**Day 1 27 0ctober 2023**

**Control flow statements**

A program’s control flow in the order in which the program’s code executes.

Python has 3 control structures:

* Sequential – default mode
* Selection – used for decisions and branching
* Repetition – used for looping i.e repeating a piece of code multiple times

Sequential

Sequential statements are a set of statements whose execution process happens in a sequence.

Selection

Selection are also known as **Decision control** or **Branching statements**

The selection statements allows a program to test several conditions and execute instructions based on which condition is true.

Some of the Decision control statements are:

* Simple if: a simple if only has one condition to check
* if-else: if-else evaluates the condition and will executed the body of **if**, if the test condition is true, but if the condition is false, then the body of **else** is executed.
* nested if: nested if statements are an if statements inside another if statement.
* if-elif-else: the if-elif-else statement is used to conditionally execute a statement or a block of statements

Repetition

A repetition statement is used to repeat a group(block) of programming instructions.

So in Python we have 2 loops/repetitive statements:

* for loop: a loop is used to iterate over a sequence that is either a list, tuple, dictionary, or a set.
* While loop: the while loops are used to execute a block of statements repeatedly until a given condition is satisfied.

**Day 2**

**Introduction to Functions**

A function is a reusable block of code that performs a specific task or a set of tasks. Functions are essential concept in programming and play a central role in structuring and organizing code.

So to create a function, you must use the **def** keyword, followed by the function name and a pair of parantheses.

**Using Functions**

Function syntax:

def function\_name(parameter <,…>:

#suite

The **function\_name** part is name used to call a function. The **parameter** part shows that zero or more parameters can be given to a function. The **suite** part is where the function is coded.

When the **return** statement is included in the function’s suite, it means a function returns a result, and the return statement can be used to break out of a function.

Three types of functions in Python:

* Ordinary functions: are functions that follow mathematical procedures. They will receive an argument, perform a specific calculation with the argument, and return a result
* Procedure functions: functions that normally do not return a result, they are called to execute a procedure.
* Factory functions: functions that do not take parameters.

Variables that are declared inside a function are called local variables, they are local to the function in which they are declared and cannot be accessed by other functions.

**The Random Module**

The basic **random()** function generates a random floating point number as output.

**Recursive Functions**

Recursion in programming is a method where the solution to a problem depends on the same solution’s result. A recursive function calls itself when it is executed.

Fibonacci recursive function:

This is a Python Function that calculates the Fibonacci sequence using a recursive algorithm. The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, typically starting with 0 and 1