Programming For Data Science

Mid-Review

Prepared by-Ahammad Nafiz

UIU Data Science Club

Introduction to Programming

What is Programming?

- Instructing computers to solve problems using algorithms.

Why Python?

- Simple syntax, versatile applications.

Setup

- Install Anaconda, use VS Code/PyCharm.

UIU Data Science Club

First Program

```python
print("Hello, World!")  # Output: Hello, World!
```

UIU Data Science Club

Variables

Definition

- Containers for storing data.

```
message = "Hello, Python!"  # String variable
count = 10                  # Integer
```

Naming Rules

- Start with letters/underscores, case-sensitive.

UIU Data Science Club

Data Types

Common Types

- Strings ("text"), integers (5), floats (3.14), booleans (True/False).

```
average = (5 + 10) / 2  # Result: 7.5 (float)
is_valid = True         # Boolean
```

UIU Data Science Club

Operators

Arithmetic

+, -, *, /, // (floor), % (modulo).

```
print(10 // 3)  # Output: 3 (floor division)
```

Comparison

==, >, <=, etc.

```
print(5 > 3)  # Output: True
```

Logical

and, or, not.

```
print(not (5 > 3))  # Output: False
```

UIU Data Science Club

Control Flow - Conditionals

if Statement

```
grade = 75
if grade >= 60:
    print("Passed")  # Indentation required!
```

UIU Data Science Club

Control Flow - Conditionals

if-elif-else

```python
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
elif grade >= 60:
    print("D")
else:
    print("F")
```

UIU Data Science Club

Control Flow - Conditionals

Nested Conditionals

```
age = 25
income = 50000
if age > 18:
    if income > 30000:
        print("Eligible for premium")
    else:
        print("Eligible for basic")
else:
    print("Not eligible")
```

UIU Data Science Club

Conditional Expressions (Ternary Operator)

```
status = "adult" if age >= 18 else "minor"
```

UIU Data Science Club

Control Flow - Loops

for() Loop

```
for num in [1, 2, 3]:
    print(num)  # Output: 1, 2, 3
```

UIU Data Science Club

Control Flow - Loops

range()

```
for i in range(0, 5, 2):  # 0, 2, 4
    print(i)
```

Control Flow - Loops

while() Loop

```
count = 0
while count < 5:
    print(count)
    count += 1  # Output: 0, 1, 2, 3, 4
```

UIU Data Science Club

Loop Control

```
# break
for i in range(10):
    if i == 5:
        break  # Exit loop when i is 5
    print(i)  # Output: 0, 1, 2, 3, 4

# continue
for i in range(5):
    if i == 2:
        continue  # Skip when i is 2
    print(i)  # Output: 0, 1, 3, 4
```

UIU Data Science Club

Nested Loops

```
for i in range(1, 3):
    for j in range(1, 3):
        print(f"{i}×{j}={i*j}")
```

UIU Data Science Club

Lists

Creating Lists

```
empty_list = []
numbers = [1, 2, 3, 4, 5]
mixed_list = [1, "apple", 3.14, True]
nested_list = [1, ["a", "b"], [2.5, 3.5]]
```

UIU Data Science Club

Accessing Elements

```
colors = ["red", "green", "blue"]
print(colors[0])    # "red" (first element)
print(colors[-1])   # "blue" (last element)
print(colors[1:])   # ["green", "blue"] (slicing)
```

UIU Data Science Club

Modifying Lists

```
colors = ["red", "green"]
colors.append("blue")       # Add to end: ["red", "green", "blue"]
colors.insert(1, "yellow")  # Insert at index 1: ["red", "yellow", "green", "blue"]
colors.extend(["pink", "orange"])  # Add multiple items: ["red", "yellow", "green", "blue", "pink", "orange"]
```

UIU Data Science Club

Removing Elements

```
colors = ["red", "green", "blue", "yellow"]
del colors[0]        # Remove by index: ["green", "blue", "yellow"]
colors.pop()         # Remove & return last item: "yellow", list becomes
["green", "blue"]
colors.pop(0)        # Remove & return item at index 0: "green", list
becomes ["blue"]
colors.remove("blue")  # Remove by value: []
```

UIU Data Science Club

List Operations

```
nums = [1, 2, 3]
doubled = nums * 2         # [1, 2, 3, 1, 2, 3]
combined = nums + [4, 5] # [1, 2, 3, 4, 5]
```

UIU Data Science Club

List Methods

```
numbers = [3, 1, 4, 1, 5, 9]
numbers.sort()          # Sort in-place: [1, 1, 3, 4, 5, 9]
numbers.reverse()       # Reverse in-place: [9, 5, 4, 3, 1, 1]
print(numbers.count(1))  # Count occurrences: 2
print(numbers.index(5))  # Find index of first 5: 1
```

UIU
Data Science
Club

```python
# Split a list into chunks
items = [1, 2, 3, 4, 5, 6]
middle = len(items) // 2
first_half = items[:middle]  # [1, 2, 3]
second_half = items[middle:]  # [4, 5, 6]

# Split a list at a specific value
data = [10, 20, 30, -1, 40, 50]
split_index = data.index(-1)
before_split = data[:split_index]  # [10, 20, 30]
after_split = data[split_index+1:]  # [40, 50]
```

UIU
Data Science
Club

List Comprehensions

```
squares = [x**2 for x in range(5)] # [0, 1, 4, 9, 16]
evens = [x for x in range(10) if x % 2 == 0] # [0, 2, 4, 6, 8]
```

