

Outline for October 10, 2023

Reading: §5.4, 6.11

Assignments: Homework 1, due October 12, 2023

Handout: “Printing”

1. `continue` and `break` statements in loops [*loop1.py*]
2. `import` statement
 - (a) `import math` [*hypotnoex.py*]
 - (b) Need the “math.” before “sqrt”
 - (c) `from math import sqrt` [*hypotnoex1.py*]
 - (d) Do not need the “math.” before “sqrt”
 - (e) Now add in exception handling [*hypotex.py*]
3. Full version of the hypotenuse program [*pythag1.py*]
4. Exception `ValueError` — built-in function or operation applied to operator with illegal value
5. Functions [*hello.py*]
 - (a) What functions are
 - (b) Defining them
 - (c) Using them
6. Quick look at using them [*quad.py*]
 - (a) Passing values to functions
 - (b) Returning values from functions
7. In more detail: passing values to functions [*args.py*]
 - (a) Formal parameters in subject definition
 - (b) Actual parameters (arguments)
 - (c) Matching arguments to formal parameters
 - (d) Local variables
8. In more detail: how Python does function calls [*quad.py*]
 - (a) Caller suspends execution at point of call, remembers where it left off
 - (b) Formal parameters assigned values from actual parameters
 - (c) Execute function body
 - (d) Return control to where caller left off
9. Refactoring code
 - (a) Compute the perimeter of a triangle [*peri0.py*]
 - (b) Collapse similar statements: make the distance between 2 points a function [*peri1.py*]
 - (c) Collapse similar statements: make the prompts a function [*peri2.py*]
 - (d) Refactor for clarity only: make the perimeter computation a function [*peri3.py*]
 - (e) Add error checking: “peri0.py” done right [*peri-c.py*]
10. Add error checking: “quad.py” done right [*quad-c.py*]