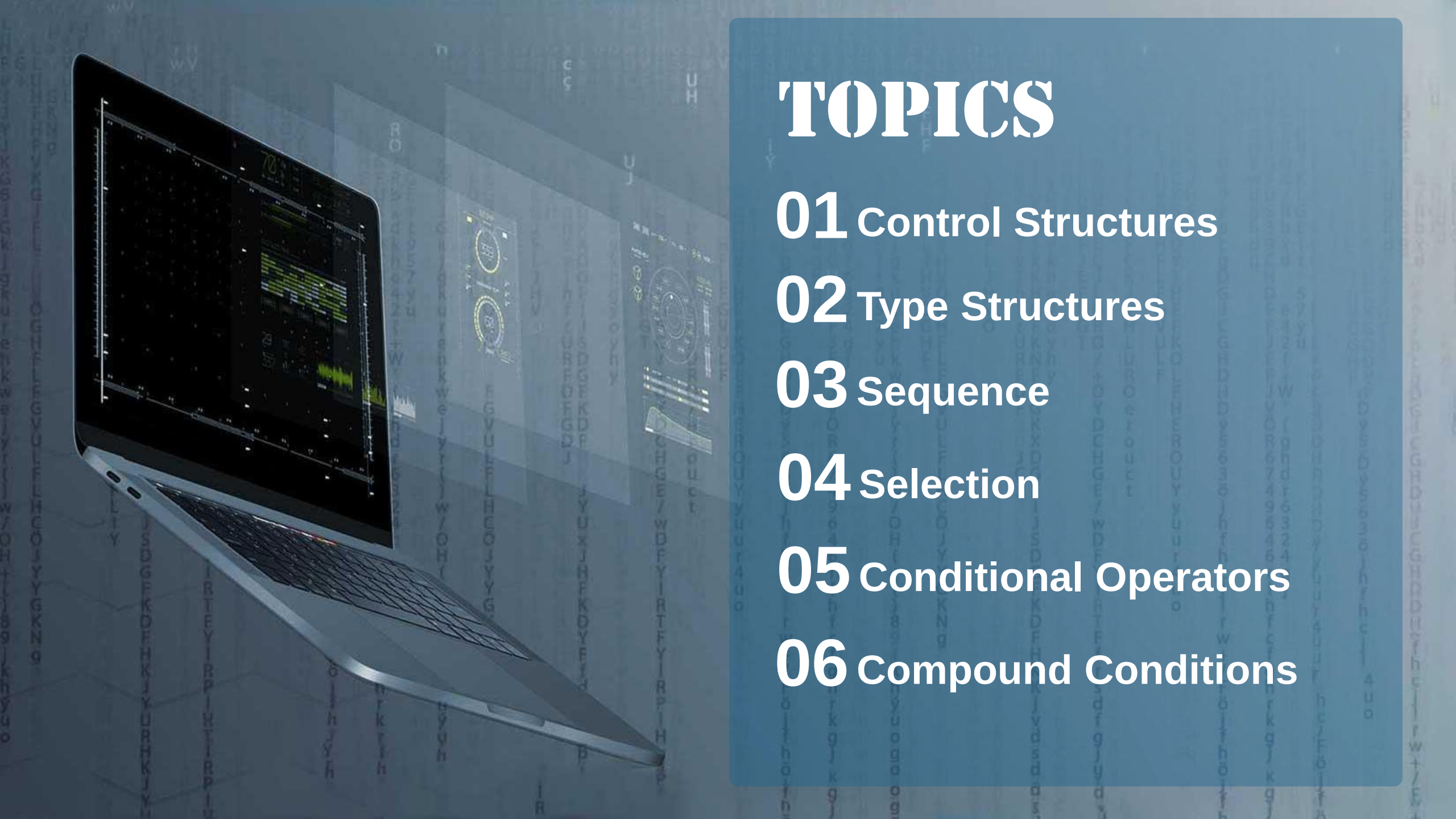
ALLPPT.com

TOT – Python Programming Essentials

DAY 3 – CONTROL STRUCTURES (SELECTION)



TOPICS

01 Control Structures

02 Type Structures

03 Sequence

04 Selection

05 Conditional Operators

06 Compound Conditions

CONTROL STRUCTURES

A control structure (or flow of control) is a block of programming that analyses variables and chooses a direction in which to go based on given parameters.

