CSCE 155H/RAIK 183H – Program Grading Rubric

Student Name(s): ___________________________  Assignment 4  Total _____ / 100

CSE Login(s): ___________________________  Grader ___________________________

You should hand in all required files through webhandin as specified by the due date. Be sure to run the webgrader to ensure that your program(s) work as expected. Print a copy of this rubric, fill out your name(s), login(s), and the assignment number and turn it in at the start of class by the due date. Be sure to have followed all the assignment-specific instructions as specified in the assignment handout. Points will be awarded based on the following categories and items. The first three categories apply to all programs and parts; individual programs will be evaluated for design and correctness.

### Scores

<table>
<thead>
<tr>
<th>Item</th>
<th>Score</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Following Instructions</td>
<td></td>
<td>2.5</td>
</tr>
<tr>
<td>Style/ Documentation</td>
<td></td>
<td>2.5</td>
</tr>
<tr>
<td>Overall Design</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Program 1 Correctness</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Program 2 Correctness</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>Program 3 Correctness</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Program 4 Correctness</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

### Grader Comments
Following Instructions

- All required soft-copy files handed in via webhandin
- Correct file name(s) and organization
- Any required hardcopies (this rubric) handed in
- Programs successfully compile and execute using the webgrader

Style

- Appropriate variable and function/method identifiers
- Style and naming conventions are consistent
- Good use of whitespace; proper indentation
- Clean, readable code
- Code is well-organized

Documentation

- Well written comments that clearly explain the purpose of each non-trivial piece of code
- Comments explain the "what" and "why"
- Comments are not overly verbose or overly terse
- Code itself is "self-documenting"; explains the "how"

Program Design

- Code is well-organized and efficient
- Code is modular; substantial pieces of it could be reused; few redundancies
- Code is easily understood and maintainable
- It is clear that sufficient testing has been performed
- Corner cases and bad input have been anticipated and handled appropriately

Program Correctness

- Source code compiles, executes as expected
- Program runs as specified: correctly reads any input; correctly formatted output
- Test cases successfully execute