CS 3053
Project – Build Test
Due Tuesday 2018.01.30 at 12:00pm.

Overview

In this assignment you will learn about the code organization and build process that we'll use for project application development. There are a lot of steps, but each one is meant to be relatively quick and easy. You must complete **all** individual tasks before proceeding to group tasks.

Individual Tasks

Individual tasks #4–6 must be completed entirely on your own.

#**1:** Introduce yourself to Gradle (https://gradle.org) and learn the basics of how to use it. Spend some time at https://docs.gradle.org/current/userguide/userguide.html to get a feel for what Gradle does and how it generally works. (You **won't** have to write your own Gradle script to compile and run your code from a command line; I've done that for you.)

#2: Download and install Java (if you don't already have it) on the system you plan to use to implement code. Information on Java can be found at http://www.oracle.com/technetwork/java/. Make sure that whatever version you're using can compile to Java 8 bytecode. Whatever Java code you write during the project will be limited to the Java 8 language features and API.

#3: Download and install Gradle on your system. Gradle is open source, so you can get it in the usual ways like downloading directly, compiling the source, using a package manager, etc. (Note that if you want to compile and run your code from inside an IDE like Eclipse, you'll have to figure that out for your particular environment. Consider having all team members use the same IDE, to make it easier to share your team's chosen code at each stage.)

#4: Go into the `ou-cs-hci` directory that came with this homework. The directory contains a `build.gradle` script file. Have a look at it; there's a summary of build commands at the top. (Also make sure to look at the section on "Alternative Start Scripts". You'll need to add and/or uncomment items to build your code in future project stages.)

#5: Compile the build. You can do this by typing `gradle installDist` on the command line. You need to be in the `ou-cs-hci` directory for this to work. Then go into the `build/install/template/bin` directory to run any of the resulting programs. Each app has a `.bat` version for running on Windows. The `base` program is very basic, but you can use the corresponding class (`Base.java`) as a starting point for your own coding. The source code files are deep in the `src` directory, as leaves in a typical Java package hierarchy. Your own code should go in the `stages` directory. Put any classes that you add into the `edu.ou.cs.hci.stages` package. You're free to add subpackages as your code gets more complex. I've copied `Base.java` to `BuildTest.java` (with corrected class and package names) to make it easier for you.

#6: To demonstrate that you've succeeded at all of the above tasks, modify the `BuildTest.java` class to display your name instead of "Build Test" in the middle of the window. Take a screenshot of your window, trim it, and put it in the `Results` subdirectory as `buildtest.jpg` or `buildtest.png`.

To turn in your individual work, first run `gradle clean` to reduce the size of your build. Second, zip your entire `project-build-test` directory. Submit your zip file to the "Individual - BuildTest" assignment in Canvas.

Group Tasks

For this assignment, individual tasks may be interleaved with group tasks.

#7: Discuss and adopt a way to share your individual builds with each other. This may be as simple as uploading your zip files to a mutually accessible store, such as you team's private area in Canvas. Feel free to take advantage of cloud storage, version control, or online project sites like GitHub. **All access to project materials must be restricted to your team. Do not allow access to anyone except your teammates, the TAs, and me.** You're free to distribute your code as you like after the course is over.

#8: Make your individual results available to each other using your adopted sharing method. Check with each other to make sure everyone can access and modify your team's collective materials.

#9: Document your individual and group decisions. Write this together. Briefly describe your individual development set-ups (what you did for tasks #1–3). Summarize your team's sharing strategy including which tools, sites, etc. you'll be using. A few paragraphs is sufficient. Use regular paragraphs and standard formatting (12 point font, 1 inch margins, etc.)

To turn in your group work, go to the "Group - BuildTest" assignment in Canvas to submit your results as a PDF. Only one team member needs to turn in the group component.