2 Software Tools

2.1 Preamble

The exercises below are intended to provide a means to test your understanding of the programming-related material covered in the course. It is highly recommended that you work through these exercises as you cover the corresponding topics in the video lectures. By doing this, you will greatly strengthen your understanding of the material in these video lectures, which will greatly reduce the amount of pain and suffering required to complete the programming assignments in the course. In exercises that require the building (i.e., compiling and linking) of code, the CMake tool should be used for this purpose. For additional information about these exercises, refer to Section 1 of this document.

2.2 Topics Covered

These exercises cover software tools, such as Git and CMake.

2.3 Exercises

1. **[git]**
   Clone the repository for the C++ lecture slides. (The URL for this repository can be found in Section 1.1.) The objective of this exercise is simply to confirm that you are able to successfully clone the repository for the C++ lecture slides. [Hint: You will need to use `git clone`.]
2. **[git]**
   In this exercise, you will be creating an exact copy of a remote Git repository and then making numerous changes to this new repository. To begin, select an empty directory for use in this exercise. This directory will be henceforth denoted `$TOP_DIR`. Then, perform the tasks listed below (in order).
   (a) In the directory `$TOP_DIR/hello.git`, create an exact copy of the Git repository with the following URL to serve as a "remote" repository in this exercise:
   `https://github.com/mdadams/cppbook_multilingual_hello.git`
   This can be accomplished by performing a bare clone operation as follows:
   ```
   cd $TOP_DIR
   git clone --bare \
     https://github.com/mdadams/cppbook_multilingual_hello.git hello.git
   ```
   Note that a backslash character (i.e., "\") that falls at the end of a line denotes line continuation.
   (b) Clone the "remote" repository `$TOP_DIR/hello.git` to the directory `$TOP_DIR/hello`. (Do not make a bare clone of the repository. If you do, you will have no working tree for making changes to the repository.) Observe that the files in the newly created `$TOP_DIR/hello` repository contain the source code for the classic hello-world program written in several programming languages, including C++, C, Fortan, and Java. [Hint: You will need to use `git clone`.]
   (c) Modify the new repository `$TOP_DIR/hello` as follows. First, remove the files `hello.f` and `hello.java` (since friends do not let friends program in Fortran, and Java is just plain evil). Edit the file `README` to remove any mention of Java and Fortran. Propagate your changes to the "remote" repository `$TOP_DIR/hello.git`. [Hint: You will need to use `git clone`, `git rm`, `git add`, `git commit`, and `git push`.]
   (d) Clone the "remote" repository `$TOP_DIR/hello.git` to the directory `$TOP_DIR/hello_2`. Confirm that your changes made in the previous step are all present in the newly cloned version of the repository. [Hint: You will need to use `git clone`.]
3. **[cmake]**
   Build (using CMake) and execute the `hello` program in the directory `cmake/examples/hello` (in the Git repository for the C++ lecture slides). A CMakeLists file is provided. You simply need to use it. (You will need to clone the repository in order to obtain a copy of the data in this directory.)
4. **[cmake]**
   In this exercise, you will need the (`$TOP_DIR/hello.git`) repository that was produced in the process of completing Exercise 2 (above). Due to the removal of the source-code files `hello.f` and `hello.java`, the `CMakeLists.txt` file

is no longer correct (as it references files that no longer exist). Fix the `CMakeLists.txt` file so that the C++ and C versions of the hello-world program can be built successfully with CMake.