The basic decision-making process in programming and flow of control determines how a computer program will respond when given certain conditions and parameters.

TYPES OF CONTROL STRUCTURES

Sequential

Sequential execution is when statements are executed one after another in order. You don't need to do anything more for this to happen.

Selection

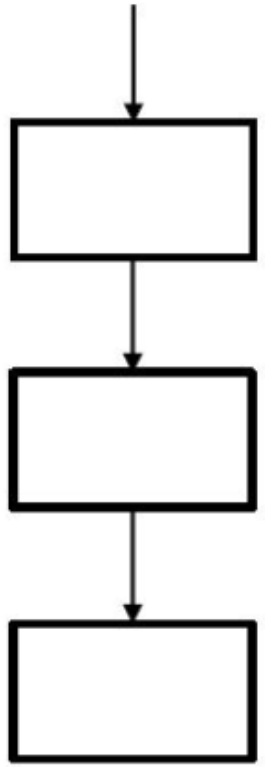
Selection used for decisions, branching – choosing between 2 or more alternative paths.

Iteration

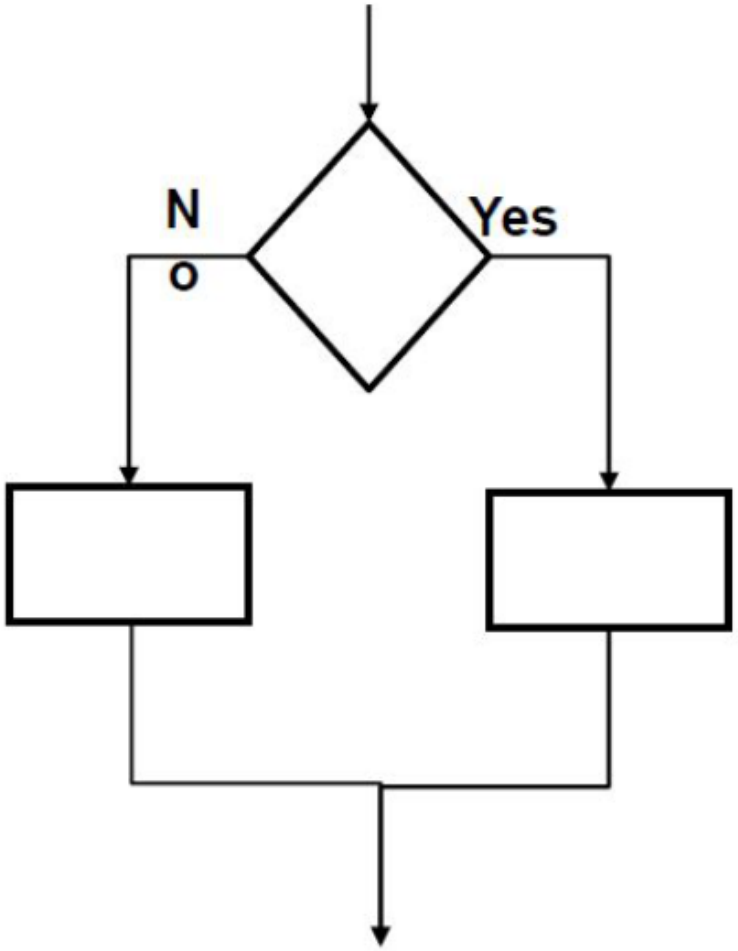
Repetition used for looping, i.e. repeating a piece of code multiple times in a row.

