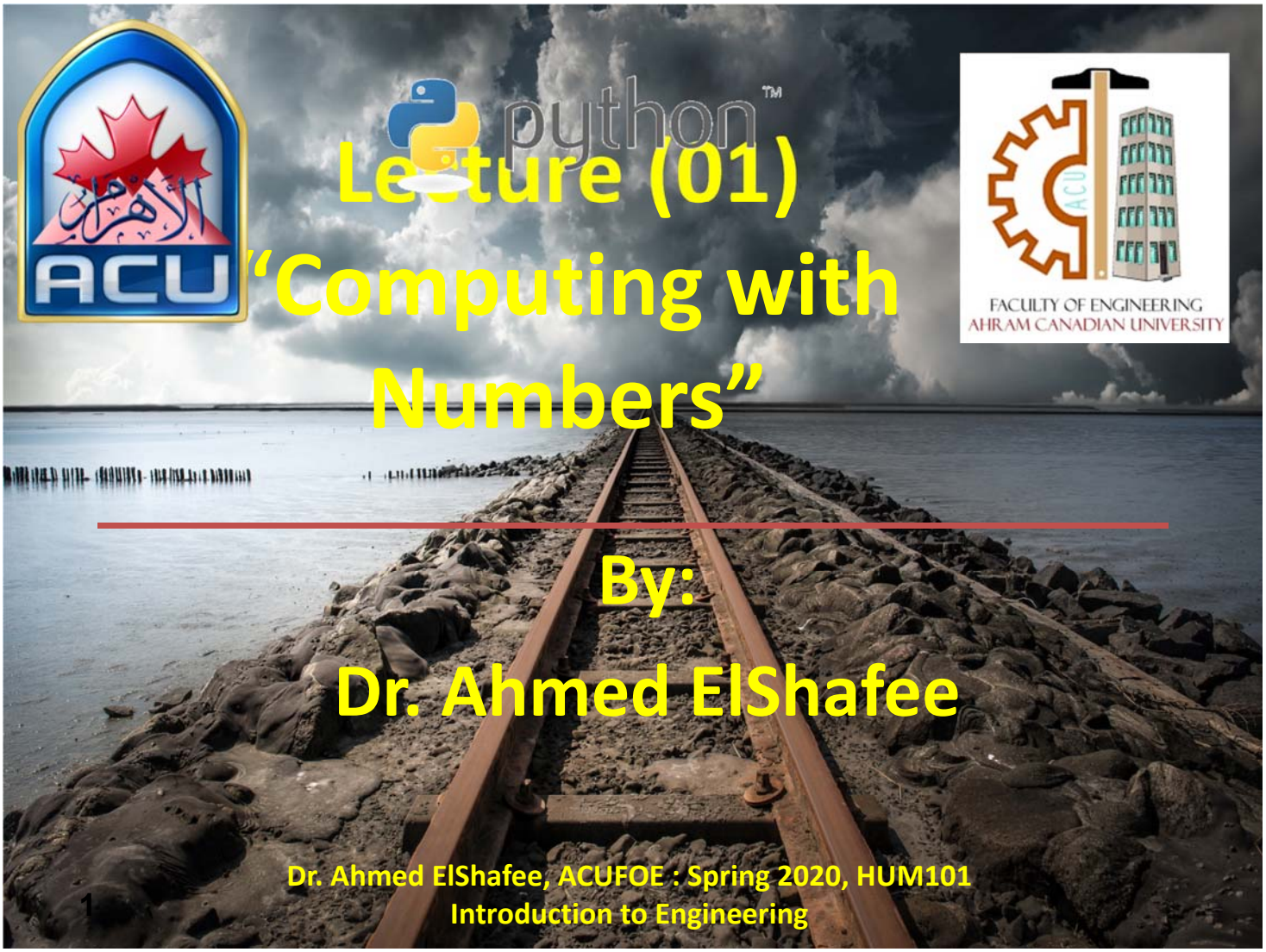
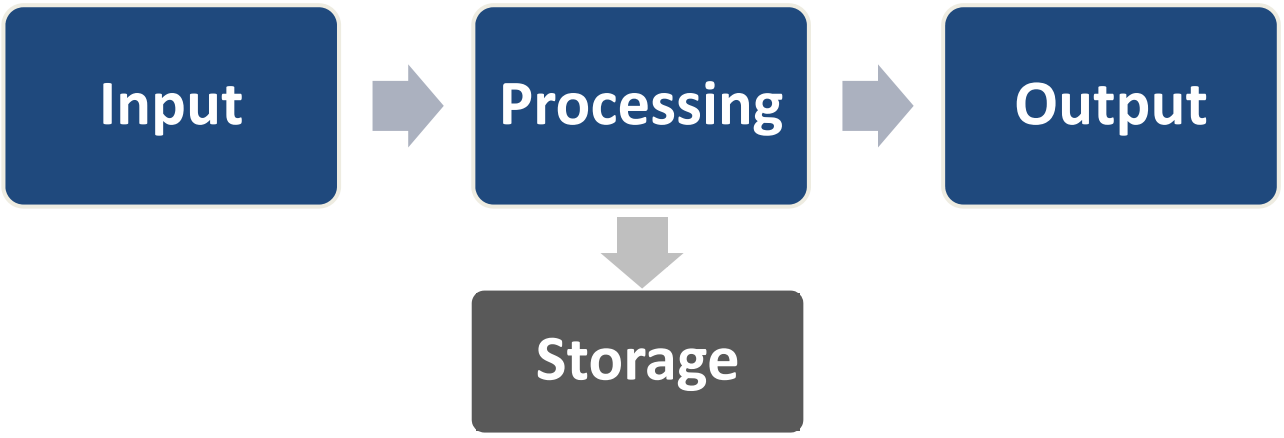



