

# ANL251 Python Programming

# Assessment Overview

| Assessment   | Description              | Weight Allocation |  |
|--------------|--------------------------|-------------------|--|
| PCOQ         | Pre-Course Quiz          | 2%                | SU1-2, before Week 1                                   |
| PCQ1         | Pre-Class Quiz 1         | 2%                | SU3-4, Week 2  |
| PCQ2         | Pre-Class Quiz 2         | 2%                | SU5-6, Week 4  |
| PCT          | Participation            | 6%                |  |
| GBA          | Group-Based Assignment   | 19%               | SU1-4, Week 5 (finalize grouping by Week 3)            |
| TMA          | Tutor-Marked Assignment  | 19%               | SU1-2, Week 3  |
| ECA          | End-of-Course Assignment | 50%               | SU1-6, after Week 6                                    |
| <b>TOTAL</b> |                          | 100%              | <b>Check Canvas announcements for exact deadlines.</b> |

## Grading Guideline (Class Participation\*)

| Indicator               | Remark  |
|-------------------------|---|
| Quality of Response     | Below average quality   |
|                         | Good and concise responses that add value to class  |
|                         | Excellent and concise responses that show good knowledge of subject and provide valuable insight to class |
| Performance Consistency | Response rate below class average   |
|                         | Similar response rate compared to Class Average   |
|                         | High response rate compared to Class Average  |

# **Study Unit 1**

## **Python Basics**

# Learning Outcomes and Learning Resources

1. Execute Python program in the Python interpreter or the PowerShell/Terminal command line
  - SU1 Chapters 1.1 ~ 1.3
  - Textbook Videos and Exercises 0 ~ 1
2. Use comments in Python scripts properly
  - SU1 Chapter 1.4
  - Textbook Video and Exercise 2
3. Solve problems using Python scripts with appropriate variable names, types and operations
  - SU1 Chapter 2
  - Textbook Videos and Exercises 3 ~ 4
  - <https://docs.python.org/3/library/stdtypes.html#numeric-types-int-float-complex>
  - <https://docs.python.org/3/reference/expressions.html#operator-precedence>

4. Construct formatted printing using format strings, the format() method and escape sequences
  - SU1 Chapter 3
  - Textbook Videos and Exercises 5 ~ 7, 9, 10
5. Create user input and implement appropriate operations based on the input
  - SU1 Chapter 4
  - Textbook Videos and Exercises 11, 12

**Seminars: discussion and activities to reinforce students' understanding**



# 1. Python Programming Environment

# Discussion

- Python Interpreter (SU1 Chapter 1.2, Textbook Video and Exercise 1)
- Execute Python program in the PowerShell/Terminal command line (SU1 Chapter 1.3, , Textbook Video and Exercise 1)
- Any other desktop Python editors or development environments?
- Are **cloud** Python development environment available?



## 2. Operators, Variables and Types

# Recap

## Operators and Data Types (SU1 Chapters 2.2, 2.3, Textbook Video and Exercise 3)

```
[>>> 2 + 2
4
[>>> 50 - 5*6
20
[>>> (50 - 5*6) / 4
5.0
[>>> 8 / 5 # division always returns a floating point number
1.6
[>>> 17 // 3 # floor division discards the fractional part
5
[>>> 17 % 3 # the modulus % operator returns the remainder of the division
2
[>>> 2 ** 7 # 2 to the power of 7
128
[>>> "Happy" + "New Year"
'HappyNew Year'
```

**Figure 1.5 Math operations in Python**

**Note:** In the examples, we can apply the plus + operator to two integers or two strings. For the integers, the values were summed, while for the strings, the strings were pasted together. The plus operator behaves differently for different data types. In Python, **how the operator behaves depends on the types applied to.**

# Discussion

Observe the code in Figure 1.5. Discuss:

- What operation does the symbol `**` represent?
- What operation does the `/` represent?
- What operation does the `//` represent?
- When applied to two int operands, which operation always evaluates to type float?
- In what orders are the operations evaluated?
- What are the three basic Python data types used?

# Discussion

## Coding style:

Put a blank space before and after every operator. For example, the first line below is good but the second line is not:


`b = 3 > x and 4 - 5 < 32`

`b= 3>x and 4-5<32`

# Discussion

## Operators

Precedence(<https://docs.python.org/3/reference/expressions.html#operator-precedence>)



|      |   |  |
|------|---|--|
| Low  | <code>or</code>   | Boolean OR   |
|      | <code>and</code>  | Boolean AND  |
|      | <code>not x</code>  | Boolean NOT  |
|      | <code>in, not in, is, is not, &lt;, &lt;=, &gt;, &gt;=, !=, ==</code> | Comparisons, including membership tests and identity tests                     |
|      | <code> </code>  | Bitwise OR   |
|      | <code>^</code>  | Bitwise XOR  |
|      | <code>&amp;</code>  | Bitwise AND  |
|      | <code>&lt;&lt;, &gt;&gt;</code>                                       | Shifts   |
|      | <code>+, -</code>   | Addition and subtraction   |
|      | <code>*, @, /, //, %</code>   | Multiplication, matrix multiplication, division, floor division, remainder [5] |
| High | <code>+x, -x, ~x</code>   | Positive, negative, bitwise NOT  |
|      | <code>**</code>   | Exponentiation [6]   |

# Quiz

| Expression     | Result |
|----------------|--------|
| 11 + 56        |        |
| 23 - 52        |        |
| 4 * 5          |        |
| 2 ** 5         |        |
| 9 / 2          |        |
| 9 // 2         |        |
| 9 % 2          |        |
| 'ab' + 'c'     |        |
| 'a' * 5        |        |
| 4 * 'bc'       |        |
| 'hello' + 5    |        |
| 'good' * 'day' |        |
| 'a' - 'b'      |        |
| 'a' / 'b'      |        |



# Quiz

What is printed by the code below?

```
print("work" + "hard" * 2 + "and" + "happily")
```

# Quiz

Select the expression(s) that result in a **SyntaxError**.