TYPES OF STRUCTURE {}

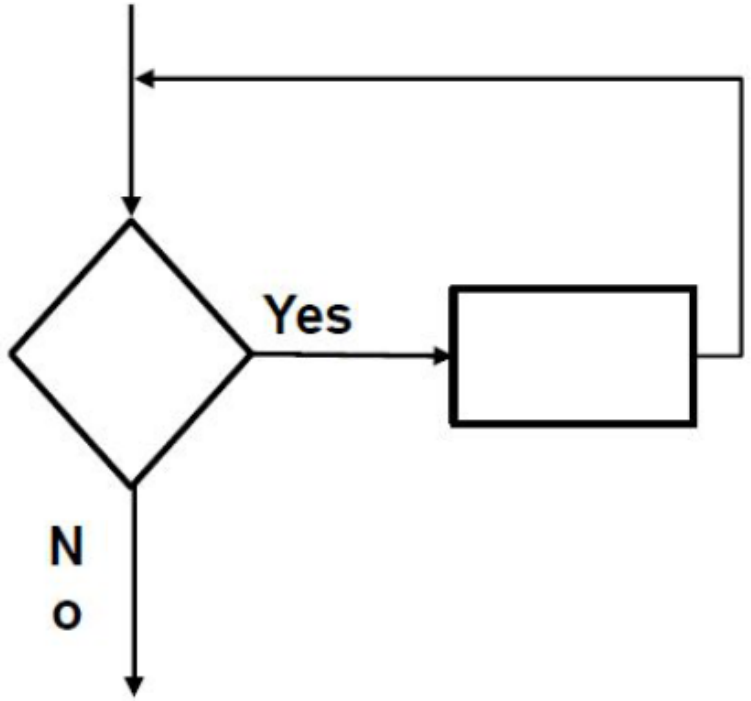
Sequence

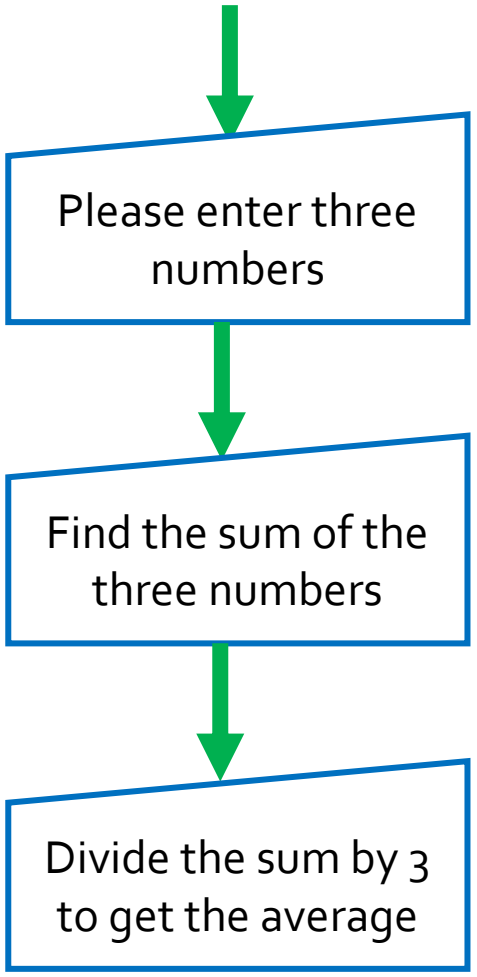


Selection



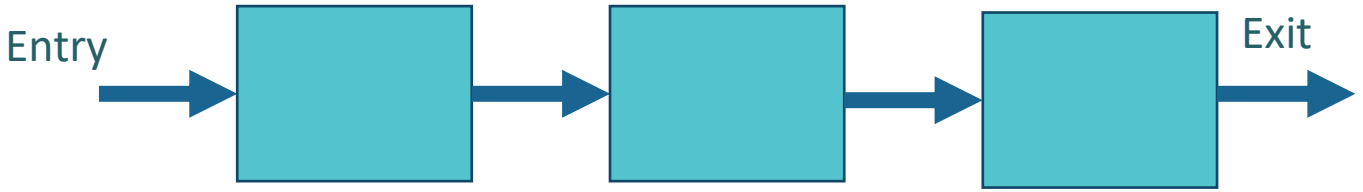
Iteration





SEQUENCE

- The simplest structure
- Control flows from one statement to the next
- Statements are executed sequentially in the order in which they appear in the program



EXAMPLE: SEQUENCE

A program that computes and prints the net pay of an employee contains the following code:

```
grossPay = 20000                          # 1st statement to be executed
netPay = grossPay – 4000                  # 2nd statement to be executed
print(“The net pay is: “ + netPay);       # 3rd statement to be executed
```

SELECTION

The selection statements are also known as decision making statements or branching statements.