# Outline

- What is a Computer?
- What is a Computer Program?
- Inside the Editor (Wing IDE)
- Arithmetic Operations (Integers and Floats)
- Assigning Variables
- Getting Input
- Producing Output
- The Math Library

# What is a Computer?

- Multipurpose device
- Accepts input
- Processes data
- Stores data
- Produces output



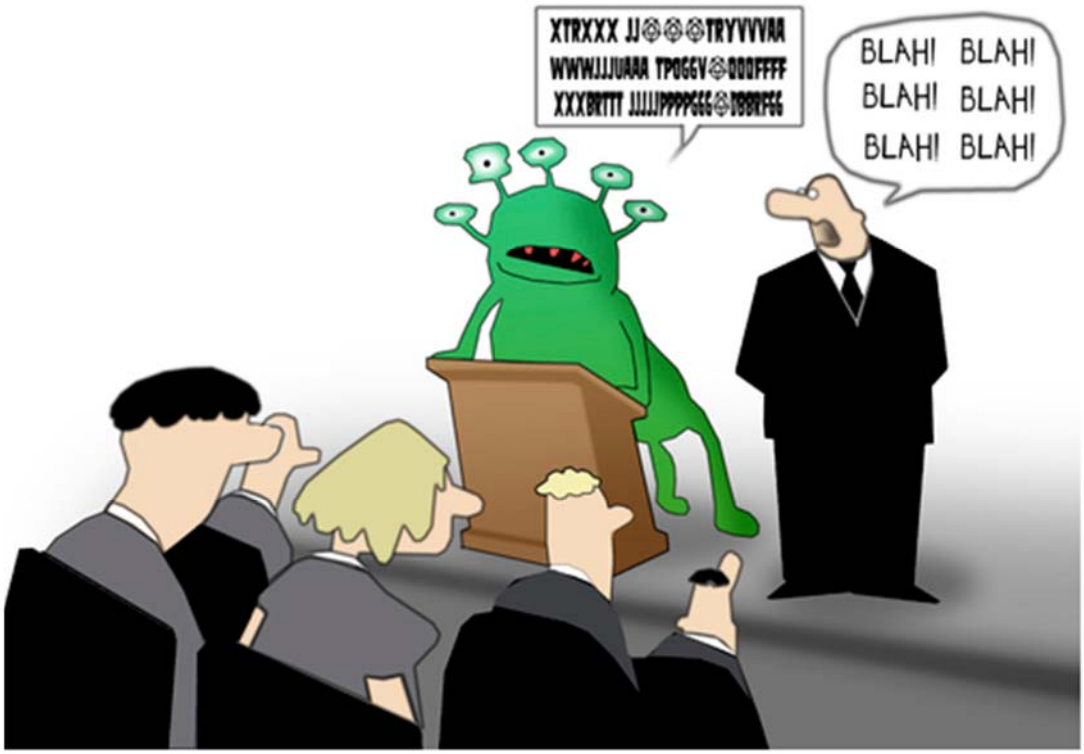

# What is a Computer Program?

- A detailed, step-by-step set of **instructions** telling a computer exactly what to do
- **Programs** are written by **Programmers** using **Programming Languages**
  - Example: Python
- Programming Languages such as Python are called **high-level** computer languages (human-readable)
- They need to be translated to a language the computer can understand (**low-level**)

# Translation Process

- There are 2 ways to translate from a High-level language to a Low-level one
  - Using an interpreter
  - Using a compiler

