ICP GROUP 3

NAMES	ID
ABIJURU JONATHAN	29071
Rumanyika Arthur	28944
Shema Aristide	27888
SANGANO Chris	28927
Ishimwe Sabin	28958
JOHNSON John	28358
UWIMBABAZI Alliance	28756
NDAHIMANA Gate Sagai	28536
keza kendra uluat	29083
ISHIMWE Wycliffe	29022
IZIHIRWE Nicia	29285

What Are Control Structures?

• Control structures are **fundamental constructs that determine the order in which instructions are executed**.

• They allow programmers to **manage the flow of a program**.
• Control structures enable programs to:
  ◦ Make decisions.
  ◦ Repeat actions.
  ◦ Execute code in a specific sequence.

• **Crucial Role:** Without them, programs would only execute one instruction after another, making it impossible to solve complex, real-world problem

The Three Primary Types of Control

types	Core function	Key characteristics
sequence	Direct linear flow	Instructions execute one after another in the order they appear
selection	Decision-making logic	Programs make choices based on conditions
Iteration	Repeating blocks of code	Allow for efficient code reuse and processing large datasets

Sequence

- **Definition:** Sequence is the simplest control structure where instructions execute in the exact order they're written.
- **Characteristics:**
  - Direct linear flow.
  - No branching or repetition.
  - It is the foundation of all programs.
- **Applications:** Perfect for basic operations like calculations, reading input, and displaying results.
- **Pseudocode Example:**

```
Start
Diclare a,b,sum as integer
Display "enter numbers"
Read a,b
Sum<- a+b
WRITE "sum is:",sum
end
```

Selection

Decision-Making Logic

- Selection structures allow programs to **choose between different paths based on conditions**. This is how programs become truly intelligent.
- **Key Features:**
  - Multiple pathways are possible.
  - Uses IF statements and conditions.

Structure Type	Description
IF	Executes code only when a condition is true
IF...ELSE	Provides an alternative path when the condition is false
Nested IF	Incorporates multiple conditions within each other for complex logic
Switch statement	Selects one option from many possible choices

Examples

```
IF Mark >= 50 THEN
    Output "Pass"
ENDIF
```

```
IF Age >= 18 THEN
    Output "Eligible to vote"
ELSE
    Output "Not eligible"
ENDIF
```

```
IF Score >= 80 THEN Output "A"
ELSEIF Score >= 60 THEN Output "B"
ELSE Output "C"
ENDIF
```

```
SWITCH (variable) CASE value1:
Statement1 CASE value n:
Statement n DEFAULT
Statement
```

Iteration

Iteration involves **repeating blocks of code until specific conditions are met**

Benefits:
- Efficient code reuse.
- Processing large datasets.
- **Goal:** Achieved through loops and repetition.

- **Three Main Loop Types:**
  1. WHILE Loop (Pre-test)
  2. DO WHILE (Post-test)
  3. FOR Loop (Fixed count)

Loop Type	Characteristics	Pseudocode Example
WHILE Loop	A **Pre-test loop** that repeats whilst the condition remains true	`WHILE Count <= 10 Output Count` `Count ← Count + 1 ENDWHILE`

DO WHILE	A **Post-test loop** that executes at least once before checking the condition.	`DO PRINT "Count:"counter Counter+1`
		` WHILE`
		`(counter <= 5)`

FOR Loop	Perfect when you **know exactly how many times to repeat**.	`FOR i=1 to 5 Do Print` `"hello" End for`

Real-World Applications

Control Structures in Action

1. **ATM Systems**
   - **Selection** for menu choices.
   - **Loops** for transaction processing.
   - **Sequence** for card validation.
2. **Traffic Management**
   - Timed **sequences**.
   - **Conditional** responses to sensors.
   - Continuous **monitoring loops**.
3. **Gaming Logic**
   - Player choices (**Selection**).
   - Scoring systems.
   - Game **loops**, and level progression algorithms

Master Control Structures for Success

Mastering these structures provides significant programming advantages:

• **Build Logical Thinking:** Control structures develop problem-solving skills essential for programming.

• **Write Maintainable Code:** Structured programs are easier to read, debug, and enhance.

• **Solve Complex Problems:** They transform simple ideas into powerful, intelligent applications