

1 Introduction

1a. Email. The PrP (Programming Project) specifies all email addresses needed. Use them and only them. Do not send an email to any other variation of an email address, or your submission will NOT BE ACCEPTED and you will get 0 points. YOU HAVE BEEN WARNED.

1b. Conventions. For the remainder we assume that the last four digits of your NJIT ID (NOT YOUR SOCIAL SECURITY NUMBER) are WXYZ. You will also need them in the in-class exams.

Checklist for submitting a Programming Project

- 0: Use your NJIT email; spam filters might cause problems. As we will do the processing of your work (extraction, compilation, execution, and testing) on `afconnect1.njit.edu` or `afconnect2.njit.edu` it is imperative that you test your submission on one of those machines and make sure everything is fine. Alternatively, you may use an osl machine (eg. `osl21.njit.edu`).
- 1: **Pack things up into an archive file.** You send us ONE file only that is an archive in one email. Make sure it DOES NOT include directories/folders or subdirectories hidden or not, or java `.class/.jar` files, or other executable/binary files. We WANT only SOURCE CODE (including C or C++ header files as needed) and a SINGLE `.txt` file. Make-sure when we dearchive your archive, IT DOES NOT CREATE ANY DIRECTORIES. You may use an archiving program such as `tar` or `zip` BUT NOTHING ELSE. The archive file that you submit must be named `prp_WXYZ.zip` or `prp_WXYZ.tar`. The first and last three characters are in lower case! The fourth character is an underscore, not a dash. The single `.txt` file of the archive should be named `prp_WXYZ.txt` and might contain compilation or other instructions, and also bug information. At a minimum it contains two words in it: **NO BUGS**.
- 2: **Compose email.** You need to send ONE email message to BOTH the instructor AND the assistant AND cc it to yourself as well. (Read the PrP for details.) This single email MUST CONTAIN BOTH OPTIONS if you plan to submit two options, in a SINGLE archive that is attached to the email. You are not to submit two emails, separately for each option, or multiple (archive) files. Read the first 10 lines of this document again and carefully. Prepare the email and attach the relevant information from Step 1: but make sure that the Subject line is structured properly per Step 3: below.
- 3: **Subject of email.** Make sure that your email has a **subject line** with three words `cs610 prp WXYZ` if the last 4 digits of your ID are WXYZ. Use spaces between the words. All characters in lower-case.
- 4: **Send email.** When you are done with the composition of your email, and have formulated a proper subject line, think twice before hitting the send button. If you CCed your email to one of your email accounts you may verify proper TRANSMISSION/RECEIPT. Some devices delay transmission. If you are ready, hit the Send Button and transmit the homework email.
We acknowledge emails promptly and during regular hours, rather quickly.

2 File/Function/Class Names

Every file or function or class name of yours must end with the last four digits of your ID. So if we specify in an option that you need to build a class in Java named say `henc` we in fact mean that your class should be `hencWXYZ` or `henc.WXYZ` stored in say `hencWXYZ.java` or `henc.WXYZ.java` respectively.

3 First line(s) of Every File you Submit

Every file name you submit must identify your authoring of the work submitted. Thus if i were you, I would have written something like

```
/* GERBESSIOTIS ALEX   cs610 PrP WXYZ   */  
or  
// GERBESSIOTIS ALEX   cs610 PrP WXYZ
```

4 Programming Project

4.1 Read requirements carefully. Follow the requirements of the Programming Project. There are two options each one worth the full programming project points. You may submit one option or both. You must submit them together in a single archive file in a single email. If you send two emails BEFORE the deadline the second or last would be taken into consideration and prior ones would be ignored. But if we receive an email after the deadline not only will that email be taken into consideration but also will it receive the appropriate penalty. Follow section 2 and 3 guidelines faithfully as well.

All options require **command line processing** and **file-based input and output**. If you are not familiar with dealing with them figure this out early in the semester and preferably by the midterm. Submissions that cannot handle command line processing or file-based input/output correctly risk of getting 0 points as ordinary testing will not work and the onus would be on you to provide detailed bug information.

4.2 Testing and debugging your code. Whether you develop/test code on a Windows, AFS Linux or Apple OS/X machine we will use one of `afsconnect1.njit.edu` or `afsconnect2.njit.edu` to do compilation, testing, and grading. These are Linux machines. It is thus strongly recommended to at least test your code/submission on any of these machines or an OSL machine (eg `osl21.njit.edu`). If our compilation or testing encounters problems on say `afsconnect1.njit.edu`, do not complain. You have been warned!

4.3 ANSI C/C++ or Java compliance. Given that AFS machines will be used for our testing and grading use standard ANSI C or C++ available there. Do a `gcc -v` or `g++ -v` to confirm the version or `javac -version` or `java -version`. Versions available on afs are found with a `module avail` and loaded with say a `module load gcc/4.9.2`. If compilation is to proceed in a certain file sequence add a note into the `prp_WXYZ.txt` file; include also a note of the compiler version used or the java launcher used. Do not use `.jar` files nor include `.class` files. The more information is available in the `.txt` file the less chances you have to get a 0.

4.4 Files types to submit. C or C++ files with `.c` or `.h` or `.cc` or `.cpp` or Java files with `.java` are acceptable. A single text file `pp_WXYZ.txt` must be included. Minimally it includes two words: NO BUGS. If it is missing, you may end up getting 0 points if your code has bugs.

5 Grading

No partial credit will be given to submitted code that does not satisfy the previous guidelines including compilation, testing platforms, names, and the existence/information of `prp_WXYZ.txt`. Moreover compliance to the specifications of PrP is expected including command-line processing and file-based input/output. If we tell you to implement something per class requirements a deviation from class requirements will get you a 0.

Default Testing Instances and Grading. The grader will first decide testing instance(s) and grade your submission based on whether it passes successfully or not these testing instances using the specified interface (eg command-line processing with file-based input/output). 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 `prp_WXYZ.txt`. If your code prompts for a file-name or deviates from the input output interface will also get you a 0.