ECOR 1041
Computation and Programming

Introduction to Python:

Types and Expressions

Carleton University

References and Tools

- *Practical Programming*, 3rd ed., Chapter 2, pp. 7 – 15
  - This is the introduction and first three sections of Chapter 2:
    - How Does a Computer Run a Python Program?
    - Expressions and Values: Arithmetic in Python
    - What Is a Type?
- We encourage you use Python to try the examples as you read the book
  - You will need to install Python and an Integrated Development Environment (IDE) on your computer

Why Python?

- It is fairly easy to learn
- It is quite powerful
- It is popular: https://spectrum.ieee.org/the-top-programming-languages-2023

References and Tools

- Python 3.12 is recommended. Python 3.11 and 3.10 are fine. Do not use Python 3.9 or older releases (installers and bugfixes are no longer provided)
  - Download installer from https://www.python.org/downloads/
- Wing 101 Release 10 is the recommended IDE
  - Download installer from http://wingware.com/downloads/wing-101

References and Tools

- It is permitted to use another IDE (e.g., Wing Personal, PyCharm, IDLE), but instructors and TAs cannot provide support for those tools

- Lab work will be graded using a test harness in the Gradescope tool
  - Ensure that your code passes all tests (you get immediate feedback upon submission – ask the TAs for help if you do not understand the feedback) before you consider you submission as final

Lecture Objectives

- Learn how to use the Python shell as an interactive calculator, to investigate Python's integer and floating point numeric *data types*

- Start learning how to build software experiments to help you learn

Learning Outcomes (Vocabulary)

- Know the meaning of these words, in the context of computation
  - Type (data type)
  - Literal value
  - Expression
  - Operator
  - Operand

Learning Outcomes (Lecture 1 and Labs 1 and 2)

- Identify literal values of types `int` and `float`
- Identify the symbols for these arithmetic operators:
  - Negation, addition, subtraction, multiplication, division, integer division
  - Remainder (modulus)
  - Exponentiation
- Know the precedence of these operators

Learning Outcomes (Lecture 1 and Labs 1 and 2)

- Evaluate arithmetic expressions consisting of literal values and operators, using the same evaluation rules as Python; in other words, predict the values Python calculates when it evaluates expressions
  - You will be asked to do this on exams
- Know how to use the Python interpreter to verify your predictions

Expressions

- The reading for this lecture introduced:
  - Python's `int` (integer) and `float` (floating point) types
  - Composing and evaluating arithmetic expressions
- The examples are not exhaustive (too many possibilities)
- Starting with information from a textbook or a reference manual, how can we learn more?

Software Experiments

- “Learning is a series of experiments in which negative results are just as important as positive results… the idea is to foster an environment in which questioning is the modus operandi. We learn because we doubt or mistrust the seemingly obvious.”
- “The key to successful learning is directed experimentation; i.e., experimentation for the sake of finding out something that wasn’t known before.”
  - Wilf Lalonde, *Discovering Smalltalk*

Expressions

- An expression describes a computation and evaluates to a value

- One of the reasons Python is a great teaching language is that it is easy to build experiments to learn about expressions

Expressions

- Python's interactive interpreter (shell) *reads* expressions typed in the shell, *evaluates* them, and *prints* (displays) the resulting values

```
>>> 1 + 2 + 3 + 4 * ((5 // 6) + 7 * 8 * 9) + 1
2023
```

Experiments: Learning About Expressions

- We will now use the shell to help us learn about expressions that perform computation using integers and floating point numbers
  - An annotated transcript is posted on Brightspace

- You will continue with this in Labs 1 and 2

Summary

- A type defines a set of values and the operations that can be applied to those values
- In Python, integers have type `int` and floating-point numbers (a subset of the real numbers) have type `float`
- Python evaluates arithmetic expressions by applying operators to operands
- Operators have precedence, which affect the order in which subexpressions are evaluated

Recap of Learning Outcomes

Carleton University

Learning Outcomes (Vocabulary)

- Know the meaning of these words, in the context of computation
  - Type (data type)
  - Literal value
  - Expression
  - Operator
  - Operand

Learning Outcomes (Lecture 1 and Labs 1 and 2)

- Identify literal values of types `int` and `float`
- Identify the symbols for these arithmetic operators:
  - Negation, addition, subtraction, multiplication, division, integer division
  - Remainder (modulus)
  - Exponentiation
- Know the precedence of these operators

Learning Outcomes (Lecture 1 and Labs 1 and 2)

- Evaluate arithmetic expressions consisting of literal values and operators, using the same evaluation rules as Python; in other words, predict the values Python calculates when it evaluates expressions
  - You will be asked to do this on exams

- Know how to use the Python interpreter to verify your predictions