PRG1

WEEK 2

NGEE ANN
SCHOOL OF INFOCOMM TECHNOLOGY

# Operators and Interactivity

**Programming I (PRG1)**

Diploma in Information Technology

Diploma in Financial Informatics

Diploma in Cybersecurity & Digital Forensics

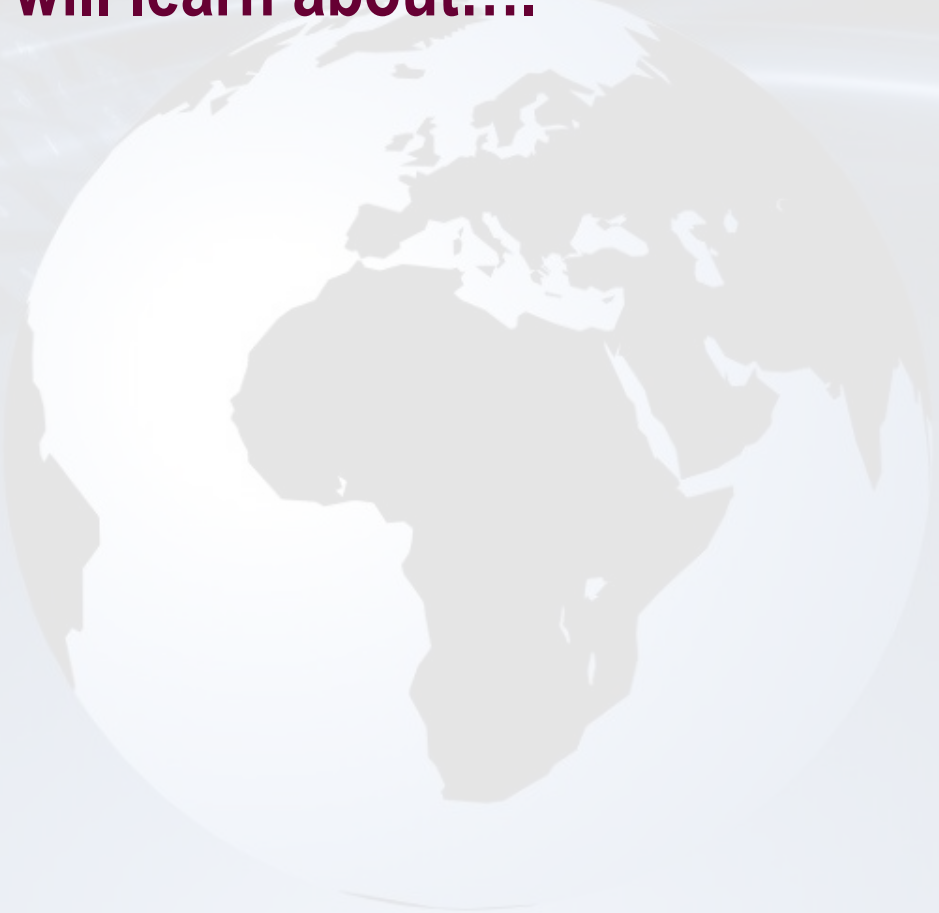
Common ICT Programme

Year 1 (2019/20), Semester 1

# Objectives

## At the end of this lecture, you will learn about….

- Strings
- Operators
- Program Interactivity
- Math Function



# Strings

# Strings

- Strings in Python can be enclosed in either single quotes (') or double quotes ("), or three of each (''') or (""")

```
>>> type('Welcome to ICT')
<class 'str'>
>>> type("Welcome to ICT")
<class 'str'>
>>> type("""Welcome to ICT""")
<class 'str'>
>>> type('''Welcome to ICT''')
<class 'str'>
```

# Strings

- Triple quoted strings can even span multiple lines:

```
>>> message="""Welcome to ICT
,hope you will enjoy
Python"""
>>> print(message)
Welcome to ICT
,hope you will enjoy
Python
>>>
```

```
>>> print("""Welcome to ICT
,hope you will enjoy
Python""")
Welcome to ICT
,hope you will enjoy
Python
```

**What is the expected output?**

```
>>>
>>> print('Welcome to ICT's graduation night! ')
```

# Operators

# Operators

- **Operators are symbols in programming languages to perform processing actions**
  - ✓ Arithmetic Operators
  - ✓ Assignment Operators
  - ✓ Relational Operators – Conditional Operators
  - ✓ Logical Operators – Conditional Operators

# Arithmetic Operators

| Operators | | Example | Given y=5; z=3 |
|---|---|---|---|
| + | addition | x = y + z | x = 8 |
| - | subtraction | x = y – z | x = 2 |
| * | multiplication | x = y * z | x = 15 |
| / | division | x = y / z | x = 1.6666666666666667 |
| % | modulus/remainder | x = y % z | x = 2 |
| // | floor division | x = y // z | x = 1 |
| ** | exponentiation | x = y ** z | x = 125 |

