

# CSC 453 All Milestones

[Revision 0: January 25, 2023]

## Due Dates:

All due dates are on the class schedule.

## Notes

The Tau language specification is a separate document. It may not be complete. If you have questions, please ask on Piazza. (Incompleteness is intentional—you need to think critically about the language.)

Feel free to publically discuss the language specification on Piazza. If you have questions, please ask.

## Template Files

The template files for all milestones are in `tau/stubs/`. You will need to copy these files to your working directory and edit them.

For every milestone, you will turn in all the files:

1. `tau.ebnf`
2. `scanner.py`
3. `parse.py`
4. `bindings.py`
5. `typecheck.py`
6. `offsets.py`
7. `assign.py`
8. `codegen.py`

For early milestones, you will simply turn in the provided stubs for most of those files.

## Standard Requirements

All milestones must meet the following requirements:

1. Use Python 3.10 as the implementation language
2. Use `black` to format your code.
3. Use `pyright` to check your code for type errors. Your code must be error- and warning-free.
4. Use `mypy` to check your code for type errors. Your code must be error- and warning-free.
5. Turn the program in via Gradescope.
6. The program must meet the specifications of the given milestone and all previous milestones.

## Turning in the program

Turn in your program via Gradescope.

## Testing

I am providing a test program and test input. More input will come as we get closer to the due date.

The tests for each milestone has a test “pickle” is a name like in `tau/milestones/m2-scanner-simple.pickle`.

```
$ python3 -m testerator.main run tau/milestones/m2-scanner-simple.pickle
```

NOTE: The you can add `--verbose` and `--crash` flags to the command line to get more information about what is happening. E.g.,

```
$ python3 -m testerator.main run --verbose --crash tau/milestones/m2-scanner-simple.pickle
```

The pickles are found in tau/milestones/.

## **Grading**

Milestones are graded on a 0% or 100% scale. If your submission passes at least 80% of the test cases, you will get full credit for this milestone. Otherwise, not credit is award.

Failing any of the formatting and typing requirements will result in a 0% grade for the milestone.

## **Comments**

Comments in computer programs are meant to improve the readability of the program by somebody in the future. Write comments accordingly.

- Too many, or too long, comments make the code harder to read.
- Too few, or too short, comments make the code harder to read.

Points will only be lost when comments are horrible.

## **Programming Techniques**

The hallmarks of good programming are simplicity, clarity and ease of understanding. Programs in CSC 453 should be written to be understood by other people. Clever techniques that obscure what the code is doing are strongly discouraged.

Good object-oriented programming is encouraged. Bad OO programming is strongly discouraged. Overuse of inheritance is a very clear sign of bad OO programming.

Programs may lose points for lack of clarity, simplicity, or ease of understanding.