CS 235 Midterm

Version .11

Instructor: R. P. Burton

May 31- June 3, 2013 (Friday through Monday)

due in the Lab on Monday not later than 7:00 p.m. on Monday

Penalty for submitting the midterm late:

20 points per day (including the weekend), advancing at 7:01 p.m. each day

Open Book (course text and your CS 142 course text only), Open Notes (including your own Lab solutions)

Open Secondary Storage Device: yours only

Open Laptop: if you wish

Open Course Website (and CS 142 course website) and the C++ API, but no other Internet resources (such as non API resources on the C++ website)

Closed Neighbor (and everyone is thy neighbor)

**\*Instructions\***

(Please read carefully)

1. This midterm consists of a C++ programming problem with optional extra credit. Read and understand the statement of the problem completely before beginning to design, code, and test. As part of your design, consider the test cases that will establish the correctness of your solution. Test your solution thoroughly before submitting it.
2. Produce a solution, which consists of your C++ code, with a comment at the beginning of each file (both .h and .cpp) which includes your name, your student ID number, and “CS 235 Spring 2013 Midterm.” Upload your completed project by compressing the files and submitting them through Gradebook with TA assistance. If a packet is not collected by a TA upon submission, your solution will not be graded and you will receive no credit for the exam. Attribute any code taken from or based on other sources (except for the course texts and the course websites). Attributed code copied from or based heavily on outside sources is worth half credit. Unattributed code copied from or based heavily on outside sources is worth no credit.
3. Understanding the problem correctly is part of the examination. If something seems unclear, ask a CS 235 TA for clarification. You may pose questions to the CS 235 TAs at any time. However, the TAs generally are not permitted to answer questions related to design, C++ implementation, debugging, or testing.
4. Prior to submitting your midterm, score it using the attached scoring sheet (this will help you maximize your points and will help us grade your exam accurately).
5. When you are finished, go to the course website and follow the link labeled “Submit Exam” in the Exam Menu.
6. Sign the Midterm Scoring Sheet to request that your midterm be graded and to certify that no unfair information related to the midterm has been received by you, either directly or indirectly, and that none will be conveyed by you. If we discover that you cheated or assisted someone in cheating, intentionally or unintentionally (including accidentally), your score for this exam may (and probably will) be rand() % 0.

We’re serious.

**Five for the Price of One**

Without using the C++ STL or any pre-defined data structures except class Array (which we do not recommend), implement ADT Stack, ADT Queue, ADT Deque, ADT IR\_Deque (input-resticted deque), and ADT OR\_Deque (output-restricted deque). All five ADTs need constructors and destructors, and need to provide the commonly associated insertions, removals and accesses and a test for equality using the “==” operator, and are implemented as template classes.

Provide an appropriate interface with a main which allows the user to run your test driver. Create your own “test driver” for this exam. Your test driver should check all equivalence classes and boundaries to verify the correctness of your implementation. The code for your test driver must be well documented; points will be based on how thoroughly your test driver tests and verifies your code. Your test driver must include, but is not limited to, creation and deletion of each structure, appropriate insertion and removal from each structure, and equality testing for each structure. Finally, your test driver must test your classes with different data types to verify that your templates are working. Your test driver should include detailed documentation as comments in the code, as well as appropriate prompts and notices in the test driver to inform the user about what is being tested at each point and the results of the tests.

**Extra Credit (5 points per activity, 10 points for completing both):**

* Design and implement your “test driver” so that the user can select, test, and verify any individual ADT or all the ADTS, reporting the results, and allowing the process to be repeated as many times as the user wishes.
* Provide class Y\_Vector with features comparable to those described above.

**Midterm Scoring Sheet**

**Student Grading TA Grading**

\_\_\_/ 10 pts \_\_\_/ 10 pts – ADT Stack

\_\_\_/ 4 pts \_\_\_/ 4 pts – Constructor and Destructor implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – All required class functions implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – ‘==’ operator

\_\_\_/ 10 pts \_\_\_/ 10 pts – ADT Queue

\_\_\_/ 4 pts \_\_\_/ 4 pts – Constructor and Destructor implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – All required class functions implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – ‘==’ operator

\_\_\_/ 10 pts \_\_\_/ 10 pts – ADT Deque

\_\_\_/ 4 pts \_\_\_/ 4 pts – Constructor and Destructor implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – All required class functions implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – ‘==’ operator

\_\_\_/ 10 pts \_\_\_/ 10 pts – ADT IR\_Deque

\_\_\_/ 4 pts \_\_\_/ 4 pts – Constructor and Destructor implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – All required class functions implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – ‘==’ operator

\_\_\_/ 10 pts \_\_\_/ 10 pts – ADT OR\_Deque

\_\_\_/ 4 pts \_\_\_/ 4 pts – Constructor and Destructor implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – All required class functions implemented

\_\_\_/ 3 pts \_\_\_/ 3 pts – ‘==’ operator

\_\_\_/ 50 pts \_\_\_/ 50 pts – Test Driving

\_\_\_/ 25 pts \_\_\_/ 25 pts – Thoroughly tests each of the 5 ADT’s base functionality

\_\_\_/ 10 pts \_\_\_/ 10 pts – Thoroughly tests each of the 5 ADT’s template use

\_\_\_/ 15 pts \_\_\_/ 15 pts – Comments documenting and justifying the completeness of the testing

\_\_\_/ 10 pts \_\_\_/ 10 pts – Extra Credit

\_\_\_/ 5 pts \_\_\_/ 5 pts – Test one or all and repeat

\_\_\_/ 5 pts \_\_\_/ 5 pts – Y\_Vector

\_\_\_/ 100 pts \_\_\_/ 100 pts – Total (before late penalties)

\_\_\_/ 100 pts \_\_\_/ 100 pts – Total (with late penalties)

Printed Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ BYU ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, 2013

(Signature) (Date)

\*signing here verifies that you have read and agree to all the provisions of the midterm

**This section to be filled out by TAs**

Day Submitted: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Received by T.A. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Days Late: \_\_\_\_\_

TA to Student Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Graded By:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_