

## COP3035 - Intro to Programming in Python

### Lab Guide 1

#### Instructions:

Go over each step in sequential order.

Practice to ensure you master the skills from each objective.

Don't forget to submit in Canvas your work to get your attendance after you complete this lab.

Please reach to any of the TAs or the instructor if you have questions.

#### Introduction to Python and the Python Shell

**Objective:** Learn to execute the Python shell and perform basic arithmetic operations.

Steps:

- Open the Python shell by typing **python** in your command prompt or terminal.
- Try basic arithmetic operations: addition, subtraction, division, and exponentiation.
- Exit the shell by typing `exit()` or pressing `Ctrl+D`.

#### Creating and Running a Simple Python Program

**Objective:** Write and execute a simple Python script.

Steps:

- Open a text editor (like Notepad or any other).
- Write a simple print statement: `print("Hello, Python!")`.
- Save the file with a `.py` extension, e.g., `hello.py`.
- Run the script in the terminal by navigating to the script's directory and typing `python hello.py`.

#### Introduction to Jupyter Notebooks

**Objective:** Create, run, and understand the basics of Jupyter Notebooks.

Steps:

- Ensure you have access to Jupyter Lab, either through an installation on your computer or using a cloud service (Anaconda Cloud, Google Collab)
- Create a new notebook.
- Write a simple command like `print("Hello, Jupyter!")` in a cell and run it by pressing `Shift + Enter`.
- Add a markdown cell by selecting 'Cell' > 'Cell Type' > 'Markdown', and type some text.
- Download the notebook.
- Export the notebook to PDF.
- Upload a different notebook.

#### Variables and Types in Python

**Objective:** Understand variables, dynamic typing, and basic operations.

Steps:

- Declare a variable in a Jupyter cell, e.g., `x = 5`.
- Print and inspect its value: `print(x)`.
- Redefine the variable (
- (dynamic typing), e.g., `x = "Python"`.
- 4. Perform basic operations, e.g., `y = x + 2` (if `x` is a number).

#### Comments and Documentation

**Objective:** Learn to add comments and use markdown for documentation.

Steps:

- In a Python cell, add a comment: `# This is a comment.`
- Create a new markdown cell and add a title using `# Title`, or `## Subtitle`, `### Sub-subtitle`.

## Introduction to Strings in Python

**Objective:** Explore string declaration, printing, and basic properties.

Steps:

- Declare a string: `my_string = "Hello, World!"`.
- Print the string with an escape sequence: `print("She said, \"Hello, World!\")`.
- Experiment with combinations of the escape sequence `\`, quotes, and single quotes.

## String Operations

**Objective:** Practice string indexing, slicing, and basic methods.

Steps:

- Access individual characters: `print(my_string[1])`.
- Slice a string: `print(my_string[0:5])`.
- Use basic string methods: `print(my_string.upper())`.
- Concatenate strings: `print(my_string + " How are you?")`.
- Multiply strings: `print('Hi'*10)`

## Exercises and Examples

**Objective:** Apply learned concepts in practical scenarios.

- Write a program to calculate the sum of two numbers entered by the user.
- Create a Jupyter notebook with a markdown introduction and Python code cells for various arithmetic operations.
- In a Jupyter notebook, declare a string variable and demonstrate string slicing by extracting different parts of the string.
- Write strings in reverse.
- Practice with string methods: `upper()`, `lower()`, `strip()`, `find()`, `replace()`, `split()`.

*Remember, practice makes perfect!*