1 Overview

For this first tutorial, you will get familiar with the development environment (Question 0) and the procedures of testing and implementing a simple C++ program (Question 1).

**Allowed headers for this task:** For this task you may only use the headers `<iostream>`. Inclusion of any other headers (not created by out) may result in a loss of some or all of your marks.

2 Question 0: Setup

This course will be using a public git repository to help with disseminating resources to you. This question is to help you access those materials and make sure you have done so. The process is very simple, but it must be done from your student.linux.cs.uwaterloo.ca account.

2.1 CS undergraduate environment

Whenever you are asked for a “host name”, please use linux.student.cs.uwaterloo.ca instead of the name of a specific machine. CSCF periodically upgrades their servers, which changes the names. It also means that you’ll get assigned to a machine so that the system loads are balanced, which hopefully means that your response time will be fast.

- To log into the CS undergraduate environment, you need to set up a password that is separate from your WatIAM/Quest password.
  - Go to [https://www.student.cs.uwaterloo.ca/password/](https://www.student.cs.uwaterloo.ca/password/) to set up your password.
  - Read the instructions for choosing a password, then type your password in the first box. When you are finished typing in the box, read the message to the right of the box to determine if your password is acceptable according to the requirements.
  - If it is not, you must try again until you get “Looks OK!”.
  - Retype your new password in the second box. When you get “Looks OK!” you can save your password by clicking the “Save” button.

- Connecting to the undergraduate environment requires good internet access (and can sometimes be a little slow) but it has several benefits:
  - Regular (hourly, nightly, weekly) backups of your files.
  - Required software is pre-installed.
  - It is an exact replica of the environment in which our testing system, Marmoset, works. If your submission works in the undergraduate environment, it will work on Marmoset.
2.2 Connecting to the CS undergraduate environment

2.2.1 Linux

- If you primarily use Windows, you can still install Linux (e.g. ubuntu) on a second partition of your hard drive.
- Most Linux distributions come installed with typical applications that you will need (e.g. `vim`, `ssh`, `scp`).
- To log in to the undergraduate environment:
  - Open a terminal.
  - Use the Secure SHell command, `ssh`. Execute the command `ssh -Y userid@linux.student.cs.uwaterloo.ca` (replace `userid` with your university userid not student number e.g. `j2smith`).
  - Enter your CS environment password when prompted (you won’t see the characters and your cursor will not move as a security feature).
  - Note that the `-Y` option allows for X11 forwarding (e.g. graphical applications).

2.2.2 Mac

- Every Mac has a Terminal application that runs a text interface for Linux. Open the Terminal application and then use the `ssh` command discussed above.
- You should be prompted to enter your password. Enter your password that you just set up previously and press enter. You will not see the characters entered, and your cursor will not move as a security feature.

2.2.3 Windows

You only need to use one of the following options but you could try a few to see which one you like.

1. Using Command Prompt
   - Every Windows installation comes with the Command Prompt application. Open this application and then use the `ssh` command discussed above.
   - Note that this Command Prompt is not running bash. While a few bash commands might work, it is not a substitute for bash.

2. PuTTY (Recommended as it is the oldest and most stable way.)
   - This is an SSH client that can be downloaded for free here: https://putty.org/
   - Open PuTTY.
   - In the “Host Name” field enter `linux.student.cs.uwaterloo.ca` (press the Save button to save this session for later use).
   - In the sidebar under “SSH”, click “X11”.
   - Click the box that says “Enable X11 forwarding”.
   - Press the “Open”.
   - Enter your CS username and password.
   - Again, it may appear that nothing is happening when you type your password but your keystrokes are being hidden for privacy.
   - Here is a link on how to setup `ssh` for PuTTY: https://devops.ionos.com/tutorials/use-ssh-keys-with-putty-on-windows/

3. Linux bash shell on Windows 10.
   - The 64-bit install of Windows 10 now supports a Linux subsystem.
• You would need to enable the "Windows subsystem for Linux” and then install the Ubuntu app from the Microsoft Store.

Once you have connected to the server you will be in a Linux environment. This is where you should be testing all your assignments, as we will be testing them on these same servers. Now that you are on the student server, create a proper subdirectory for your work in this course. Then run the following command to create a copy of the course repository:

```
git clone ssh://linux.student.cs.uwaterloo.ca/u/cs247/pubrepo/1225/.git
```

Then you should have a directory named `1225` in your current working directory. In that directory you should find (currently) two directories, `tools` and `tutorials`. `tutorials` is the directory that stores code examples from tutorials, and `tools` contains the testing tools for this course.

In the future more tutorial materials will be added to the `tutorials` directory and future assignment files will be provided here as well. Whenever you want to get the most up to date files in this directory all you have to do is navigate to it as your current working directory, and run the command `git pull`. It is advised that you do this everytime you go to work on CS247!

**Submit:** In the `tutorials/tut0` directory you will find a file named `q0.txt`. Submit that exact text file as your answer to this question to Marmoset.

## 3 Question 1: Guess A Bad Number

In this question you will be testing and implementing a very simple C++ program. You are asked to guess a special(bad) number in the range of 1 to 10.

This program takes an input from `cin` and

- If the input is not an integer, the program prints: “Input is not an integer!”.
- If the input is not in the range of 1 to 10, the program prints: “Integer is not in the range of 1 to 10!”.
- If the input is 2, the program prints: “That is 2 bad.”.
- Else, the program prints: the value of the input followed by “ is a good number!”.

**Due Date 1 Task:**

Your task is not to write this program, but to design a test suite for this program. Your test suite must be such that a correct implementation of this program passes all of your tests, but a buggy implementation will fail at least one of your tests. Marmoset will use a correct implementation and several buggy implementations to evaluate your test suite in the manner just described.

Your test suite should take the form described in tutorial: each test should provide its input in the file `testname.in`, and its expected output in the file `testname.out`. The collection of all `testnames` should be contained in the file `suiteq1.txt`.

Zip up all of the files that make up your test suite into the file `tut0q1a.zip`, and submit to Marmoset.

**Due Date 2 Task:**

Implement the program above in C++. Call your program `guess.cc` and submit to Marmoset.