# Introduction to Computer Programming(Group 5)

***Team Members***

1.Tona Elsie 29743 -
2. Mugabo Imena Axel 28768  -
3.NIYONSENGA Francine 29704
4. TETA Agnes : 29526 -
5.Ahishakiye Germain 28393 -
6. Rukiriza happy Patrick 29738-
7. Murenzi Paterne 28302 -
8. NTARINDWA Jackson 25593
9.Micomyiza justin 28780 -

# Control structures: Sequences, Selection/Decision making, and Iterations.

We will  be talking about Control Structures in programming —these are the rules that tell a computer what to do, when to do it, and how to do it.”

**What Are Control Structures?**
In programming, we use control structures to control the flow of actions, They decide which instruction comes first, which one to skip, and which to repeat, Without them, a program would just run line by line — no decisions, no logic.

**There are three main types:**

1. **Sequence:** The program runs step by step in order, Example: Following a recipe (boil water → add tea → serve).
2. **Selection:** The program chooses what to do based on a condition.
3. **Iteration:**  The program repeats actions until a condition is met

Control structures make programs **logical**, **organized**, and **intelligent** — allowing them to decide and repeat actions just like humans do.

# Sequence Structure

Let’s start with the simplest one — Sequence, It’s like following a list of steps in order, from top to bottom, without skipping any.

**Example:** Boil water, Add tea leaves

In code based here is the example

```
print("Enter your name")

name = input()

print("Hello", name)
```

The computer runs each line one after another, just like following a recipe, Sequence makes sure things happen in the right order.”

# Sequence structure Continuation

“If you skip or mix the order, your result can be wrong. **Like an Example:** You can’t type a document before turning on the computer!

**Why Sequence Is Important**

- Sequence makes sure things happen in the right order.
- If you skip or mix the order, your result can be wrong.

**Example:** You can’t type a document before turning on the computer!

- Ensures steps are followed correctly and orderly
- Prevents errors caused by skipping steps
- Used in most programs as the foundation for logic

**Example:**

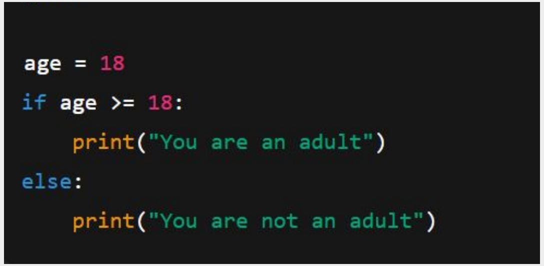
- You must open your document before typing — that’s sequence in action.

# Selection (Decision Making)

Selection allows a program to choose between two or more paths based on a condition, It helps programs make decisions automatically.

**Types of Selection**

1. If Statement – Executes when a condition is true.
2. If-Else Statement – Chooses between two options.
3. Nested If – Checks multiple conditions inside another.
4. Switch cases - lets your program choose what to do based on one value



```
age = 18
if age >= 18:
    print("You are an adult")
else:
    print("You are not an adult")
```

# Selection continuation ....

a) **Simple If Statement**
Checks one condition.
If true → execute a block of code.
If false → skip it.

b) **If-Else Statement**
Provides two options: one block executes if the condition is true, another if false.
Output depends on the condition being true or false.

c)**Else-If / Elif / Nested If**
Used when there are multiple conditions.
Checks conditions in sequence, executes the first one that is true. The program checks each condition one by one.Stops at the first condition that is true.

# Selection continuation ....

A **switch case** lets your program **choose what to do based on one value** — like a cleaner version of many `if...else` statements.

## How it works

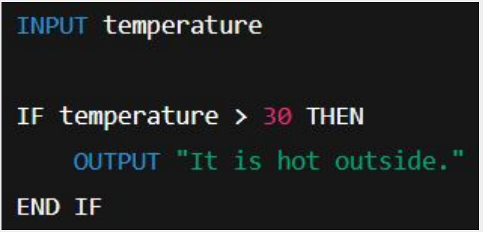
- The **switch** looks at a value (like a number or word).
- Each **case** checks if the value matches something.
- The program runs the code under the first matching case.
- The **break** stops it from running the rest.
- If no case matches, the **default** runs.

```
SET grade TO "B"

SWITCH grade
    CASE "A":
        OUTPUT "Excellent"
        BREAK
    CASE "B":
        OUTPUT "Good"
        BREAK
    CASE "C":
        OUTPUT "Average"
        BREAK
    DEFAULT:
        OUTPUT "Invalid grade"
ENDSWITCH
```

