

EECS4315 (Section Z) Winter 2023
Guide to Programming Test 2
WHEN: 18:30 to 19:30, Tuesday, April 4
WHERE: William Small Centre (WSC) 108

CHEN-WEI WANG

Last Updated: March 29, 2023

1 Policies

- This programming test is in-person and **strictly** individual: plagiarism check may be performed and suspicious submissions will be reported to Lassonde for **a breach of academic honesty**.
- This programming test will account for **10%** of your course grade.
- This test is **purely** a programming test, assessing if you can write **valid** PlusCal algorithms and assertions **free of** syntax, type, and logical errors.
- **Structure of the Test:**
 - At 18:30 on the test day, all WSC machines will be rebooted to the “lab-test mode” (where there is **no** network connection and you are expected to use the **TLA+ toolbox only**).
 - During the test, you will be expected to:
 - * Launch TLA+ toolbox on a designated workspace.
 - * Create a list of (two or three) modules with the instructed names.
 - * For each module:
 - Implement an algorithm for the given problem, with the **required** input and output variables.
Caveat. It is absolutely critical for you to use the **exact names** of input and output variables as required by the test instructions (otherwise, grading assertions using these names will fail to work on your submitted modules).
 - Specify assertion(s) that correctly (and completely) formulate the described preconditions and/or postconditions of the algorithm.
 - Auto-translate the written PlusCal algorithm and assertions into TLA+ syntax, and use the TLC checker to verify the algorithm against the assertions you write.
Caveat. It is absolutely critical for you to be sure that all loops you write **terminate**; otherwise, a non-terminating loop may cause the TLC checker to hang indefinitely, and you will need to waste time to kill the process and re-start the tool (or to even re-start).
 - * You are **solely responsible** for:
 - **leaving enough time (≈ 3 minutes) to export the completed .tla module files and upload/submit them to WebSubmit;** and
 - **submitting the right module files for grading.**

– **Submission for Grading:**

- Like your assignments, submission (of module `.tla` files) for this programming test must be through the WebSubmit link (which will be provided during the test).
- It is your sole responsibility for making sure that the correct version of each module files is submitted. **After** clicking on the **submit** button on WebSubmit, you should **re-download** the module files and make sure that they are the right ones to be graded. **No** excuses or submissions will be accepted after your attempt times out.

– **Grading Criteria**

In each required module file:

- If your submitted file, loaded in the TLA+ toolbox and auto-translated to the TLA+ model, contains any errors (due to e.g., syntax errors, type errors), the TA will attempt to fix the errors for you, if they can be fixed quickly. If succeeded, there will be a *20% penalty* on the allocated marks for that module.
- If the input and output variables you use are **not** exactly as instructed, the TA will attempt to fix this for you. If succeeded, then there will be a *10% penalty* on the allocated marks for that module.
- To assess the correctness of your **PlusCal** algorithm, we will check it against (automatically using the TLC checker) our assertions (which are consistent and complete with respect to the problem descriptions give to you).
- To assess your specified assertions, we will:
 - * grade them manually for its completeness (i.e., whether or not there are missing cases) and correctness (i.e., whether or not the logic is correct); **and/or**
 - * grade them automatically by replacing your algorithm with an incorrect one (and see if your assertions would fail as expected) and/or a correct one (and see if your assertions would pass as expected).

2 Format

The format of this programming test will be identical to that of your **Lab2**: given informal problem descriptions (on the required inputs, outputs, and input-output relations), implement **PlusCal** algorithms and specify the appropriate assertions (which formulate the algorithm's preconditions and/or postconditions).

- As a reminder of the basic syntax, the following document will be made available to you during the test:

<https://d3s.mff.cuni.cz/f/teaching/nswi101/old/pluscal.pdf>

You're advised to go over the above document prior to the test so that you can easily find what you need during the test.

- You will have access to the tool for generating a state graph.
- You will be expected to create models (of modules) and verify their correctness (via the TLC checker). To create models (by instantiating constants), see Lab1.

3 Coverage for the Test

- Lab1
- Lab2
- Lab3 will not be covered.
- You do not need to review lecture materials.

4 Practice Questions

- By the end of Friday, March 31, some practice questions will be made available on the *Section Z eClass site*. Solutions to the practice questions will be made available by early Monday, April 3.
- This practice test will **not** be graded, but you may practice submitting it.
- It is important to note that these questions are meant for familiarizing yourself with the **format** and **workflow** of the test, and they represent **only** as an example: you are expected to study **all** materials as listed in Section 3.
- For extra practice, you may want to find problems at the similar level difficulty as in EECS1021/EECS1022.

5 Simulating the Programming Test

It is highly recommended that you simulate taking the programming test by following these steps:

Preparation

- Login into a machine under remotelabs (using your EECS account): <https://remotelab.eecs.yorku.ca/>. Choose a machine under the **ea** category.
- Launch the Firefox web browser (under Activities) and login into the Section Z eClass site.
- Download and open the **PracticeTest1.pdf** file from eClass onto the Desktop.

Start the Test

- Start a timer (say for 60 minutes).
- Launch TLA+ toolbox (under Activities)
- Tackle the test by: creating the modules as instructed, implementing algorithms, and specifying assertions.
- **Before you submit, you should make sure that there is no error in any of the module files.**

Submission

- **It is a recommended practice that you submit intermediate versions of your developed modlues (e.g., every 15 to 20 minutes).**
- Upload all required **.tla** module files to the WebSubmit link for grading:

<https://webapp.eecs.yorku.ca/submit/?acadyear=2022-23&term=W&course=4315Z&assignment=PT2>