WO1 Clayton E. Williams
C II: potter
2023.06.09

Setup

Due to inability to test main, and poor design at the start, the first step was to refactor and move functions from `potter.c` into their own respective source files.

Suite Runner

The suite runner that will be used for this project is check. Check can be cloned from https://github.com/libcheck/check.git and includes documentation for setup as well configurations to include with `make`.

Test Suites

The test suites for this project will cover each of the following source code files:

- `struct_helper.c`
- `io_helper.c`
- `clean.c`
- `grade_and_print.c`
- `find_stu_index.c`

Test Cases/Unit Tests

The test cases for potter will focus around opening/reading a file, confirming user input, and verifying creation and resizing the arrays that hold the student names and times.

- `struct_helper.c`
  - `create_struct()`: assertions will be made that the size of `stu_list` = 16, `capacity` = 1, `size` = 0 and that it fails to malloc if `capacity` = 0.
  - `create_student`: test various good and bad inputs for student info and assert they pass/fail respectively.
  - `reallocate_stu_list`: assert that the temporary student structure is saved back into existing structure and the size equals new size based on iteration.
- `io_helper.c`
  - `get_num_entries`: assert returns 6; assert fails with invalid file name.
  - `get_user_input`: assert `reallocate_stu_list` is called when size = capacity; assert `create_student` is called; assert `stu_list` returned.
- `clean.c`
  - `clean`: assert `stu_list` is NULL

  `grade_and_print.c`

- - `grade_and_print`: redirect stdout to .txt file; assert diff stdout.txt and Pottery.txt = 0.
- `find_stu_index.c`
    - `find_stu_index`: similar to grade and print. assert the output when passed "Seth Rogen" with `-s` flag equals Pottery.txt. Do the math for all 6 entries and write those outputs to a text file and assert they equal those names passed as arguments. assert invalid input prints error message and exits.