# Order of Arithmetic Operators

- When more than one operator appears in an expression, the order of evaluation depends on the rules of precedence.
- Order of precedence (highest to lowest) is:

| Operator | Description | |
|---|---|---|
| ( ) | Parentheses (grouping) | |
| ** | Exponentiation (raise to a power) | *right to left* |
| * / // % | Multiplication, division, floor division, modulus | *left to right* |
| + - | Addition, subtraction | *left to right* |
| = | Assignment | |

# Activity 1

- In what order would you solve this mathematical equation?

$$E_{wg} = 6.112 \times e^{(17.67 \times T_w)/(T_w + 243.5)}$$

**Where,**

$T_w$ = Wet Bulb Temperature
e = Vapor Pressure

# Mathematical Formulae to Python

| | Mathematical Formulae | Python Statement |
|---|---|---|
| 1. | y = ax + b | |
| 2. | y = (a + b) (a - b) | |
| 3. | y = 2[(a + b) (a - b)] - x | |
| 4. | $y = 1 - \frac{2a}{3b}$ | |
| 5. | a = -b | |

# Assignment Operator =

- **Assigns value to variable**
- **Assignment is from right to left**
  - ✓ E.g. What is the value of **k** after each line?

    k = 6

    k = k + 1

    - adds 1 to value of variable **k**
    - assigns new value back to the variable **k**

# Assignment Statements

## Simple Assignment

*<variable> = <expr>*

- ✓ *variable* is an identifier, *expr* is an expression
- ✓ The expression on the RHS is evaluated to produce a value which is then associated with the variable named on the LHS.

```
x = 3.9 * y * (1-z)
fahrenheit = 9/5 * celsius + 32
x = 5
```

# Assignment Statements

- **Variables can be reassigned as many times as you need during the execution of your program!**

```
>>> class_size=0
>>> class_size
0
>>> class_size=18
>>> class_size
18
>>> class_size = class_size + 1
>>> class_size
19
>>> print("The class size is", class_size)
The class size is 19
```

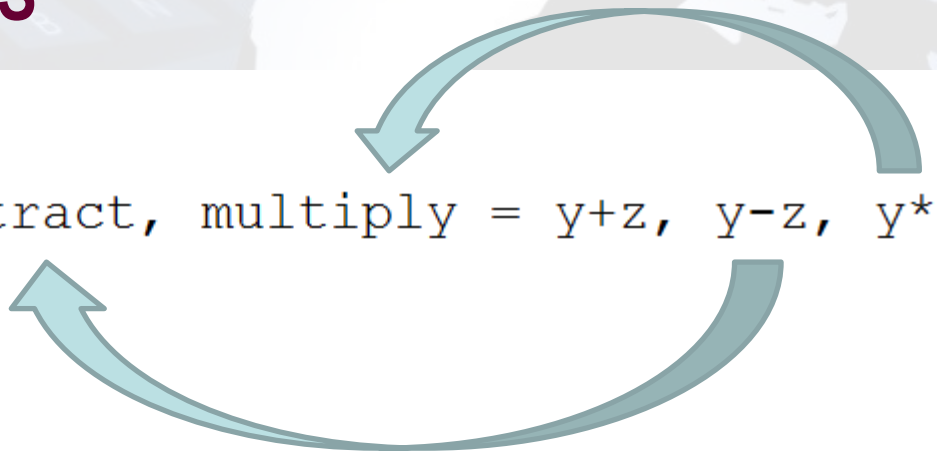
# Simultaneous Assignment

- Several values can be calculated at the same time

  <var>, <var>, ... = <expr>, <expr>, ...

- Evaluate the expressions in the RHS and assign them to the variables on the LHS

```
>>> y = 5
>>> z = 3
>>> add, subtract, multiply = y+z, y-z, y*z
>>> add
8
>>> subtract
2
>>> multiply
15
```



# Activity 2

- Given num1 = 5, num2 = 3, how would you swap the values in num1 and num2 ?

# Relational Operators

- ***Relational*** **operators compares the two operands and produce a *Boolean* value of either *True* or *False*.**

| Operator | Description |
|---|---|
| == | equal |
| != | not equal |
| > | greater than |
| >= | greater than or equal |
| < | less than |
| <= | less than or equal |

# Activity 3

- Assume x = 4, y = 3, z = 2, evaluate the following expression

| Boolean Expression | Result |
|---|---|
| x < y + z | |
| y == 2 * x + 3 | |
| z <= x - y | |
| z > x % z | |
| x != y | |

# Logical Operators

- ***Logical* operators allow us to build more complex Boolean expressions from simpler Boolean expressions.**

| Logical Operator | Description |
|---|---|
| and | produces True only if both Boolean expressions are true |
| or | produces True if either of the Boolean expressions is true |
| not | Negates a Boolean value |

# Operators - Example

- **Assume x = 1, y = 4, z = 14**

| Boolean Expression | Result |
|---|---|
| x <= 1 and y == 3 | |
| x <= 1 or y == 3 | |
| not (x > 1) | |
| not x > 1 | |
| not (x <= 1 or y == 3) | |
| x <= 1 or y > 1 and z < 1 | |
| (x <= 1 or y > 1) and z < 1 | |