UIU
Data Science
Club

Strings

```python
s1 = "hello"
s2 = 'world'
multiline = """This is a
multiline string"""
```

UIU Data Science Club

String Operations

```
greeting = "Hello" + " " + "World"  # Concatenation
repeated = "Python! " * 3  # "Python! Python! Python! "
```

UIU Data Science Club

String Indexing and Slicing

```
word = "Python"
print(word[0])      # "P" (first character)
print(word[-1])     # "n" (last character)
print(word[0:2])    # "Py" (slice from 0 up to 2)
print(word[2:])     # "thon" (slice from 2 to end)
print(word[:3])     # "Pyt" (slice from start to 3)
print(word[::2])    # "Pto" (every 2nd character)
print(word[::-1])   # "nohtyP" (reversed)
```

UIU Data Science Club

String Methods

```
name = "ada lovelace"
print(name.upper())         # "ADA LOVELACE"
print(name.title())         # "Ada Lovelace"
print(name.strip())         # Remove whitespace
print(name.replace("a", "A"))  # "AdA lovelAce"
print(name.split(" "))      # ["ada", "lovelace"]
print("@".join(["user", "example.com"]))  # "user@example.com"
```

UIU Data Science Club

```python
# split() method - splits a string into a list
sentence = "Python is fun to learn"
words = sentence.split()  # Default splits by whitespace: ["Python", "is", "fun", "to",
"learn"]

csv_data = "apple,banana,cherry"
fruits = csv_data.split(",")  # Split by comma: ["apple", "banana", "cherry"]

date = "2023-04-15"
year, month, day = date.split("-")  # Split by dash: year="2023", month="04", day="15"

print("Hello, World".find("World"))  # 7 (index where found)
```

UIU Data Science Club

String Checks

```
print("123".isdigit())     # True
print("abc".isalpha())     # True
print("Python".startswith("Py"))  # True
print("code".endswith("de"))      # True
print("Hello, World".find("World"))  # 7 (index where found)
```

UIU Data Science Club

Formatting

```
name = "Alice"
age = 30
# f-strings (Python 3.6+)
print(f"{name} is {age} years old")
# format() method
print("{} is {} years old".format(name, age))
# Named placeholders
print("{name} is {age} years old".format(name=name, age=age))
```

UIU Data Science Club

Immutability & Workarounds

```
s = "hello"
# s[0] = "H"  # Error! Strings can't be changed
s = "H" + s[1:]  # Create new string: "Hello"
```

UIU Data Science Club

Functions - Basics

Definition and Calling

```
def greet(name):
    print(f"Hello, {name}!")

greet("Alice")  # Output: Hello, Alice!
```

UIU
Data Science
Club

Return Values

```
def square(n):
    return n * n

result = square(4)  # result = 16
```

UIU Data Science Club

Multiple Return Values

```
def get_coordinates():
    return 10, 20

x, y = get_coordinates()  # x = 10, y = 20
```

UIU Data Science Club

Variables Scope

```
def scope_example():
    local_var = "I'm local"
    print(local_var)

scope_example()
# print(local_var)  # Error - not defined outside
function
```

UIU Data Science Club

Functions - Parameters

Positional Arguments

```
def subtract(a, b):
    return a - b

result = subtract(10, 5)  # result = 5
result = subtract(5, 10)  # result = -5 (order matters!)
```

UIU Data Science Club

Keyword Arguments

```
def describe_pet(animal, name):
    print(f"{name} is a {animal}.")

describe_pet(name="Buddy", animal="dog")  # Order doesn't
matter
```

UIU Data Science Club

Default Values

```python
def info(country, capital="unknown"):
    print(f"Capital of {country}: {capital}")

info("Canada")  # Capital of Canada: unknown
info("France", "Paris")  # Capital of France: Paris
```

UIU Data Science Club

Lambda Functions

```
# Named function
def double(x): return x * 2

# Equivalent lambda
double_lambda = lambda x: x * 2

# Common with functions like map, filter
numbers = [1, 2, 3, 4]
squared = list(map(lambda x: x**2, numbers))  # [1, 4, 9, 16]
evens = list(filter(lambda x: x % 2 == 0, numbers))  # [2, 4]
```

UIU Data Science Club

Tricky Problems

UIU
Data Science
Club

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# What's the output?
print(numbers[::2])
print(numbers[::-1])
print(numbers[-3::-2])
print(numbers[5:1:-1])
```

UIU
Data Science
Club

```
# What's the final value of s?
s = "hello"
s = s.replace("l", "L")
s = s.upper()[:3] + s[3:]
print(s)
```

UIU
Data Science
Club

```
# What's the output?
x = 0
while x < 10:
    x += 2
    if x == 6:
        continue
    if x == 8:
        break
    print(x, end=" ")

# What's the sum?
total = 0
for i in range(1, 5):
    if i % 2 == 0:
        total += i
    else:
        total += i**2
print(total)
```

UIU
Data Science
Club

```python
def add_item(item, lst=[]):
    lst.append(item)
    return lst

print(add_item(1))
print(add_item(2))
print(add_item(3, []))
print(add_item(4))
```

UIU
Data Science
Club

```python
word = "hello"
letters = list(word)

for c in word:
    letters.pop(0)

print(letters)
```

UIU
Data Science
Club

Thank You

UIU
Data Science
Club