Assignment 1 Marking Scheme

Part A

======

Total marks: 100

For each incorrect answer to one subquestion, 5% will be deducted from part A.

For situations like repeated mistakes of the same type, the considered number of mistakes may vary to reduce the penalty for a single type of mistake.

For reasons like lack of explanation/unclear explanation, a deduction of marks may apply even though the results are correct.

Part B

======

Total marks: 100

Part B will first be marked regarding the correctness of the code, i.e. the number of test cases that failed in the testing. Correctness makes up 50% of part B. The remaining 50% are for other evaluation criteria (see below for details).

The following was used as an APPROXIMATE guide for the correctness grading for Assignment 1:

Total Failed Test Case Count Approximate Range

random

<table>
<thead>
<tr>
<th>Failed Test Cases</th>
<th>Approximate Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25/25</td>
</tr>
<tr>
<td>1-2</td>
<td>22/25</td>
</tr>
<tr>
<td>3-4</td>
<td>18/25</td>
</tr>
<tr>
<td>above 4</td>
<td>0-17/25</td>
</tr>
</tbody>
</table>
As required by the instructor, any code that cannot be run (due to a failed build) is deemed to fail all test cases. In cases where the code could not be tested (due to a failed build), correctness points cannot be given. Also, there is no excuse for the code not building, since each student is required to run the assignment_precheck script on their assignment submission prior to the deadline, and this would flag the presence of any build issues.

For the remaining 50% of the marks, issues found in the following categories will result in deductions of marks:

Bad code quality, or bad coding style.
Insufficient testing, poor test cases.
Vague commit history.
Lack of comments.