1 Overview

A five line summary of this document follows.

You submit a flat zip or tar archive file which after expansion does not create directories or hidden files and contains source code file(s) and a .txt file. Every file name (including the .txt and the archive's) must have the last four digits of your NJIT ID as part of the filename before the suffix/extension. Every document in the archive contains (in the form of comments for source code) your first name, last name and last four digits of your NJIT ID. Every class name should end with the last four digits of your NJIT ID.

1.1 NJIT ID: Where to find it

If you don't know your NJIT-ID, login to `my.njit.edu` and there you may locate your NJIT-ID (not your SSN) and extract its last 4 digits. More info in Document 1, Appendix F. For the sake of an example in the remainder, the Programming Project (PrP) will be denoted by `prp` and the last four digits of an NJIT ID by `1234`. In the remainder, we will be using the underscore symbol rather than the dash (minus) symbol, where needed. **The symbol _ is the underscore symbol, not the minus/dash - symbol.** (Java detests dashes.)

2 Canvas submissions and filenames used

RULE-0. Accidental submit and resubmit. The last of multiple PrP submissions will be graded. Do not send files by email, they will be discarded. Do not submit binary files or executables.

RULE-1. Canvas and Homeworks. You are to submit a HW through Canvas Quizzes. You may submit at most three times; the last submission will be graded.

RULE-2. Canvas and PrP. You are to upload and submit the PrP in a single flat .tar or .zip file through Canvas Assignments (source code files and a .txt file). Canvas will reject submissions greater than 5,000,000B.

RULE-3: File-name of a PrP submission.
(i) The archive filename would include the last four digits of your NJIT ID before the suffix. I would have used a file named `prp_1234.tar` or `prp1234.zip`. The same applies to individual source code files and object names e.g. `Object1_1234.java`, `code1234.c`. (ii) The first line of every file you submit in the archive must contain your first and last name and the last four digits of your NJIT ID. Use a comment statement for source code files. (iii) The single .txt file named for example `prp_1234.txt` within the archive must also contain a
"I HAVE READ THE COLLABORATION SECTION OF THE SYLLABUS. DOCUMENT 3 and PRP ADHERED."
in capital case as shown (no quotation marks needed). Furthermore, it may contain compilation and/or execution instructions, a bug report and other useful information. We do not read your submission code, we just test it. But WE DO read the .txt file!
(iv) You write code and YOU test it/debug it before submission on an OSL/AFS machine! Test it means: extract files from the archive, make sure no directories are created and files expand flat, compile your files, and then run or interpret your code on that machine. Check file names and the first line of every file in the archive to adhere to (i) and (ii) and (iii) above. Compiling on a GUI environment is different from using the grading environment: manipulating text files on Windows or MAC-OSX can cause newline, linefeed problems. In linux you may check for hidden directories with `ls -al`.

3 Testing and Grading

3.0 OSL or AFS machine access OFF-CAMPUS. To connect to an OSL or AFS machine you may need to use VPN (install and activate it). Then you may need to use a secure shell client. Read document "Connecting to *NIX at NJIT" (last link of Section B of the course web-page) on how these work for a Windows machine or visit ist.njit.edu or in google search for 'NJIT ssh' or 'NJIT VPN' (without the quotes). OSL and AFS machines are linux machines with host names osl2.njit.edu through osl31.njit.edu or afsconnect1.njit.edu or afsconnect2.njit.edu etc. Extracting files, compilation and execution on these machines is through the command line (eg unzip, tar, gcc, g++, javac and java).
3.1 Read requirements carefully. The PrP requires **command line processing** and **file-based I/O**. If you are not familiar with them figure this out early in the semester. Submissions that cannot handle command line processing or file-based input/output correctly risk of getting 0 points as ordinary testing will not work.
3.2 Testing and Debugging. Make sure your archive zip or tar file extracts its constituent files flat without creating directories/ subdirectories/folders hidden or not on the OSL or AFS machine of your choice, and you use those available tools to compile and run your code. Do a `gcc -v` or `g++ -v` or `javac -version` or `java -version` to confirm and report in the .txt file the version of compiler used. Versions available on afs are found with a `module avail gcc` and loaded with say a `module load gcc/9.1.0`. If compilation is to proceed in a certain file sequence add a note in the .txt file, as needed. DO NOT DO testing 'remotely' using 'software of the remote platform' to edit files on 'AFS'. If you do not know what a newline becomes in Linux/Unix, Windows, MAC-OSX, be prepared for nasty surprises.
3.3 File types to submit. C or C++ files with `.c` or `.h` or `.cc` or `.cpp` or Java files with `.java` are acceptable. A single .txt file must be included per RULE-3. The more info you have in it the less chances you have to get a 0. Do not include binary files (.jar, .class etc) or canvas might reject your submission. Do not send files by email.
3.4 Presence of directory structure is not allowed Beware of MAC OSX: it has a tendency of generating 0pt PrP submissions because it creates hidden directories. Do not skip step 3.2 above, and read the last sentence of the previous page!
3.5 Grading For a HW, grading is more or less straighforward. For PrP, the grader will first decide and create testing instance(s) and then grade your submission based on whether it passes successfully or not those testing instances using the specified interface (eg command-line processing). One of the testing instances or variation of it is described in the PrP itself and may also be found in a text file in Section B. If your code does not pass any of these testing instances, it will get 0 points, unless there is a detailed bug report that YOU HAVE PROVIDED in the .txt file, that the grader can read, comprehend and build some testing instances to partially test your code. If no information is provided by you, the grader will NOT read your code.
3.6 Grade and Canvas Grading. The PrP or HW grade will be made available in Canvas. Ignore canvas grade accumulations; canvas has no clue about the course grading scale or scheme. The only deadline is before noon (12-o'clock noon that canvas interprets and calls it 12PM is not midnight but noon time) of the day specified in the calendar of Document 1 (Syllabus).] ■