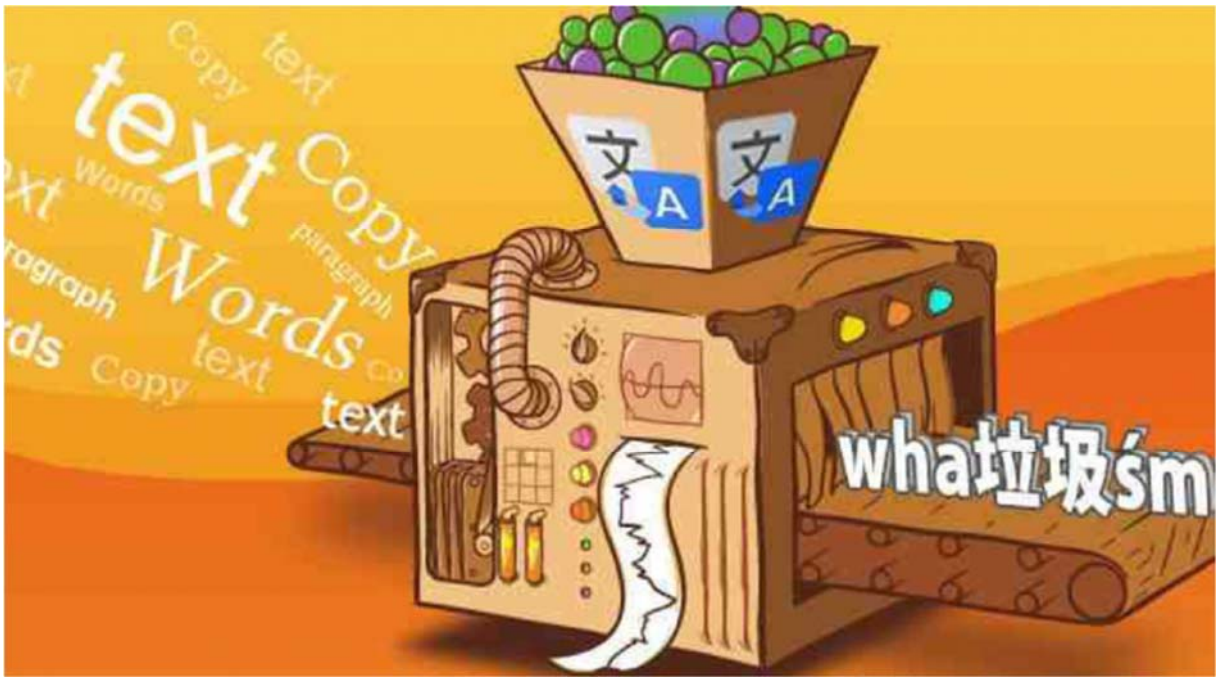
# Interpreter



# Interpreters

- Just like real life language translators, you can think of an interpreter as a program that takes your code and translates it to the processor can understand **line by line.**
- Each time you need to translate a statement, you need to pass it through the interpreter
- Each time you run your code, it has to go through the interpreter again
- Interpreted programs are very flexible (developed and run interactively)

# Compiler



# Compilers

- Compilers are programs that translate all of your code in one go, producing binary or object code that can be understood by the processor .
- Compiling is similar to one-shot translation. You give a translator an entire document and get back the translation for the whole thing..
- Once compiled, each time you run your code, you just invoke the object code (you do not need to re-compile)
- Compiled programs are faster than interpreted ones because you only need to translate the code **once**

- Wing101 is the IDE we will be using throughout this course. You will be writing, running, and testing all your code in Wing.
- You must have Wing installed on your laptop before next week.

Download Python (Select version 3.6.4):

- https://www.python.org/downloads/

Download Wing:

- https://wingware.com/downloads/wingide-101

Online shell:

