

CISC 458

Tutorial 3

Finishing Phase I

Revised by Ahmed Harby from Previous Versions

Thursday, February 2, 2023

What to Hand In

What to Hand In

- The marking scheme can be found on OnQ under “Project Requirements”.
- Your (group) submission must include:
 - Documentation
 - Source Code
 - Test Suite
 - A README File

Documentation

- You must hand-in a complete documentation for the modifications you have made to the scanner/screener.
- Your documentation must include:
 - The location of changes
 - Where did a change occur? Where was something added?
 - Explanations/Details of the changes
 - How did you change it? How was it implemented?
 - Justification for the changes
 - Why did you change it? How does the change make it more like a **Quby** compiler?
- Documentation for the Test Suite is also necessary:
 - What does each test file do?
 - Expected output or error? (success or fail)

Documentation Options

- **Preferred:** In separate text files (one for general changes, one for test suites)
- **Optional:** For Test Suite files:
 - In filename (e.g. `old_comment__fail`)
 - As a comment in file (not when testing commenting!)

Documentation: Change Example

- **Acronym:** WHEW (Where? How? Explain Why?)

Where? In `stdIdentifiers`, in the keywords section.

How? Removed keywords `until ...`

- Include code snippet or describe it
- What else needed to be changed as a result?
(update `parser.pt`)

Why? `until ...` no longer recognized as keywords in Quby

- A complete copy of the modified source code must be submitted (the `ptsrc` directory).
- Make sure to run `make clean` before submitting.

- A set of sufficient tests to confirm that your scanner/screen works as required.
- **Suggestion:** Unit Testing
 - Small, isolated, specific tests
 - Test only one thing per file
 - Design cases that also cover error cases
 - Do **MANY** to ensure coverage
- Provide an argument for the completeness for your test suite in your documentation, as well as any other important information.

- Each test case should include:
 - Documentation: Purpose of the test case
 - See previous slides for more details.
 - The input program to the **Quby** compiler
 - `ssltrace` output from the compiler (`-e`)
- You may want to write a script to automate this process.

- A small file containing the following information:
 - Your group account (e.g. `cisc458x`)
 - The name, ID#, and NetID of each student in the group.
- If your directory structure and file names are not very obvious, explain them here.

What to Hand In

- Hand in **electronic copies** only.
- We can handle most common formats, but if you are using something exotic, let us know ASAP, as you may have to reformat.
 - **Suggestion:** Use files that can be opened on Linux running on the Queen's computers.
- Short and concise is preferable to long essays . . .
- We don't mark grammar or spelling, but please make it understandable and clear.

When to Hand it In

- **NO LATE SUBMISSIONS** without a convincing and documented explanation.
- Explanations should be sent to Dr. Karim, preferably before the deadline.

How to Hand it In

Suggested Directory Structure

mysub

- ptsrc/ (complete source code)
- README.{txt,pdf,...}
- documentation.{txt,pdf,...}
- testSuite/
 - testName.pt (the **Quby** test source file)
 - testName.eOutput (ssltrace output using -e flag)
 - testName.txt (documentation for that test case)
 - Purpose of the test case
 - Result: expected vs actual output
 - In most cases, can be merged with testName.pt using clear filenames and comments.
- **Optional:** autorun (optional script to run the test cases)
- **Note:** Using the existing ptsrc/test/ directory instead of testSuite/ is fine, but you must indicate which are the test cases!

Directory Structure Tips

Tip: Instead of copying `ptsrc` to the submission directory `mysub`, create a symbolic link:

- `../ptsrc` should be path to the `ptsrc` directory.

```
cd mysub  
ln -s ../ptsrc
```


How to Submit

- To submit, you must use the `turnin` command:

```
turnin -cisc458 -v mysub
```

- `-v`: flag for verbose output - show all the files you have submitted to ensure they are all there
 - `mysub`: is the folder containing all your files, as described in the previous slide
- There is no need to compress your files; `turnin` will take care of this for you.
- `turnin` also works with symbolic links (no need for hard copy of file to be in `mysub` directory).

Expected Output of turnin

```
Submitted files:
phase1-cisc458x/
phase1-cisc458x/mysub/
phase1-cisc458x/mysub/ptsrc/parser/
phase1-cisc458x/mysub/ptsrc/parser/Makefile
phase1-cisc458x/mysub/ptsrc/parser/parser.pt
phase1-cisc458x/mysub/ptsrc/parser/parser.ssl
phase1-cisc458x/mysub/ptsrc/parser/prints.pt
phase1-cisc458x/mysub/ptsrc/parser/scan.ssl
phase1-cisc458x/mysub/ptsrc/parser/stdIdentifiers
phase1-cisc458x/mysub/ptsrc/parser/scan.def
phase1-cisc458x/mysub/ptsrc/parser/parser.def
...
<Additional source files, docs, and test suite>
...
```

Note: cisc458x will be replaced by your group account.

- Document, Document, Document!
- Make sure your Test Suite is complete and **documented**!
- Make sure your compiler actually compiles using `make scanner`!
 - We will be using your scanners to run on our own tests.
- If you face problems with `turnin`, let us know immediately!

- Introduction to Phase 2