Purpose of Tutorials

➢ Review some concepts from lectures
➢ Cover new materials that may be helpful for assignments
➢ Code demos to help understand certain concepts better

Let us know how we can improve the Tutorials; we appreciate your feedback!
Topics for this Tutorial

➢ Section 1: Setting Up

➢ Section 2: Software Tests

➢ Section 3: produceOutputs and runSuite
Section 1: Connecting to the Linux Environment
Connecting to the Linux Environment

➢ Please follow the document posted on the course website

➢ If you have any problem connecting to the CS Linux server, please raise your hand
Clone the Git Repository

- `git clone ssh://linux.student.cs.uwaterloo.ca/u/cs247/pubrepo/1225/.git`
Section 2: Software Tests
What are Software Tests

➢ Verify if the software works as intended

➢ Essential part of software development

➢ It is as important as writing code
Approaches for Testing - Human testing

➢ A person manually verifies if the software is working as intended. For example:
  - by inspecting the code and looking for flaws,
  - or by walking through the software operation as a user and verifying the results.

➢ Difficult to scale human testing for large software
Automated (machine) testing

➢ Test suites are implemented that automatically test the software and compare the results with the expected ones.

➢ Contain a list of input sets and matching expected outputs.

➢ Running the program with each input set and checking the actual output against the expected output.

➢ Human developer can later identify the errors.
Testing is not Debugging!

**Testing** is the process that identifies the errors but does not solve them.

**Debugging**, on the other hand, is the process that identifies the causes of known errors to solve them.
Section 3: produceOutputs and runSuite
Invoking the program

➢ ./produceOutputs suite.txt ./myprogram
➢ ./runSuite suite.txt ./myprogram
Now, it is time for you to attempt Question 1!