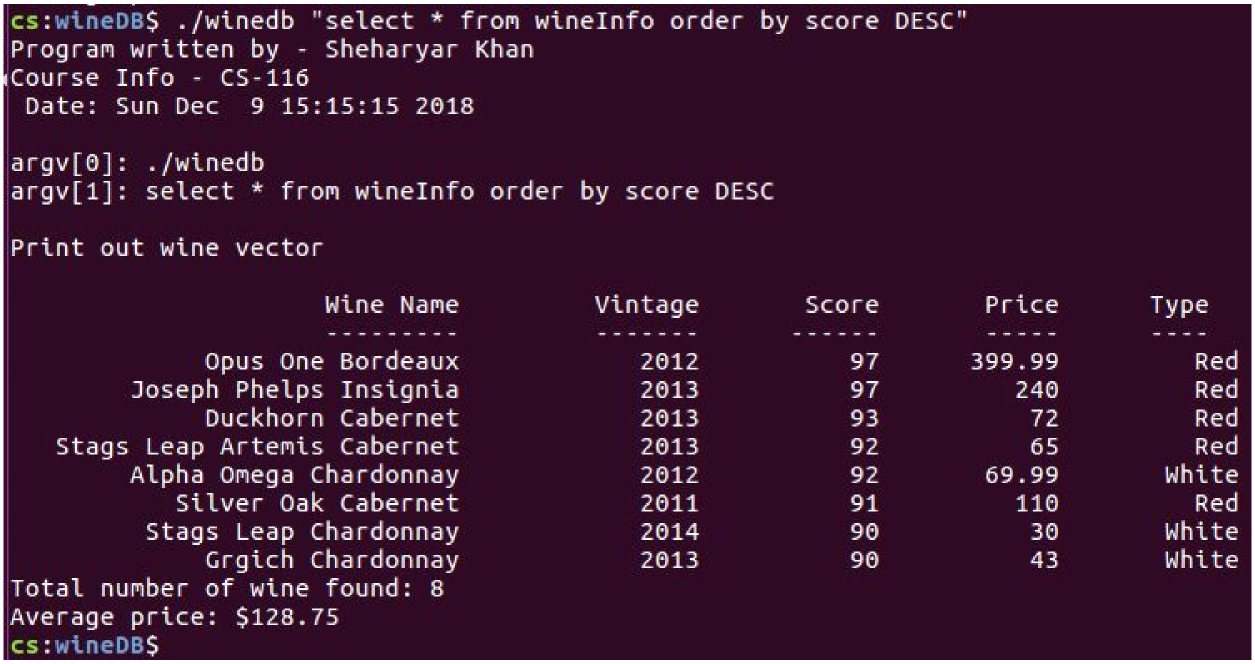
```
cs:wineDB$ ./winedb "select * from wineInfo order by price ASC"
Program written by - Sheharyar Khan
Course Info - CS-116
 Date: Sun Dec  9 14:55:29 2018

argv[0]: ./winedb
argv[1]: select * from wineInfo order by price ASC

Print out wine vector

                     Wine Name         Vintage       Score       Price      Type
                     ---------         -------      ------       -----      ----
         Stags Leap Chardonnay            2014          90          30       White
             Grgich Chardonnay            2013          90          43       White
   Stags Leap Artemis Cabernet            2013          92          65         Red
        Alpha Omega Chardonnay            2012          92       69.99       White
             Duckhorn Cabernet            2013          93          72         Red
           Silver Oak Cabernet            2011          91         110         Red
        Joseph Phelps Insignia            2013          97         240         Red
             Opus One Bordeaux            2012          97      399.99         Red
Total number of wine found: 8
Average price: $128.75
cs:wineDB$ 
```

All the wines with a price between 20 and 50:



```
cs:wineDB$ ./winedb "select * from wineInfo where price>20 and price<50"
Program written by - Sheharyar Khan
Course Info - CS-116
 Date: Sun Dec  9 15:13:13 2018

argv[0]: ./winedb
argv[1]: select * from wineInfo where price>20 and price<50

Print out wine vector

                     Wine Name         Vintage       Score       Price      Type
                     ---------         -------      ------       -----      ----
         Stags Leap Chardonnay            2014          90          30       White
             Grgich Chardonnay            2013          90          43       White
Total number of wine found: 2
Average price: $36.50
cs:wineDB$
```

All the wines in descending order by score:



```
cs:wineDB$ ./winedb "select * from wineInfo order by score DESC"
Program written by - Sheharyar Khan
Course Info - CS-116
 Date: Sun Dec  9 15:15:15 2018

argv[0]: ./winedb
argv[1]: select * from wineInfo order by score DESC

Print out wine vector

                     Wine Name         Vintage       Score       Price      Type
                     ---------         -------      ------       -----      ----
             Opus One Bordeaux            2012          97      399.99         Red
        Joseph Phelps Insignia            2013          97         240         Red
             Duckhorn Cabernet            2013          93          72         Red
   Stags Leap Artemis Cabernet            2013          92          65         Red
        Alpha Omega Chardonnay            2012          92       69.99       White
           Silver Oak Cabernet            2011          91         110         Red
         Stags Leap Chardonnay            2014          90          30       White
             Grgich Chardonnay            2013          90          43       White
Total number of wine found: 8
Average price: $128.75
cs:wineDB$
```