# DSA (Activity)

CS-2023-15 Sher Muhammad

Professor Nazeef Ul Haq

Fall 2024

## 1.Exeucting code using command Prompt:

open a command prompt or terminal, navigate to the directory containing the script, and use the command "python script_name.py" (replace "script name" with the actual filename).

## 2.Code Compilation and Execution in Python:

In Python, code is written in .py files, which are then compiled into bytecode by the Python interpreter when the script is run. This bytecode is a low-level representation of the code that the Python Virtual Machine (PVM) executes. In contrast, for C#, you write code in .cs files, which are compiled into Intermediate Language (IL) by the C# compiler during the build process. At runtime, the Common Language Runtime (CLR) JIT-compiles this IL into native machine code before executing it. The key difference is that Python compiles code to bytecode at runtime and interprets it, whereas C# compiles code to IL at build time and converts it to native code just before execution.

## 3.Data Type of Variable in Python:

In Python, the data type of a variable is inferred automatically based on the value assigned to it. For example, if you assign x = 10, Python infers that x is an integer. To verify the data type of a variable, you use the type () function. For instance, type(x) will return <class 'int'> for an integer variable x. This dynamic typing allows for flexibility in coding but requires the use of type () to explicitly check the variable's data type.

## 4.Mutable and Immutable Data Types:

In Python, mutable data types can be changed after they are created, while immutable data types cannot be changed once they are created. Mutable data types are List, Dictionary and set while Immutable data types are Tuple, String and Integer

## 5.Recursion Pros and Cons:

Recursion is a technique where a function calls itself to solve smaller instances of the same problem.

**Advantages** include simpler code for complex problems and a natural fit for certain algorithms. **Disadvantages** are potential inefficiency and the risk of stack overflowing with deep recursion.

## 6.Recursion Evaluation in Memory:

When a recursive function is called, each call creates a new stack frame with its own local variables and return address. These frames are stacked up until the base case is reached. Once the base case is hit,

Each function returns its result, and the stack frames are removed one by one, unwinding the stack and returning the result.