



# Lecture slides - Week 4

## Programming in Python - Control Flow

---

Dr. Aamir Akbar

Director of both [AWKUM AI Lab](#) and [AWKUM Robotics](#), [Final Year Projects \(FYPs\)](#) coordinator,  
and lecturer at the department of Computer Science  
Abdul Wali Khan University, Mardan (AWKUM)

1. Control Flow
2. Problem Solving Approach with Examples
3. Download Source Codes

# Control Flow

---

# Decision Making in Python i

In Python, decision making is achieved using the following control flow statements:

- **if statement:** It is used to execute a block of code only if a condition is true.

```
1  if condition:  
2      # Code to execute if the condition is true
```

The code inside the if block is indented. In Python, indentation is used to group statements together. Indentation in Python is typically done using spaces or tabs, but it must be consistent within a block of code.

- **else statement:** It is used in combination with the if statement to execute a block of code when the condition is false.

```
1  if condition:  
2      # Code to execute if the condition is true  
3  else:  
4      # Code to execute if the condition is false
```

# Decision Making in Python ii

- **elif statement:** Short for *else if*, it allows you to check multiple conditions in sequence.

```
1  if condition1:
2      # Code to execute if condition1 is true
3  elif condition2:
4      # Code to execute if condition2 is true
5  else:
6      # Code to execute if no conditions are true
```

A coding example illustrating the use of the above statements or decision making in Python:

```
1  if age < 18:
2      print("You are a minor.")
3  elif age >= 18 and age < 65:
4      print("You are an adult.")
5  else:
6      print("You are a senior citizen.")
```

# Loops in Python i

In Python, loops are used to execute a block of code repeatedly. There are two main types of loops: **for** and **while** loops.

- **for loop:** It is used when you know in advance how many times you want to execute a block of code. The loop iterates over a sequence (e.g., a list, tuple, or string) or a range of numbers.

```
1  for item in sequence:  
2      # Code to execute for each item in the sequence
```

- **while loop:** It is used when you want to execute a block of code as long as a condition is true. The loop continues to execute until the condition becomes false.

```
1  while condition:  
2      # Code to execute while the condition is true
```

# Loops in Python ii

Coding examples illustrating the use of loop types:

**For Loop Example** - Printing numbers from 1 to 5:

```
1 for num in range(1, 6):  
2     print(num)
```

**While Loop Example** - Counting down from 3 to 1:

```
1 count = 3  
2 while count > 0:  
3     print(count)  
4     count -= 1
```

# Logical and Comparison Operators i

Name	Python Operator	Python Example	C++ Operator	C++ Example
Logical AND	and	x and y	&&	x && y
Logical OR	or	x or y		x    y
Logical NOT	not	not x	!	!x

**Table 1:** Logical Operators in Python and C++



## Logical and Comparison Operators ii

Name	Python Operator	Python Example	C++ Operator	C++ Example
Equal to	<code>==</code>	<code>x == y</code>	<code>==</code>	<code>x == y</code>
Not equal to	<code>!=</code>	<code>x != y</code>	<code>!=</code>	<code>x != y</code>
Less than	<code>&lt;</code>	<code>x &lt; y</code>	<code>&lt;</code>	<code>x &lt; y</code>
Greater than	<code>&gt;</code>	<code>x &gt; y</code>	<code>&gt;</code>	<code>x &gt; y</code>
Less than or equal to	<code>&lt;=</code>	<code>x &lt;= y</code>	<code>&lt;=</code>	<code>x &lt;= y</code>
Greater than or equal to	<code>&gt;=</code>	<code>x &gt;= y</code>	<code>&gt;=</code>	<code>x &gt;= y</code>

**Table 2:** Comparison Operators in Python and C++

# Problem Solving Approach with Examples

---

# Problem-Solving Approach: Odd or Even Numbers

**Problem Statement:** In a sequence of numbers (i.e., a range), identify each number as either odd or even.

**Coding Example:**

# Problem-Solving Approach: Odd or Even Numbers

**Problem Statement:** In a sequence of numbers (i.e., a range), identify each number as either odd or even.

## Coding Example:

```
1 # Iterate through the range and identify numbers
2 for num in range(10):
3     if num % 2 == 0:
4         print(f"{num} is even")
5     else:
6         print(f"{num} is odd")
```

# Problem-Solving Approach: Multiples of 3 and 5

**Problem Statement:** In a range of numbers, identify numbers that are multiples of both 3 and 5.

**Coding Example:**

# Problem-Solving Approach: Multiples of 3 and 5

**Problem Statement:** In a range of numbers, identify numbers that are multiples of both 3 and 5.

## Coding Example:

```
1 # Iterate through the range and identify numbers
2 for num in range(1, 21):
3     if num % 3 == 0 and num % 5 == 0:
4         print(f"{num} is a multiple of both 3 and 5")
5     else:
6         print(f"{num} is not a multiple of both 3 and 5")
```

# Problem-Solving Approach: Grade Classification

**Problem Statement:** Given a student's marks, classify it into a grade category.

**Coding Example:**

# Problem-Solving Approach: Grade Classification

**Problem Statement:** Given a student's marks, classify it into a grade category.

## Coding Example:

```
1 # Get the student's marks
2 marks = int(input("Enter the student's marks: "))
3
4 # Classify the marks into a grade category
5 if marks >= 90:
6     grade = "A"
7 elif marks >= 80:
8     grade = "B"
9 elif marks >= 70:
10    grade = "C"
11 elif marks >= 60:
12    grade = "D"
13 else:
14    grade = "F"
15
16 # Print the grade
17 print(f"The student's grade is {grade}")
```



# Problem-Solving Approach: Guess the Number (Problem)

**Problem Statement:** Implement a number guessing game. The computer selects a random number between 1 and 100, and the player tries to guess it. Provide feedback if the guess is too high or too low.

# Problem-Solving Approach: Guess the Number (Solution)

```
1  import random
2
3  # Generate a random number between 1 and 100
4  target_number = random.randint(1, 100)
5
6  # Initialize the guess and attempts
7  attempts = 0
8  guess = 0
9
10 # Start the guessing game
11 while guess != target_number:
12     guess = int(input("Enter your guess: "))
13     attempts += 1
14
15     if guess < target_number:
16         print("Too low! Try again.")
17     elif guess > target_number:
18         print("Too high! Try again.")
19     else:
20         print(f"Congratulations! You guessed the number {target_number}
21               in {attempts} attempts.")
```