EECS3311 Fall 2019
Guide to Lab Test 1 (Section A and Section E)
When: Registered Lab Session on **Wednesday, October 2**

1 Format

- This in-lab programming test accounts for **5%** of your course grade.
- You need **not** study Lab 1 for this lab test.

  However, it is assumed that you are able to use similar constructs such as conditionals, loops, and `across` to write implementations and contracts.
- This test requires programming solutions **only**. No questions expecting written answers will be given.
  - An on-screen instruction sheet (in HTML or PDF) will be available in the beginning of the test.
  - You will be given a starter Eiffel project to start with.
  - You are required to develop your solution in this starter project using Eiffel Studio.
  - By the end of the test, you are required to submit the developed project (without the `EIFGENs` folder) via a terminal (like how you did in Lab 1).
- This programming test is meant for **assessing your ability to write Eiffel programs with valid implementation and contracts**.
- Therefore, your submitted project must **compile**: any syntax or type errors will lead to receiving a **zero** for this lab test.

2 Rules

- You must show up for your registered session only.
- There will be a **seating plan**: you must sit at your assigned seat.
- Bring a piece of photo ID (i.e., YU card or driver license).
- No mobile phone usage is allowed during the test.
- No data sheet will be allowed.
- The slides on Eiffel Syntax and Common Eiffel Errors will be made available to you.
- You may bring pen/pencil and a piece of blank paper for sketching your solutions.

3 Background Study

This in-lab programming test assumes that you have **competence** on the materials covered in:

- Tutorial series on DbC and TDD (from Lab 0):

  https://www.youtube.com/playlist?list=PL5dxAmCmjv_6r5VfzCQ5bTznoDDgh__KS

- Lecture on Writing Complete Postconditions.
- Use of the (nested) across syntax to write contracts.
- Use of the ARRAY and LINKED_LIST library classes (creation, queries, and constructors).
- Writing tests for your implementation

4 Practice Questions

This exercise is only meant as an example: you should prioritize on mastering the background study.

https://www.eecs.yorku.ca/~jackie/teaching/lectures/2019/F/EECS3311/exercises/EECS3311_F19_Labtest_1_Exercise.pdf