This project is a group project. List Group members below. Group #____________

Print first and last names  Section #  e-mail address
1.
2.
3.

Please provide brief description of how each member of the group participated in the project.
1.

2.

3.

Please list all sources: web pages, people, books or any printed material, which you used to prepare your report and implementation of the code for the project (Note: the boxes expands so that you can list many sources).

<table>
<thead>
<tr>
<th>Types of Sources</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td></td>
</tr>
<tr>
<td>Web Material (give URL)</td>
<td></td>
</tr>
<tr>
<td>Printed Material</td>
<td></td>
</tr>
<tr>
<td>Other Sources</td>
<td></td>
</tr>
</tbody>
</table>

I certify that I have listed all the sources that I used to develop solutions to the submitted project report and code, and that I did not plagiarize.

<table>
<thead>
<tr>
<th>Your signature</th>
<th>Last 4 Digits of UIN</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Group Project 1 (200 pts) Online Store
Assigned February 24, 2010. Due Apr 2, 2010 by 5:00pm.

Please see the course web page for the group assignments

Objective: In this group project, you must write a C++ code that completes the class design of an existing C++ interface for an online store. Note that “std_lib_facilities.h” should not be used. Also, test your program to ensure it works.

General Guidelines

A copy of your project report and electronic submission of your source code are due March 24, 2010 on CSNET by 5:00pm. Please return a hard copy report to your TA.

This project must be done in groups of three students. See the course page for the group assignments. Use the cover sheet above for your hard copy report.

The program should be coded in C++ and has to be well documented (see PPP §7.6.4). The online store interface code is in the zip file online_store.zip, which can be downloaded from the course website. The zip file also includes sample input files to use in developing your program.

Make sure your final program can be compiled using g++ before submission because your program will be tested on a CS UNIX system.

Provide all tests done to verify correctness of your program in your report. Give an explanation why you chose such tests.

The assignment will be graded focusing on: program design, correctness and provided report. Your program will be tested on TA’s input files.

Format of three store data files (N is the number of records of each type):

Filename: catprd.dat
    N [Category_ID Cat_name]. . .
    N [Product_ID Category_ID Prod_name Prod_price]. . .

Filename: cust.dat
    N [Customer_ID Cust_name Cust_addr]. . .

Filename: carts.dat
    N [Cart_ID Customer_ID Cart_date]. . .
    N [Cart_item_ID Cart_ID Product_ID Prod_quantity]. . .

All ID fields are at least four digit numbers. All name fields are a single word. The address field is a street number followed by a one word street name followed by a zip code. A date field is three numbers: year month day separated by a space.
Specific Instructions

1. First, each person should establish a build environment for the online store application. From the course website, download a zip file named online-store.zip, which contains the source code for establishing a build project under Visual Studio C++. You will complete the program in a series of increasingly difficult steps. First, make sure you can build, link and run the distributed program without making any modifications to the source code. This is your starting point. To get credit on this step, show your TA that you can build and run the program on March 1, or 2, 2010 in lab.

2. Second, decide as a team who will do which problem in set: A. Individual Problems. Each of you should individually work your selected problem. Make your changes to the original source code. Add additional comments that include your initials and the problem solved (A.1, A.2, or A.3) on each line you add or change. Comment out any lines you delete, don’t just delete them. Test your code. Demonstrate your solution to your teammates and to your TA during a lab. Individually, submit your source code electronically on CSNET. After all three sub problems are submitted, work as a team to merge your solutions into a single source (which will be used in the next step). Turn in an interim team report and electronic submission of your combined team source code on CSNET by March 5th, 2010, 5pm. Please turn in the cover sheet and a hard copy of your interim team report to your TA. Your TA will check to see that all three problems are solved in the merged code and apply the grading rubric.

3. Third, decide as a team who will do what problems in set: B. Individual Problems. As before, each of you should work your problem. Start from the merged code turned in on Step 2. Add comments that include your initials, B.1, B.2, B.3, etc. Demonstrate your solution to your teammates and to your TA. Turn in your individual code on CSNET. Then merge your solutions into a single source. Test to make sure all the solutions in the combined source! Turn in an interim team report and electronic submission of your source code on CSNET by March 12th, 2010, 5pm. Again use the cover sheet for your hard copy interim team report. Your TA will check to see that all three problems are solved and apply the grading rubric.

4. Fourth, decide as a team who will do what problems in set: C. Individual Problems. You know what to do by now, but the problems are harder. Start from the merged code turned in on Step 3. Submit your individual problem source code on CSNET. Turn in a team report and electronic submission of your team’s merged source code on CSNET by March 26th, 2010, 5pm.

5. Finally, decide as a team which problem to solve in the group of D. Team Problems. Work from a single source (from step 4) and identify what source code changes you make using your initials, and problem solved—D.1, D.2 or D.3. Test your code and demonstrate the final results to your TA. Turn in a hard copy of your final program that will run on Unix, team report and electronic submission of your source code on CSNET by Apr 2, 2010, 5pm. Please turn in your final team report to your TA.
Summary of what you turn in and when:

Step 1: Turn in nothing, but show your TA on March 1, or 2, 2010 that you can build the online_store to get credit for this step.
Step 2: Turn in your individual source code on CSNET (no individual report in the submission). Turn in your merged team source code with a report on CSNET. Turn a hard copy of your team report to your TA. Do all this by March 5th, 2010, 5pm.
Step 3. Same as Step 2. Do all this by March 12th, 2010, 5pm.
Step 4. Same as Step 2. Do all this by March 26th, 2010, 5pm.
Step 5. Turn in your team source code and a final team report on CSNET. Turn in a hard copy of the report to your TA. Do all this by Apr 2nd, 2010 by 5:00pm.

