Course Policy

This document specifies the general course policy for ling570, ling571 and ling572. Note that the course policy for a particular course in a particular quarter could vary and in that case the difference will be highlighted in that course's course policy.

In this document, "we" refers to the instructor, the TA, and the grader; "you" refers to the students.

1. Communication outside the classroom

- We will use Announcement, Discussion board, and Inbox etc. in Canvas to communicate with you. If you don't check Canvas often, you should set the canvas notification properly (e.g., instant notice or daily notice for Canvas activities).
- If you need to contact us outside the classroom, please come to our office hours or email us via Canvas.

2. Choice of programming languages for your assignments

At the beginning of the quarter, we will provide a list of recommended programming languages, which are all available on Patas.

a) You should choose a language on the list.

b) If you strongly prefer a language that is not on the list,

   i. you need to get an approval from the instructor beforehand, and the approval is at the discretion of the instructor.

   ii. please be aware that, if your code does not run properly, it could be harder for us to give you partial credit if we are not familiar with the language.

   iii. if the programming language is not currently installed on patas, the installation (by our system administer) can take time, and no extension will be given due to the delay caused by the installation.

3. Assignment due date and late penalty

Assignment due date:

- Unless specified otherwise, all the assignments are due at 11pm on the same day (e.g., every Thursday).
- No submission will be accepted TWO days after the due date.

Late penalty:

- Late penalty is 1% (for the 1st hour), 10% (for the first 24 hours), and 20% (for the first 48 hours).
- The score with late penalty = the score before taking the penalty * (1 – late penalty). For instance, if your submission is 26 hours late, your score will be score-before-taking-penalty * 80%.

Extension:

- If you need an extension, you need to contact the instructor at least 24-hours before the due time.
- The approval of an extension is at the discretion of the instructor.
- An extension is at most TWO days.
- If an extension is given to a student, the student still needs to submit within TWO days of the original due time; in other words, the extension will only reduce late penalty, but will not change the time that the submission area is closed.
- If an extension is given to the whole class, the submission area will close after two days of the new due date.

4. Files to be submitted

Unless specified otherwise, you should submit two files for each assignment:

- The 1st one is called readme.txt or readme.pdf, which includes your message to the grader and answers to questions that do not require coding.
- The 2nd one is called hw.tar.gz, which a zipped tar file that includes all the shell scripts, source codes, binary code, and other files for the assignment.
- Suppose all your files for the assignment are stored under directory X on patas, you should run the following to generate hw.tar.gz:

```
cd  X
tar -zcvf hw.tar.gz *
```

For coding assignments, you must submit shell scripts:

- The name of your shell script and the command line for the script have to be exactly the same as specified in the assignment.
- The shell script should be very short, and simply call the code written in a programming language of your choice (see Section 1 for the choice of programming languages).
- For submission, you must submit the shell script, the source code in the other language, and the binary code for that language (if the language requires compilation).

- The binary code must be produced on patas. We will not compile the source code for you.

We will provide a script called check_hwX.sh (e.g., check_hw1.sh for hw1) that checks whether hw.tar.gz contains all the required files.

- Before submitting your tar file, you should run the following on patas. Here, $course-root-dir is the root dir of the course (e.g., it is dropbox/17-18/572 for ling572 in winter 2018), $hw.tar.gz is your hw.tar.gz file.

```
cd $course-root-dir/hwX/
check_hwX.sh   $hw.tar.gz
```

- For each assignment hwX, we will provide a file called submit-file-list, which lists all the files that should be included in hw.tar.gz.
  - For the shell script, the list includes only the name of shell script, not the name of source or binary code.
  - check_hwX.sh will open your shell script and check whether the source/binary code called by the shell script is included in the tar file. For instance, if your shell script foo.sh called foo.py, check_hwX.sh will check whether foo.py is included in the tar file. But if foo.py calls other modules (say foo-module.py), check_hwX.sh will NOT check whether foo-module.py is included in the tar file.
  - It is your responsibility to ensure that all the source/binary code called by the shell script (directly or indirectly) should be included in hw.tar.gz.

- Note that check_hwX.sh checks whether the tar file includes all the required files, but it will not check whether the files have the correct format or whether the content is correct.

5. Regrading for three common problems

If you lose points due to the following "easily fixable" problems, you can request regrading:

- Wrong submission or missing files:
  - To show that you completed the assignment before the due time, you need to show the grader the timestamp of the files on your local patas directory.
  - To avoid this problem, please remember to run check_hw.sh before submission.

