Introduction to Computer Programming and Computational Thinking

February 29, 2022

Session 3

Innovation 129

Logistics

Instructor and Mentor

- ✘ Mr. Leang (instructor)
- ✘ Jesse

Schedule

- ✘ 10:00 AM - 11:10 AM (AM Session)
- ✘ 11:15 AM - 12:15 PM (Lunch)
- ✘ 12:20 PM - 1:50 PM (PM Session)
- ✘ 1:50 PM - 2:00 PM (Departure)

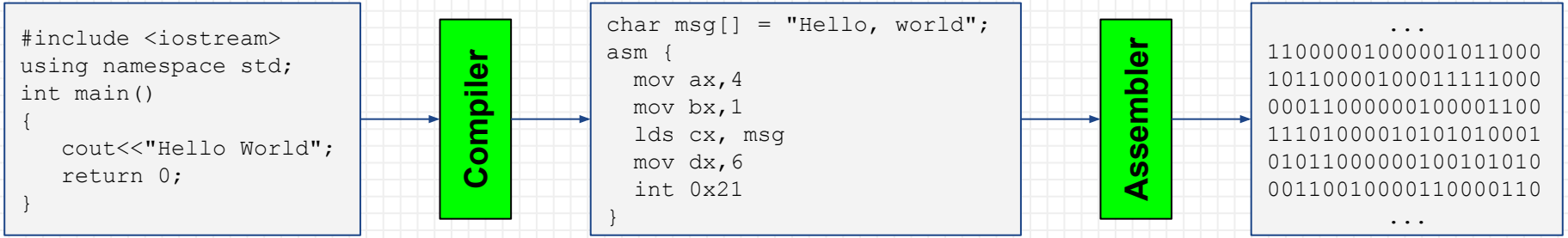
Recap of Session 1

Hello World

- ✘ Install ATOM editor
- ✘ Install Python
  - ✗ **Add Python 3.8 to PATH** during the installation
- ✘ Open up a blank document in the ATOM editor
- ✘ Type: `print("Hello World!")`
- ✘ Save the file as **hello.py** into your **Documents** folder
- ✘ Open up the command prompt
- ✘ Execute your script:
  - ✗ Switch into the directory containing hello.py
    - `cd Documents`
  - ✗ Call the python program to interpret and execute your code
    - `python hello.py`

Recap of Session 1

Programming for a Computer



High level language
(e.g., C, C++, Java)

Assembly language
(e.g., x86)

Machine code

Note: Python uses an interpreter to convert Python code to Python bytecode.

Recap of Session 1

Programming Environment

- ✘ A text editor
  - ✗ ATOM https://atom.io
  - ✗ Sublime https://www.sublimetext.com
- ✘ A Python interpreter
  - ✗ Python 3.8.0 https://www.python.org

Recap of Session 2

Python Data Types

- ✘ Integer: `1`
- ✘ Floating-point: `1.0`
- ✘ String: `'Early Identification Program'`
- ✘ Math operations: `+, -, *, /, //, %, **`
- ✘ Operator precedence (PEDMAS)
  - ✗ `5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 =`
- ✘ Python `type()` function
- ✘ String methods: `.capitalize()`, `.upper()`, `.lower()`, `.swapcase()`, `.count(`*`substring`*`)`
- ✘ Type casting (conversion)
  - ✗ `int()`, `float()`, `str()`

Recap of Session 2

Math Operators in Python

Operation	Result
`x + y`	Sum of `x` and `y`
`x - y`	Difference of `x` and `y`
`-x`	Change the sign of x
`x * y`	Product of `x` and `y`
`x / y`	Quotient of `x` and `y`
`x // y`	Quotient from floor division of `x` and `y`
`x % y`	Remainder (modulus) of `x / y`
`x ** y`	`x` to the `y` power

Recap of Session 2

Math Operators in Python

Let x = 5 and y = 2	Result
`print(x + y)`	7
`print(x - y)`	3
`print(-x)`	-5
`print(x * y)`	10
`print(x / y)`	2
`print(x // y)`	2
`print(x % y)`	2
`print(x ** y)`	25

Recap of Session 2

Operator Precedence

In Python, operators will be evaluated in order of precedence.

Order of operation - **PEDMAS**

1. **P**arentheses `( )`
2. **E**xponent `**`
3. **M**ultiplication `*`
4. **D**ivision `/ // %`
5. **A**ddition `+`
6. **S**ubtraction `-`

After **PEDMAS**, order goes left to right.
Use parentheses to override order.

Recap of Session 2

Operator Precedence

```
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
5 + (4 - 2) * 2 + 4 % 2 - 4 // 3 - (5 - 3) / 1 = ?
5 + 2 * 2 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 2 * 2 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 4 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 4 + 4 % 2 - 4 // 3 - 2 / 1 = ?
5 + 4 + 0 - 1 - 2 = ?
5 + 4 + 0 - 1 - 2 = ?
9 + 0 - 1 - 2 = ?
9 - 1 - 2 = ?
8 - 2 = ?
6
```

Recap of Session 2

String

In Python, a string is a sequence of characters.

```
print("Early Identification Program")
print('Early Identification Program')
print("""Early Identification Program""")
print("""Early
Identification
Program""")
```

Recap of Session 2

Methods for String Manipulation

Method	Result
String.capitalize()	Converts first character to capital letter
String.upper()	Returns uppercased string
String.lower()	Returns lowercased string
String.swapcase()	Returns string with casing swapped
String.count(*substring*)	Returns occurrences of substring within string

Recap of Session 2

Type Casting (Type Conversion)

```
print(float(1))
print(int(1.0))
print(str(1.0))
print(str(1)+1)
print(str(1)+'1')
print(int(1.0)+'1')
print(int(1.0)+int('1'))
```

Basic Data Types in Python

- ✘ Integer
- ✘ Floating-point
- ✘ String
- ✘ **Boolean**
- ✘ **List**
- ✘ **Tuple**
- ✘ **Dictionary**

Boolean

A data type with only two possible values (`True`, `False`) used for logic.

Typecast using `bool()`

Boolean in Python

```
print(True)
print(False)
print(bool(0))
print(bool(1))
print(bool(1123.23))
print(bool(-500))
print(bool('a'))
```

Comparison Operators in Python

Operation	Description
`x == y`	**True** if the value of x is **equal** to the value of y
`x != y`	**True** if the value of x is **not equal** to the value of y
`x <> y`	**True** if the value of x is **not equal** to the value of y
`x > y`	**True** if the value of x is **greater than** the value of y
`x < y`	**True** if the value of x is **less than** the value of y
`x >= y`	**True** if the value of x is **greater than or equal to** the value of y
`x <= y`	**True** if the value of x is **less than or equal to** the value of y

List

A collection of items in a particular order. Items are contained within **square brackets** with items separated with a comma.

```
my_empty_list=[]
my_grocery_list=['milk','bread']
my_number_list=[1.6180339887,3.1415926535]
my_list=['milk',2.0,300]
```

List Methods

Method	Description
`append()`	Adds an element at the end of the list
`clear()`	Removes all the elements from the list
`count()`	Returns the number of elements with the specified value
`extend()`	Add the elements of a list, to the end of the current list
`index()`	Returns the index of the first element with the specified value
`insert()`	Adds an element at the specified position
`remove()`	Removes the first item with the specified value
`reverse()`	Reverses the order of the list
`sort()`	Sorts the list