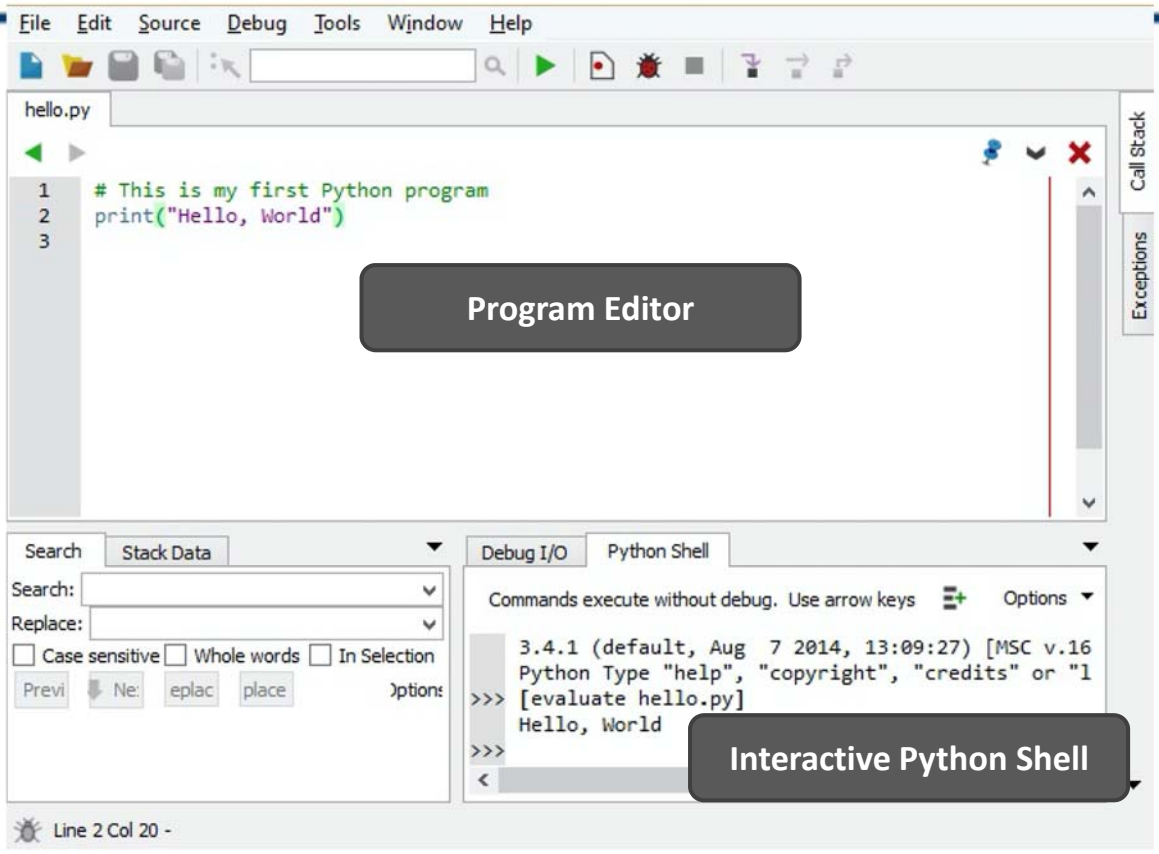
- https://www.python.org/shell/

Mobile IDE

- **Pydroid 3 - IDE for Python 3**
- https://play.google.com/store/apps/details?id=ru.iiec.pydroid3

# Let's Code!!!

# Inside the Editor (Wing IDE)



# Expressions – Integer Operations

```
>>> 5 + 2
7
>>> 3 * 4
12
>>> -5 + 3
-2
```

```
>>> 13 / 2
6.5
>>> 15 % 10
5
>>> 2 ** 3
8
```

# Expressions – Float Operations

```
>>> 5.5 + 4.5
 10.0
>>> 3.2 * 2
6.4
>>> -4.0 + 2.0
-2.0
```

```
>>> 13.0 / 2
6.5
>>> 15.0 % 2
1.0
>>> 2.0 ** 3
8.0
```

# Arithmetic Operators

| Operator | Symbol | Example | Result |
|---|---|---|---|
| - | Negation | -5 | -5 |
| * | Multiplication | 8.5 * 2.5 | 21.25 |
| / | Division | 11 / 3 | 3.66666 |
| % | Remainder | 8.5 % 3.5 | 1.5 |
| + | Addition | 11 + 3 | 14 |
| - | Subtraction | 5 – 19 | -14 |
| ** | Exponentiation | 2 ** 5 | 32 |

# Operators Precedence

| Operator | Symbol |
|---|---|
| ** | Exponentiation |
| - | Negation |
| *, /, % | Multiplication, division, and remainder |
| +- | Addition and subtraction |

- **Example:**

>>> 150 – 50 * (2 + 3)

-100

**Note:** Brackets ( ) have the highest precedence

# Variables & Assignment Statement

- A **Variable** is a name with a value associated with it
- Variables **names** can use *letters, digits*, and the *underscore* "_" symbol
