```
$Id: submit-checklist.mm,v 1.18 2018-01-04 16:42:46-08 - - $
PWD: /afs/cats.ucsc.edu/courses/cmps104a-wm/Syllabus/submit-checklist
URL: http://www2.ucsc.edu/courses/cmps104a-wm/:/Syllabus/submit-checklist/
```

Computer Science is an engineering discipline and is very exacting. Do your work carefully and verify each step. Generally, 1/2 of the points are for submitted source, and the other 1/2 of the points for the results of testing. Even one wrong character in a source file or **`Makefile`** can cause disaster. The following applies to all projects and labs.

1. Before you begin.

(a) Print out all files provided in the assignment directory. This will include the assignment specifications and listings of source code provided.

(b) Start early and study the specifications and code provided early, during the first few days of the assignment cycle.

(c) Check the due date in the **`README`** at the root of the course directory. Excuses like: "I was under the impression that ..." or "I though that ..." in connection with the due date are just silly. Check on the due date to be sure. You are responsible for proper calendar management.

(d) The submit command will always be of the form:
`submit` *course-init.qtr asgx filename...*
Assignments used in previous quarters might not have been updated, or possibly a **`Makefile`** will have a submit directive from a previous quarter. Make sure the course code (e.g., **`cmps104a`**), the instructor's initials (e.g., **`wm`**), and the quarter (e.g., **`w18`**), and the assignment name (e.g., **`asg1`**) are all correct.

(e) If you are not sure which submits are open, type the command:
`submit` *course-init.qtr*
and you will see an error message listing the available submits. Obviously, if you are taking a different course, substitute the appropriate code for that course.

2. As you are working.

(a) Always build your software using **`gmake`** with a working **`Makefile`**. Do not type in the command **`javac`**, **`gcc`**, **`g++`**, or any other language name at the terminal.

(b) Every time you edit a file, check it into **`RCS`** or other archival system. This should happen automatically via the **`Makefile`** with **`gmake ci`**.

(c) Run **`checksource`** on all source files every time you edit them. This should happen automatically via the **`Makefile`** with **`gmake ci`**.

(d) Every time you edit some files, run a test suite against the jar or executable.

(e) Submit early. Submit frequently.

(f) Verify! **Verify!** ***Verify!*** Check it again! **Check it again!** ***Check it again!***

3. When you are finished.

(a) Is the directory `/afs/cats.ucsc.edu/courses/cmps012b-wm/bin/` in your `$PATH` variable, so that you can use the scripts in that directory?

(b) If you are doing pair programming, and you run the script `partnercheck` in the directory containing your `PARTNER` file, does it report information about the user named therein?

(c) Did you run `checksource` on all of the files you are about to submit, without complaint?

(d) Did you put a `submit` target in your `Makefile`, and does `gmake submit` successfully submit all necessary files?

(e) Did you type the command `gmake` just before submitting to verify that the `Makefile` and all of your source code is compilable? If the grader runs `gmake` and the build fails, you lose 1/2 of the points for the program, even for a trivial error.

(f) Did you copy the files from the `.score` directory and check on the insertuctions to the graders?

(g) Did the test scripts from that directory work successfully?

(h) Verify! **Verify!** ***Verify!*** Check it again! **Check it again!** ***Check it again!***

4. After you submit.

(a) The graders will only look at is what you submit before the due date.

(b) Did you verify the names of the files in the actual submit directory?

(c) Submit is a program that copies files into the submit hierarchy. For example, if you type
`submit` *course-init.qtr asgx filename...*
then your code will be present in the directory
`/afs/cats.ucsc.edu/class/`*course-init.qtr*`/`*asgx*`/$USER`
where `$USER` is your username. The class volume and assignment name will vary from course to course and from quarter to quarter, but always follows the same pattern.

(d) Use `ls` to verify what you have submitted. Make sure you understand how to use `submit` well in advance of the due date. Example command:
`ls -la /afs/cats.ucsc.edu/class/`*course-init.qtr*`/`*asgx*`/$USER`

(e) In the submit directory itself, the names of the files you submit are prefixed by a sequence number. When submit is locked, only the latest will be kept for the graders. For example, if you submit `README` three times, you will see `1_README`, `2_README`, and `3_README`. The file `3_README` is what the graders will see as `README`.