- Code crashes:
  - Your code may be tested on new test files that you have not seen before.
  - Your code should be tested on patas and all the required output files should be produced by running your code on patas, not on Windows, Mac, or other OS.
  - If your code requires a specific version of interpreter or compiler, make sure that the correct version is specified in the shell script (e.g., for python and perl) and used to compile the code (e.g., for C++ and Java).

  - If your code crashes, we will NOT debug the code for you. We will try to give you partial credit if your code is easy to follow and/or your readme file explains under what conditions your code might crash.
  - If you can easily fix your code (e.g., your code uses a different version of python interpreter), you can request a regrading and tell us what changes should be made to avoid the crash.

- Once again, the output files required by the assignments must be produced by running the code on patas, not on Windows or Mac. If you did not use patas to get the output and the output format is wrong (e.g., it includes Windows linebreaks), you can lose points.

For the whole course, you can request regrading for at most two assignments:

- There will be a penalty of 10% for the part that requires regrading. For instance, suppose your original grade is 70 points, and you require regrading for question Q2 for which you currently get 10 points. After regrading, your new score for Q2 is 30 points; now your new grade for the assignment will be (70-10) + 30*(1-10%) = 87 points.
- The regrading request must be made within 7 days after the grade is available on Canvas.

6. Other grading questions

If you have questions about your grade that are not related to the cases mentioned in Section 4, you should contact us within 7 days after the grade is available to resolve the issue. Such discussion will not count toward the two regrading requests and are not subject to the 10% penalty.

Also, in general, we will not provide sample code for assignments.

7. Discussion board

The discussion board is mainly used for students to discuss any course-related issue:

- We will check the Discussion board, but will not reply to any specific posts on the board.
- If we need to clarify certain confusion or address common questions, we will start a new discussion thread and pin the thread. That way, if you prefer, you only need to check pinned discussion threads, and can skip other discussion threads.
- We won’t be able to answer all the questions on the discussion board. If your questions on the board are not answered by your peers or us on the board, please ask them in class or during office hours.
- We won’t be able to debug code for you. For debugging questions, it would be better if you work with your peers to help each other.

8. Class participation

A student who actively participates in class and Canvas discussion may receive up to 2% bonus added to the final grade.

9. Rubric for assignments

For any coding assignment, the rubric consists of two parts: the standard portion (25 points) and the assignment-specific portion (normally 75 points). The standard portion is explained in the following table:

2 pts	hw.tar.gz submitted
2 pts	readme.txt or readme.pdf submitted
6 pts	All files and folders are present in expected locations
10 pts	Programs run to completion
5 pts	The output of programs on patas match submitted output

10. Grade Scale

Unless specified otherwise, we use the following table to convert grade percentage to the final grade:

Grade percentage	Final grade
98.0 -- 100%	4.0
96.0 -- 97.9%	3.9
94.0 -- 95.9%	3.8
92.0 -- 93.9%	3.7
90.0 -- 91.9%	3.6
88.0 -- 89.9%	3.5
86.0 -- 87.9%	3.4
84.0 -- 85.9%	3.3
82.0 -- 83.9%	3.2
80.0 -- 81.9%	3.1
78.0 -- 79.9%	3.0

76.0 -- 77.9%	2.9
74.0 -- 75.9%	2.8
72.0 -- 73.9%	2.7
……	

11. Incomplete

Per UW policy (at https://registrar.washington.edu/enrollment-and-records/incomplete-grades/), incomplete grades may only be awarded if you are doing satisfactory work up until the last two weeks of the quarter.

An "incomplete" grade is given only under extremely unusual circumstances (e.g., health issues, family emergency). You should contact the instructor ASAP and work out a plan for completing the course in a timely fashion.

12. Collaboration and Academic honesty

We encourage you to collaborate in various forms such as participating in Canvas discussion boards, forming study groups, helping your peers with coding problems. Nevertheless, you are expected to complete assignments by yourself (or with your teammates for team projects). Any kind of plagiarism is prohibited. For instance, it would not be permissible for one student to create a computer file containing the answers and then for other students to copy that file and submit it as their own work. Copying code from the web or from classmates is plagiarism. For more information, see the UW policy at

http://depts.washington.edu/grading/pdf/AcademicResponsibility.pdf

If you are not sure whether certain kind of collaboration is allowed, you should contact us first.