- You create a new variable simply by giving it a value
- **Examples:**
  - X = 2
  - grade1 = 85.5
  - table_length = 300
- Note that Python variables are **Case Sensitive**
  - x ≠ X
  - grade ≠ Grade

# Variables & Assignment Statement

- An assignment statement is executed as follows:
  1. Evaluate the expression on the right of the **=** sign
  2. Store that value with the variable on the left of the **=** sign
- **Examples:**

| | |
|---|---|
| **x = 5 + 2** | **x = 7** |
| **y = 12 / 5** | **y = 2.4** |
| **z = 12 + 5 * 2** | **z = 22** |
| **y = x * 2** | **y = 14** |
| **x = x + 1** | **8** |
| **z += 3** | **z = 25** |

**combined operators**

# Assigning Input

- The purpose of an input statement is to get some information from the user
- **Example:**
  - x = input( )
  - x=int(input())
    - If the user enters 5 then x = 5
- Input with Prompt:
  - grade = input ("please enter a grade:")
- Note: the user can enter an expression
  - x = input("Please enter an expression:")
- If the user enters 3 + 2 then x = 3 + 2

# Simultaneous Assignment

- Python allows assigning values to multiple variables in 1 statement
- **Examples:**

```
>>> x,y = 3,5
x = 3 and y = 5
>>> x1,y1 = input("please enter 2 numbers:").split()
20   30
X1 = 20 and y1 = 30
```