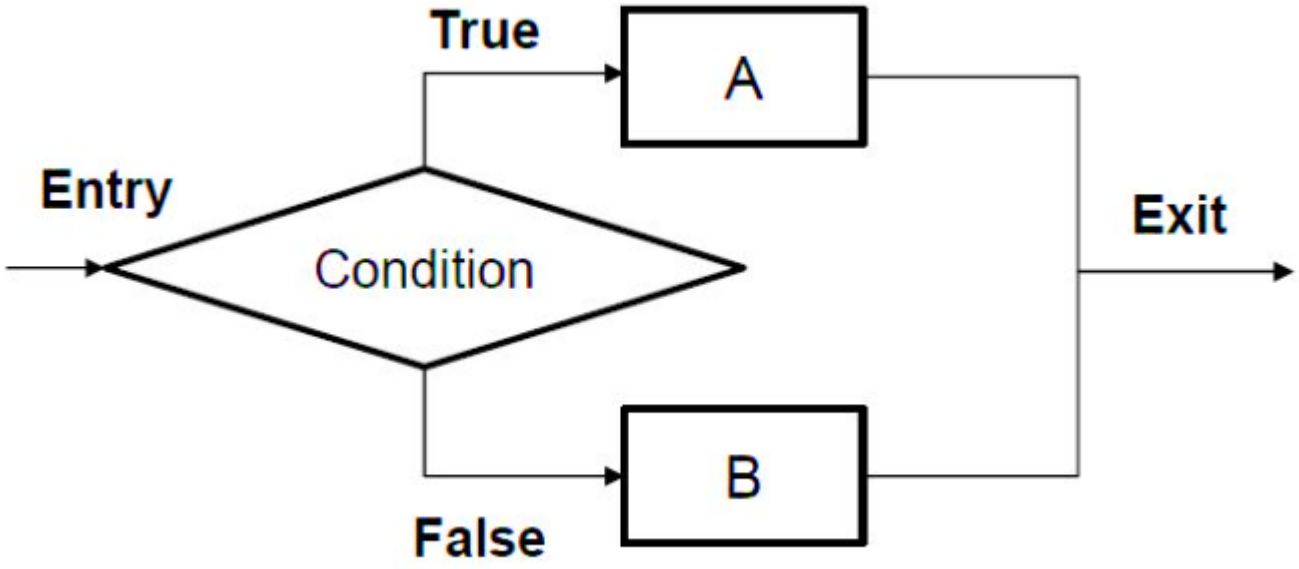
Provides a choice between two alternatives

3 Components of the structure

- A condition to be tested
- The statement to be performed if the condition is satisfied (process A)
- The statement to be performed if the condition is not satisfied (process B)

SELECTION

- Entry to the structure is through the condition.
- Exit is through the execution of process A or process B



SELECTION

Program code for single alternative selection:

```
if condition:
   statement1

if condition:
   statement1
   statement2

   where condition is a Boolean expression or Boolean variable
```

SELECTION

Program code for multiple true selection:

```
if condition:
   statement1
elif condition:
   statement2
else:
   statement3
```

CONDITIONAL OPERATORS

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

SOME COMMON ERRORS

Using > instead of >=

Example: "Write a program that would give customers over age 60 a discount of 20%"

- The phrase "over age 60" is not clear. Does it include the age 60 or not? Clarify what it means.
- Similar phrases that have unclear meaning are: "not more than", "at least", "not under".

DAY 3
ACTIVITY 1

- Write a program that will compute for the students average
- The user will input the following:
  - ❑ Name
  - ❑ Math
  - ❑ Science
  - ❑ English
- If the average is equal and above 75 status is Passed else You Failed the semester

```
Name: Vinz
Math: 80
Science: 90
English: 85
Average: 85.0

POSSIBLE OUTPUT:
Output 1:
Congratulations! You passed
the semester. But you need
to re-enroll English subject

Output 2:
Sorry, You Failed the
semester
```

DAY 3 ACTIVITY 1

COMPOUND CONDITIONS

- Have more than 1 conditional expression
- The result of the compound expression depends on the individual result of each condition

Format:

```
(condition1) and (condition2)
(condition1) or (condition2)
```

DAY 3
ACTIVITY 2

- Write a program that will check the employees years in service and office.
- The user will input number for years and in service and the following offices (IT, ACCT, HR)
- Check the following conditions

Office	Years	
	more than or equal 10 years	below 10 years
IT	10000	5000
acct	12000	6000
hr	15000	7500

DAY 3 ACTIVITY 2

BUILT-IN METHODS

- isupper() - Returns True if all characters in the string are upper case
- islower() - Returns True if all characters in the string are lower case
- isdigit() - Returns True if all characters in the string are digits
- isalpha() - Returns True if all characters in the string are in the alphabet