# Order of Operators

## Operator Precedence, from highest to lowest:

| Operator | Description | |
|---|---|---|
| ( ) | Parentheses (grouping) | |
| ** | Exponentiation (raise to a power) | *right to left* |
| * / // % | Multiplication, division, floor division, modulus | *left to right* |
| + - | Addition, subtraction | *left to right* |
| <, <=, >, >=, !=, == | Relational operators | |
| not x | Boolean NOT | |
| and | Boolean AND | |
| or | Boolean OR | |

# Type conversion

- int(), float(), str(), and bool() convert to integer, floating point, string, and boolean (True or False) types, respectively

| Example | Output |
|---|---|
| print (4.0 / 2.0) | |
| print ( int(4) / int(2) ) | |
| print (float(4) / float(2) ) | |
| print (int (3.1415926) ) | |
| print (str(3.1415926)) | |
| print(bool(1)) | |
| print(bool(0)) | |

# Activity 4

- Given the following algebraic expressions, write the equivalent Python expression.

| Algebraic Expression | Python Expression |
|---|---|
| $y = ax^2 + bx + c$ | |
| $y = \frac{ax^3 + bx^2 + cx}{4}$ | |
| $p = \frac{1}{3} + rs - uv$ | |
| $s = \frac{4}{3(r+34)} - 9(a+bc) + \frac{3+d}{a+bd}$ | |
| | |

# Program Interactivity

# Program Interactivity

*<variable> = input(<prompt>)*

- **Built-in function in Python for getting input from user and store it into a variable.**

```
>>> name = input("Enter your name: ")
Enter your name: Joel
>>> name
'Joel'
```

- ✓ First the prompt is printed
- ✓ The input part waits for the user to enter a value and press <enter>
- ✓ The expression that was entered is captured as a string of characters and is assigned to the variable.

# Assigning Input

- You need to convert the string to int or float in order to use the numeric value.

```
>>> height = int(input("Enter your height in cm: "))
Enter your height in cm: 179
>>> height
179
```

```
>>> weight = float(input("Enter your weight in kg: "))
Enter your weight in kg: 55.8
>>> weight
55.8
```

# Example – CalBMI.py (assign inputs)

```
# This program calculates the body mass index of a person

#Input values for variables height & weight
height = float(input('Enter your height in m: '))
weight = float(input('Enter your weight in kg: '))

bmi = weight / (height * height)

#Display bmi
print('Your height is ' + str(height) + 'm')
print('Your weight is ' + str(weight) + 'kg')
print('Your bmi is ' + str(bmi))
```

**Output**

```
Enter your height in m: 1.73
Enter your weight in kg: 60
Your height is 1.73m
Your weight is 60.0kg
Your bmi is 20.04744562130375
```

```
Enter your height in m: 1.45
Enter your weight in kg: 60
Your height is 1.45m
Your weight is 60.0kg
Your bmi is 28.53745541022592
```

# Math Functions

# Math Functions

- Python comes with a large number of *modules* of its own which have functions for different purposes
- Most common module is **math** that has mathematical functions for computation.
  - ✓ Usage: **import math**

```
>>> import math
>>> math.sqrt(5)
2.23606797749979
```

```
>>> math.pi
3.141592653589793
```

```
>>> math.pow(2,2)
4.0
```

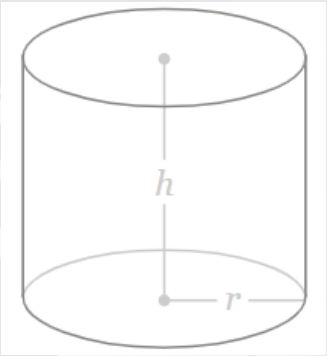
# Math Functions

| Math function | Description |
|---|---|
| factorial(x) | Returns the factorial of x |
| ceil(x) | Returns the smallest integer greater than or equal to x. |
| exp(x) | Returns e**x |
| pow(x, y) | Returns x raised to the power y |
| sqrt(x) | Returns the square root of x |
| trunc(x) | Returns the truncated integer value of x |
| pi | Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...) |
| e | mathematical constant e (2.71828...) |

**Refer to: https://docs.python.org/3/library/math.html for other math functions**

# Activity 5

**Write a program that inputs the radius and height values of a cylinder and displays the surface area of the cylinder.**




$$A = 2\pi r h + 2\pi r^2$$

# Activity 6

**Write a program that accepts the time in seconds as input and converts it into hours, minutes and seconds.**

```
Please enter the time to be converted, in sec: 8600
Time: 2 hr, 23 min 20 sec
```

# Reading Reference

- **How to Think Like a Computer Scientist: Learning with Python 3**
  - ✓ Chapter 2

http://openbookproject.net/thinkcs/python/english3e/variables_expressions_statements.html

# Summary

- Type function to find out the data type
- Operators
- Program Interactivity
- Use of Math Functions