# The Print Function

- The purpose of the **Print** statement is to produce output to the user
- **Examples:**

```
>>> print(3 + 5)
8
>>> x = 10
>>> print(x)
10
```

# The Math Library

- Python provides many other useful mathematical functions in a special **math library**
- A **library** is just a file that contains some useful functions
- To use a library you need to **import** the file
- **Example:**

```
from math import *
```

- After importing a library you can use all the functions defined inside

# The Math Library

| Python | Mathematics | English |
|---|---|---|
| `pi` | $\pi$ | An approximation of pi. |
| `e` | $e$ | An approximation of $e$. |
| `sin(x)` | $\sin x$ | The sine of x. |
| `cos(x)` | $\cos x$ | The cosine of x. |
| `tan(x)` | $\tan x$ | The tangent of x. |
| `asin(x)` | $\arcsin x$ | The inverse of sine x. |
| `acos(x)` | $\arccos x$ | The inverse of cosine x. |
| `atan(x)` | $\arctan x$ | The inverse of tangent x. |
| `log(x)` | $\ln x$ | The natural (base $e$) logarithm of x |
| `log10(x)` | $\log_{10} x$ | The common (base 10) logarithm of x. |
| `exp(x)` | $e^x$ | The exponential of x. |
| `ceil(x)` | $\lceil x \rceil$ | The smallest whole number $>= x$ |
| `floor(x)` | $\lfloor x \rfloor$ | The largest whole number $<= x$ |



# Using the Math Library

- **The square root function (sqrt)**

  >>> x = sqrt(9)

  x = 3.0

- **The sin function**

  >>> x = sin(30)

  -0.988031624092861 8

  >>> x = radians(30)

  >>> sin(x)

  0.4999999999999994

- **pi variable**

  >>> print(pi)

  3.14159265359

# Casting (Type Conversion)

- In Python you can convert a variable from a type to another
- To get the type of a variable use the type(var) function
- Example:

```
>>> x = 3
>>> type(x)
<type 'int'>
>>> x = float(x)
>>> type(x)
<type 'float'>
>>> str (1 + 3)
'4'
```

| float(expr( | Convert expr to a floating point value |
|---|---|
| int(expr( | Convert expr to an integer value |
| str(expr( | Return a string representation of expr |



# Input From User

```
x=input('Enter any number')
Print(x+2)
>>>TypeError: must be str, not int

x=int(input('enter num1'))
print(x+2)
>>>4
```

# Problem 1

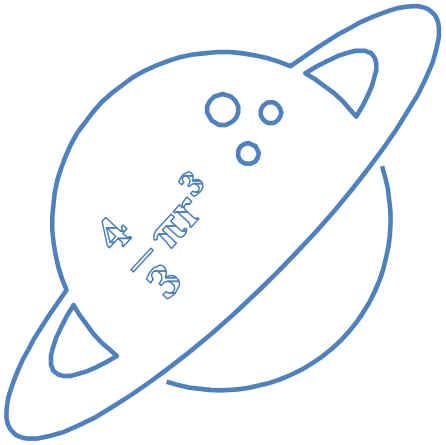
- Write a program to determine the length of a ladder required to reach a given height when leaned against a house
- The height and angle of the ladder are given as inputs

$$= nel \ \frac{hgieht}{\sin elgna}$$



# Problem 2

- Write a program that asks the user for the radius of a sphere and prints its volume



# Problem 3

- Write a program that asks the user for the radius of a sphere and prints its volume.
- $v = \frac{4}{3}\pi r^3$
- What is the volume of a sphere with a radius of 5?
- Hint: 392.6 is wrong!

# Python Tutor

- PythonTutor is an online tool for visualizing code.
- Check it at www.pythontutor.com/visualize.html
- Remember: Python executes line by line


Thanks,..
See you next week isA,..