**DSA Lab 1**

**Activity 1:**

To run Python programs from the **Command Prompt** (CMD) in Spyder. The first step is to open cmd by pressing Windows + R and then Activate anaconda environment by using command “conda activate base”. Now navigate to the folder with the python file by using cd. Run the python file by typing python and filename.py. The output will be displayed.

**Activity 2:**

Python is a high-level, interpreted programming language known for its simplicity and readability. In python code is compiled before execution So there is a difference from other languages. In python, there is no template like in C and C#. Python also give errors on indentation. So, indentation is necessary in python. The bytecode is platform-independent, making it highly portable across operating systems. Python has Quick iterations, suitable for scripting and rapid development

**Activity 3:**

In Python, you don't need to explicitly specify the data type of a variable when you declare it. Python is a dynamically typed language, which means the data type is automatically inferred based on the value assigned to the variable**.** We can find out the datatype of the variable by using type term. Python is flexible than other programming languages.

**Activity 4:**

**Mutable Data Types:**

* **List:** Lists are mutable, meaning you can change, add, or remove elements after the list is created.
* **Dictionary**: Dictionaries are mutable. You can add, remove, or modify key-value pairs.
* **Set**: Sets are also mutable. You can add or remove elements, though sets do not allow duplicate elements.

**Immutable Data Types:**

* **String:** Strings are immutable in Python. Any modification to a string results in a new string being created.
* **Tuple:** Tuples are immutable. Once a tuple is created, its elements cannot be changed, added, or removed.
* **Frozen Set:** A frozen set is an immutable version of a set. You cannot add or remove elements once a frozen set is created.

**Activity 5:**

Recursion is a programming technique where a function calls itself to solve a problem. The function repeatedly invokes itself with smaller or simpler inputs, approaching a base case that terminates the recursive process. It simplifies code. It is natural fit for certain problems. It reduces complexity for division problems but the cons of recursion are performance overhead, harder to debug, higher memory usage. So the summary is that it simplifies code and good for problem with recursive structure and disadvantages are harder to debug and high memory usage.

**Activity 6:**

When a recursive function is called, each call is pushed onto the callstack, creating a new stackframe for that specific call. Each frame holds information such as local variables and function parameters. As the recursion deepens, new frames are added to the stack, waiting for the base case to return a result. Once the base case is reached, the stack begins to unwind, returning results to the previous frames until the initial call is resolved. This process can lead to high memory usage if the recursion depth is large.