# Examples of Selection using programming

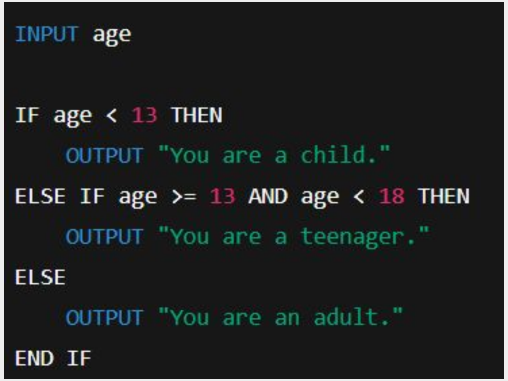
Simple if code statement



```
INPUT temperature

IF temperature > 30 THEN
    OUTPUT "It is hot outside."
END IF
```

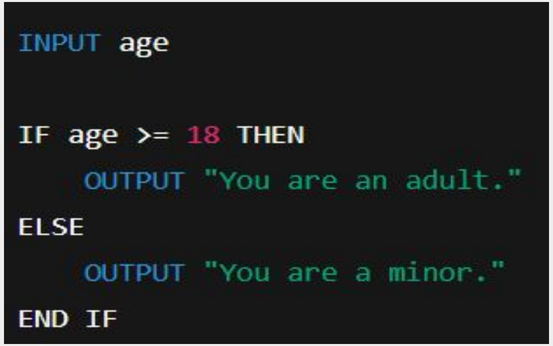
Nested-if Statement



```
INPUT age

IF age < 13 THEN
    OUTPUT "You are a child."
ELSE IF age >= 13 AND age < 18 THEN
    OUTPUT "You are a teenager."
ELSE
    OUTPUT "You are an adult."
END IF
```

Else statement



```
INPUT age

IF age >= 18 THEN
    OUTPUT "You are an adult."
ELSE
    OUTPUT "You are a minor."
END IF
```

# Iteration Loop

**What is Iteration?**

- Iteration is a control structure that allows a program to repeat a set of instructions multiple times.
- Instead of writing the same code over and over, we use loops.
- Iteration saves time and makes programs efficient.

**Example:**

Watering 5 plants → you repeat the same action 5 times.
Checking each student's grade in a class → repeat the same steps for each student.

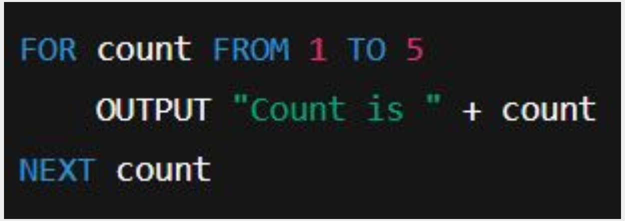
**Types of Loops**

For Loop – Repeats a known number of times.
While Loop – Repeats as long as a condition is true.
Do-While Loop – Runs once, then checks the condition (in some languages).

# Iteration Cont…

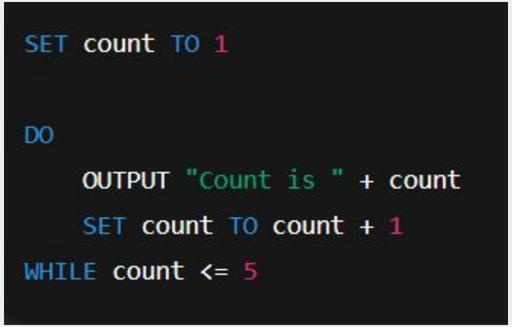
For Loop: Repeats a known number of times.

**Example:** Countdown from 5



```
FOR count FROM 1 TO 5
    OUTPUT "Count is " + count
NEXT count
```

**Do-while loop:** Executes at least once, then checks the condition.

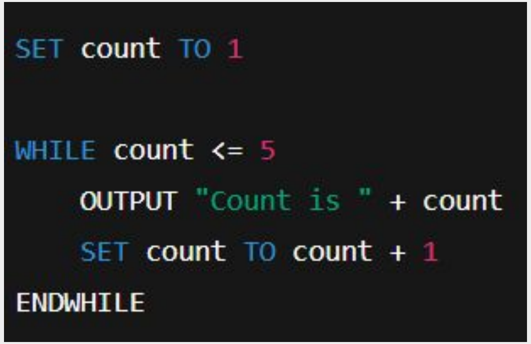


```
SET count TO 1

DO
    OUTPUT "Count is " + count
    SET count TO count + 1
WHILE count <= 5
```

While Loop:Repeats until a condition becomes false.

**Example:** Print numbers 1 to 5



```
SET count TO 1

WHILE count <= 5
    OUTPUT "Count is " + count
    SET count TO count + 1
ENDWHILE
```

# Thank You !

Any Questions?