CS 142 Final Exam

Version 0.8

Instructors: R. P. Burton, K. Seppi, and E. Child

Monday, December 16, 2013 thru Friday, December 20, 2013, 8:00 p.m.

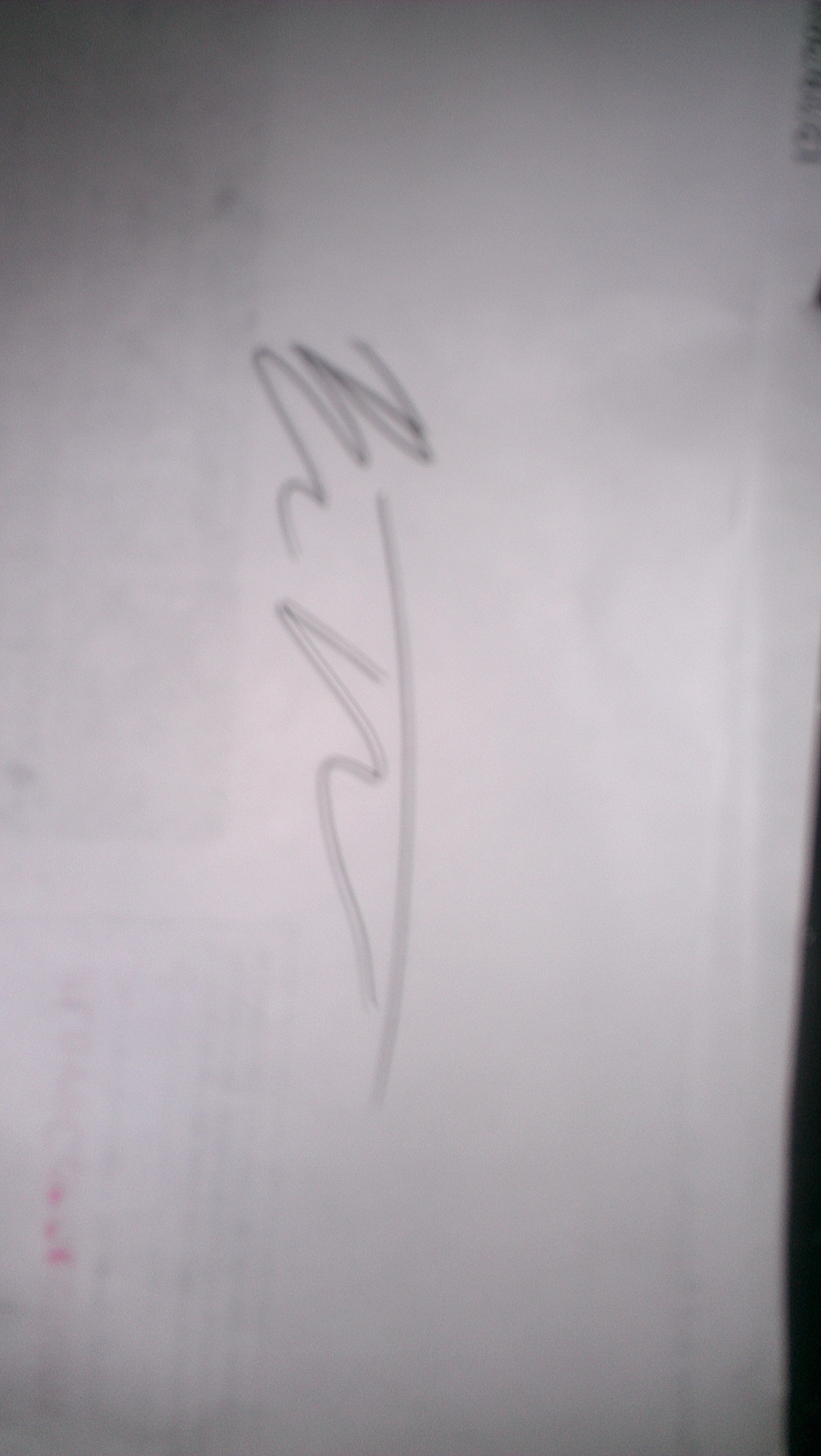
We will not be accepting late finals.

Allowed materials include your CS 142 course text, your own notes, your own prior lab solutions, the CS 142 course website, and cplusplus.com

Disallowed materials include all other text resources, all other Internet

resources, and all neighbors (remember, everyone is thy neighbor).