☐  $8 / (3 / (2 / 3))$

☐  $6 + -2$

☐  $4 **$

☐  $5 * (3 + 2)$

# Recap

## Variables and Types (SU1 Chapters 2.1, 2.2, Textbook Video and Exercise 4)

```
cars = 100
space_in_a_car = 4.0
drivers = 30
passengers = 90
cars_not_driven = cars - drivers
cars_driven = drivers
carpool_capacity = cars_driven * space_in_a_car
average_passengers_per_car = passengers / cars_driven

print("There are", cars, "cars available.")
print("There are only", drivers, "drivers available.")
print("There will be", cars_not_driven, "empty cars today.")
print("We can transport", carpool_capacity, "people today.")
print("We have", passengers, "to carpool today.")
print("We need to put about", average_passengers_per_car, "in each car.")
```

Figure 1.4 Creating and using Python variables

### Note:

The **= (single-equal)** assigns the value on the right to a variable on the left.

The **== (double-equal)** tests whether two things have the same value.

# Discussion

Observe the code in Figure 1.4. Discuss:

- What values do the variables refer to after the code is executed?
- What are the good variable **naming conventions**? Follow the naming conventions in your code to be submitted for TMA, GBA and ECA.

# Quiz

Are the following legal Python names?

1\_score

score1

score\_1

hours@n

cube's

cubes

Are the following Python names identical?

Seconds

seconds

# 3. Formatted Printing



# Recap

Using format strings(SU1 Chapter 3, Textbook Video and Exercise 5)

```
my_age = 35 # years
my_height = 74 # inches
my_weight = 180 # lbs
my_eyes = 'Blue'
my_hair = 'Brown'

print(f"He's {my_height} inches tall.")
print(f"He's {my_weight} pounds heavy.")
print(f"He's got {my_eyes} eyes and {my_hair} hair.")

total = my_age + my_height + my_weight
print(f"If I add {my_age}, {my_height}, and {my_weight} I get {total}.")
```

**Figure 1.6** Printing format strings

# Recap

Using the format() method of string (SU1 Chapter 3, Textbook Videos and Exercises 6, 7)

```
hilarious = False
print("Isn't that joke so funny?! {}".format(hilarious))

print("Mary had a little lamb. Its fleece was white as {}".format('snow'))
```

**Figure 1.7** Printing format strings using the format() method

# Recap

**escape sequences** (SU1 Chapter 3, Textbook Videos and Exercises 9, 10)

```
print("Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug")
print("I am 6'2\" tall.") # escape double-quote inside string
print('I am 6\'2" tall.') # escape single-quote inside string
```

**Figure 1.8** Escape sequences in Python scripts

# Quiz

## escape sequences

Select the statement(s) that will result in `SyntaxError`.

`greeting = "I'm feeling a bit lucky"`

`greeting = 'I'm feeling a bit lucky'`

`greeting = "\'m feeling a bit lucky"`

`greeting = '\'m feeling a bit lucky'`

# Quiz

What will be printed for each of the following?

```
print('How\nare\nyou?')
```

```
print('3\t4\t5')
```

```
print('\\')
```

```
print('\')
```

```
print('don\t')
```

```
print('He said, \'hi\'.')
```

```
print("It's fun!", "Don't you think?")
```

```
print("\\n is the newline character")
```

```
print("this \n is the newline character in Python")
```

# 4. User Input



# Recap

Processing user input (SU1 Chapter 4, Textbook Videos and Exercises 11, 12)

```
name = input("Name? ")
age = input("How old are you? ")
height = input("How tall are you in metres? ")
weight = input("How much do you weigh in kilograms? ")

print(f"{name} is {age} old, {height} tall and {weight} heavy.")

bmi = float(weight)/float(height)**2
print(f"Your BMI is {bmi}.")
```

Figure 1.9 User input in Python

## Note:

- The user's inputs all come in as **strings**, even if they are typed as numbers.
- How to get a number from input for math calculation? Use `int(input())` or `float(input())`, which gets the number as a string from `input()`, then converts it to an integer using `int()`, or to a float using `float()`.

# Discussion

Converting between int, float and str

After the second line of code in Figure 1.9 has been executed and the user types 21 followed by Enter/Return, what type of value does variable age refer to?

# Quiz

What is printed after executing each of the code below?

```
print('3' * 5)
```

```
print (float(str(45)))
```

```
print(str(int('99'))=='99')
```

```
print(int('-99.9'))
```

```
print(float('-9.9.9'))
```

```
print(int('7 eggs'))
```

```
print(str(int('4')+int('2'))+'eggs')
```

*Thank  
you*

