Fourth Industrial Summer School

Day 2

Python Programming II

Session Objectives

- ✓ Conditional statements
  - if
- ✓ Loops
  - while
  - for

Part A

Conditional statements

Conditions

- Similar to other programming languages (if statements)
  - No switch support

- Python relies on indentation
  - Using whitespace to define code scope.
  - Other programming languages often use {}

```
ahmedAge = 30
farisAge = 30
if ahmedAge > farisAge:
  print("Ahmed is older than Faris")
elif ahmedAge < farisAge:
  print("Ahmed is older than Faris")
else:
  print("They have the same age")
```

```
They have the same age
```

Conditions

- Logical Operators

Operator	Meaning
==	equals
!=	Not equals
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Conditions

- **Compound** conditions uses the following logical operators to combine conditions
  - or
  - and

```python
num1 = 100
num2 = 200

if num1 > 0 and num2 > 0:
  print("Both numbers are positive")
else:
  print("Either numbers is negative")
```

```
Both numbers are positive
```

AND	T	F
T	T	F
F	F	F

OR	T	F
T	T	T
F	T	F

Hands on session

Problem Solving

Part B

Loops

Loops

- Two primitive loops in Python
  - while
    - Used to iterate over numbers.
  - for
    - Used to iterate over a list, a tuple, a dictionary, a set, or a string.

```python
num = 0
while num < 6:
  print(num)
  num += 1
```

```
0
1
2
3
4
5
```

While

- Two statements used in loops
  - break
  - continue

```
num = 0
while num < 6:
  if num == 4:
    break
  print(num)
  num += 1
```

```
0
1
2
3
```

```
num = 0
while num < 6:
  num += 1
  if num == 4:
    continue
  print(num)
```

```
1
2
3
5
6
```

For

- Iterating over a list

```
cars = ["BMW", "Lexus", "Honda"]
for x in cars:
  print(x)
```

```
BMW
Lexus
Honda
```

- No indexing is required beforehand
- Iterating over a String

```
for x in "Honda":
  print(x)
```

```
H
o
n
d
a
```

For

- We can use the break and continue statement

```
cars = ["BMW", "Lexus", "Honda"]
for x in cars:
  if x == "BMW":
    continue
  print(x)
```

```
Lexus
Honda
```

```
cars = ["BMW", "Lexus", "Honda"]
for x in cars:
  if x == "Lexus":
    break
  print(x)
```

```
BMW
```

For

- else after a for loop
  - Can be used to execute a set of code after loop

```
cars = ["BMW", "Lexus", "Honda"]
for x in cars:
  print(x)
else:
  print("Done printing")
```

```
BMW
Lexus
Honda
Done printing
```

For

- Nested loops
  - Loop inside a loop

```
cars = ["BMW", "Lexus", "Honda"]
colors = ["Red", "Blue"]
for x in cars:
  for y in colors:
    print(x,y)
```

```
BMW Red
BMW Blue
Lexus Red
Lexus Blue
Honda Red
Honda Blue
```

Hands on session

Problem Solving