**Instructions**

1. This midterm consists of a C++ programming problem. Read and understand the statement of the problem completely before beginning to design, code, and test. Understanding the problem correctly is part of the examination. If something seems unclear, ask a CS 142 TA (but no one else) for clarification.
2. As part of your design, consider test cases that will establish the correctness of your solution.
3. Produce a solution, which consists of *your* C++ code, with a comment at the beginning of your file which includes your name, your student ID number, and “CS 142 Fall 2013 Final Exam.” Save your complete source file(s) using the online submission script. Attribute any code taken from or based on other sources (except for the course text and the course website). Attributed code taken from or based heavily on other sources is worth half credit. Unattributed code taken from or based heavily on other sources is worth no credit.
4. You may pose questions to the CS 142 TAs at any time. However, the TAs generally are not permitted to answer questions related to design, C++ implementation, debugging, or testing.
5. Code well. This includes choosing good names for identifiers, avoiding magic numbers, formatting in a communicative way so that your code is reasonably self-documenting, choosing good control structures, using appropriate data structures, encapsulating, using inheritance and polymorphism appropriately, etc. Provide comments where required or appropriate, especially to help the TAs understand what your code is intended to do. Remember, you will be graded not only on whether your code (when entered, compiled, and executed) accomplishes the specified task, but also on how clearly and efficiently it accomplishes the specified task, and on how closely you adhere to the stated requirements and to good programming practices.
6. When you are finished, score your solution as indicated on the Scoring Sheet. If you are within 15% of your score, you will not lose or gain extra points. If you are at within 16% to 25% of your score, you will lose 2.5% of your score. If you are not within at least 25% of your score, you will lose 5% of your score. After you complete the grading sheet, go to Learning Suite and follow the instructions under the Contents Tab in Exam Submission.
7. Sign here (1) to certify that you reviewed all your posted scores for the semester, notifying a CS 142 TA in writing to any discrepancies, (2) to request that your midterm be graded, and (3) to certify that no unfair information related to the midterm has been received by you, either directly or indirectly, and that none has been or 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() % 1. Discipline also may include Honor Code Office involvement, which can lead to loss of opportunities at Brigham Young University.

\_Benjamin Thompson\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12/20\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ 2013

(Name) (Date)

**Final Scoring Sheet**

Name:\_Benjamin Thompson\_\_ Section #:\_2\_ Day Submitted: \_12/20/13\_ T.A. Present \_\_\_\_\_\_\_\_

**Caution**: You will forfeit 10 points if your code has syntax errors and does not compile.

**Student TA Grading**

**\_30\_/50 \_\_\_\_/50 Inheritance and Polymorphism**

\_0\_/20 \_\_\_\_/ 20 Stores all Accounts in one vector in Bank

\_20\_/20 \_\_\_\_/ 20 Account types represent individual classes

\_10\_/10 \_\_\_\_/ 10 Account types inherit from one or more base classes

**\_0\_/10 \_\_\_\_/10 Opening Appropriate Accounts**

\_\_0\_/5 \_\_\_\_/ 5 Bank correctly opens accounts based on type of account

\_\_0\_/5 \_\_\_\_/ 5 Bank correctly does NOT add invalid account information to Bank

**\_0\_/15 \_\_\_\_/15 Appropriate Transactions**

\_0\_/5 \_\_\_\_/ 5 Accounts correctly withdraws/writes checks or rejects & applies

penalties, all according to account type

\_0\_/5 \_\_\_\_/ 5 Correctly advances month, adds/removes interest, adds appropriate

fees to accounts.

\_0\_/5 \_\_\_\_/ 5 Accounts correctly deposit amounts.

**\_0\_/10 \_\_\_\_/10 Closing Accounts Appropriately**

\_0\_/5 \_\_\_\_/ 5 Can close accounts only if balance is non-negative

\_0\_/5 \_\_\_\_/ 5 Closes accounts by removing from Bank

**\_5\_/5 \_\_\_\_/5 Created and included a test file (must be different from Test1.txt)**

**\_10\_/10 \_\_\_\_/10 Style and Organization**

\_5\_/5 \_\_\_\_/5 .ccp and .h files are separated

\_5\_/5 \_\_\_\_/5 Accounts hold appropriate variables

**\_0\_/5 \_\_\_\_/ 5 Extra Credit: Implemented a self-written sorting algorithm.**

**\_45\_/ 100 pts \_\_\_\_/ 100 pts TOTAL**

**\_\_\_\_/ 100 pts Adjusted TA Grader\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Student to TA*

Comments:\_Sorry, I had a horrible time trying to code this, I did a ton of this on a plane, because of the weather Thurday. I do feel like this final was cryptic in what was actually being looked for. I do not know how to store the accounts in a base class that doesn’t have any private members. I am not aware of a time that we ever talked about it in class, so that was my big hiccup. Hope you have a Good Holiday!\_

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