A. Individual Problems for Step 2 (Pick one each)

1. When exit is selected, output a message and confirm that the user did not accidentally select exit. Create an Address type to be used in the Customer type. Complete the read() function in Menu by reading and storing the customer data.
2. The program should output its own identification information before displaying the initial main menu. Add the Date type from Chapter 9 to the project for use in the Cart type. Complete the read() function in Menu by reading and storing the cart and cart item data.
3. Complete the display_main_menu() and display_info() for the Menu type. Complete the read() function in Menu by reading and storing the category and product data.

B. Individual Problems for Step 3 (Pick one each)

Each one of you should select a unique (1) Show, (2) Find and (3) Update query (see last page of this document) and complete the appropriate functions in the Menu class.

C. Individual Problems for Step 4 (Pick one each)

1,2,3: Take two each of the Other queries given at the end of the document and code them. Honors students also complete the two delete queries.

D. Team Problems for Step 5 (Work all problems as a team):

1. In your merged code, error check the data read in step B to make sure no cart refers to a non-existent customer, no product refers to a non-existent category, and no cart item refers to a non-existent product, or cart. Verify category, cart, customer and product IDs are unique.
2. In your merged code, modify the Exit code added to main() to call a new function you add to Menu that writes out updated data files that reflect the new state of the data after a session.
ENGR-112 Spring 2009, Project 2 Individual Grading Rubric for each student

Group #: ___________  Student Names_________________________
_________________________
_________________________

Step 1: Build project (20 points)

Failure to get the online_store to build: -10
Failure to get online_store to link: -10
Failure to get the online_store to run after it is built: -10
(Note: at most 20 points will be deducted if the subtractions add up to more than 20).

Step 2: A. Individual Problems (20 points)

Failure to correctly implement one problem: -15
Failure to turn in individual source code: -10
Failure to document individual changes in source code: -5
Failure to build: -10, or link: -10, or run: -10 without terminating abnormally
(Note: at most 20 points will be deducted if the subtractions add up to more than 20).

Step 3: B. Individual Problems (30 points)

Failure to correctly implement one problem: -20
Failure to turn in individual source code: -15
Failure to document individual changes in source code: -10
Failure to compile: -15, or to link: -15, or run: -15 without errors
(Note: at most 30 points will be deducted if the subtractions add up to more than 30).

Step 4: C. Individual Problems (30 points)

Failure to correctly implement one problem: -20
Failure to turn in individual source code: -20
Failure to document individual changes in source code: -15

Failure to compile: -20, or to link: -20, or run: -20 without errors

(Note: at most 30 points will be deducted if the subtractions add up to more than 30).

**Individual Project Grade:** ___ / 100

**Overall Project Grade is the sum of Individual and Team Grades:** ___ / 100 + 100
ENGR-112 Spring 2009, Project 2 Team Grading Rubric for Overall Team

Group #: __________  Student Names__________________________

Step 2: A. Individual Problems (15 points)

Failure to implement all three problems: -5

Failure to turn in merged source code: -10

Failure to turn in interim team report containing all sections: -10

Failure to document changes to merged source code: -5

Failure of merged source code to build without errors: -10, or link: -10, or run: -10

(Note: at most 15 points will be deducted if the subtractions add up to more than 15).

Step 3: B. Individual Problems (20 points)

Failure to implement all three problems: -10

Failure to turn in merged source code: -10

Failure to turn in interim team report containing all sections: -10

Failure to document changes to merged source code: -5

Failure of merged source code to build without errors: -10, or link: -10, or run: -10

(Note: at most 20 points will be deducted if the subtractions add up to more than 20).

Step 4: C. Individual Problems (30 points)

Failure to correctly implement all three problems: -20

Failure to turn in merged source code: -20

Failure to turn in interim team report containing all sections: -10

Failure to document changes in merged source code: -10

Failure of merged source code to compile: -20, or to link: -20, or run: -20
(Note: at most 30 points will be deducted if the subtractions add up to more than 30).

**Step 5: D. Team Problem (25 points)**

Failure to correctly implement either problem: -25

Failure to turn in final team report containing all sections: -20

Failure to document individual changes in final source code: -10 per missing member

Failure of source code to compile: -20, or to link: -20, or run: -20

(Note: at most 25 points will be deducted if the subtractions add up to more than 25).

**Team Project Grade: ___ / 100**
Queries

Format of the output results is specified in the queries.

I. Show queries (prompt the user to obtain the desired query):
   a. list all categories sorted by category ID [print name and number of products in each]
   b. list all customers sorted by name [print name, ID and Address]
   c. list all products sorted by price [print name, ID and price]

II. Find queries (prompt the user to obtain the desired query):
   a. Calculate total sales for a given category
   b. Calculate total sales for a given customer
   c. Display total sales for a given product

III. Update queries (prompt the user to obtain the desired query):
   a. Add a new cart item to an existing cart
   b. Add a new product to an existing category
   c. Add a new customer

   Other queries (I, II, and III) (prompt the user to obtain the desired query):
   a. list all cart IDs and checkout price sorted by date [print ID, date and price for each]
   b. list all product names and cost in a given cart by ID [print name and cost based on qty]
   c. Display the five top selling products by revenue regardless of category
   d. Display the cart with the largest check out total
   e. Add a new cart given a customer and a date
   f. Add a new category
   g. Honors students add a query that allows a user to delete a Cart_item from a cart, and delete a Product from a Category.

Sample Data Files

A sample of all three input files is included in the online-store.zip file. Note: that .dat files open with wordpad under Windows.

catprd.dat
cust.dat
carts.dat