# DS3030: Data Analytics Lab
# Assignment 1

Date: Aug 04, 2025

Timing: 2:00 to 5:00 PM Max mark: 15

**Instructions**

- Submit one .ipynb file containing all answers named as [student name]_assignment[number].ipynb
- Write the questions in separate text blocks before the answers.
- Write justifications/comments as required.

1. **Write a function that takes a person's date of birth in the format DD-MM-YYYY as input and calculates their age in the year 2060.**
   Test Input: Enter your date of birth (DD-MM-YYYY): 15-08-2000
   Test Output: You will be 60 years old in 2060. **(1)**

2. **Write a function that takes two string inputs representing numbers, converts them to integers, and prints their sum.**
   Test Input: Enter first number: "25"; Enter second number: "75"
   Test Output: The sum is: 100 **(1)**

3. **Write a Python program that:**
   **- Accepts three user inputs: Name, Age, Favorite programming language**
   **- Stores these as a list of words forming a sentence like:**
   **"My name is Alice, I am 21 years old, and I love programming in Python."**
   Test Input:
   Enter your name: Alice
   Enter your age: 21
   Enter your favorite programming language: Python
   Test Output:
   Using += : My name is Alice, I am 21 years old, and I love programming in Python.
   Using join(): My name is Alice, I am 21 years old, and I love programming in Python.
   Formatted using str.format(): My name is Alice, I am 21 years old, and I love programming in Python. **(1)**

4. **Write a program that removes all punctuation from a sentence using str.translate() and string.punctuation.**
Test Input: Enter a sentence: Hello, world! How are you doing today?
Test Output: Cleaned sentence: Hello world How are you doing today **(1)**

5. **Write a Python program that asks the user to enter a sentence. Then, without modifying the original string, perform and print the following transformations:**
**- Convert the sentence to all uppercase letters**
**- Convert the sentence to all lowercase letters**
**- Capitalize only the first letter**
**- Replace the word 'Python' with 'Java'**
**- Remove leading and trailing spaces**
**Print the original sentence again at the end.**
Test Input: Enter a sentence: Hello, Python World!
Test Output:
Original string: ' Hello, Python World! '
Uppercase: ' HELLO, PYTHON WORLD! '
Lowercase: ' hello, python world! '
Capitalized: ' hello, python world! '
Replaced 'Python' with 'Java': ' Hello, Java World! '
Stripped whitespace: 'Hello, Python World!'
Original after all operations: ' Hello, Python World! ' **(1)**

6. **Create a list (lists) for letters, digits, and special characters; Using indexing, access the required lists and print the string "DS3030: Data Analytics"**
(HINT: You may use Python 'string' module) **(1)**

7. **Print the transpose of a matrix (implemented as a list of lists) without using looping/slicing operator.**
(HINT: Using Python list comprehension) **(1)**

8. **Create a list of strings, say A, and make two copies, say B, C.**
**- Update items in B such that A remains unaffected**
**- Update items in C such that they reflect in A as well.**
(HINT: Using Python object referencing principle) **(1)**

9. **Create a nested dictionary (dictionary of 3 dictionaries) using the dict() constructor such that the first dictionary has values of type strings, the second has integers, and the third has booleans**
**- Print the second dictionary two different ways**
**- Delete the third dictionary**
**- Print the keys of the first dictionary using only keys() function**
**- Add a constant 10 to each value in the second dictionary and print the updated values using items() function** **(1)**

10. **Create a list of 100 numbers with repeating as well as non-repeating elements.**
**- Print a dictionary with frequencies indexed by the respective elements**
**- Print the most frequent element from a dictionary**
(HINT: You may use Python 'random' module, range, and max functions) **(1)**

11. **Write a function to find the largest gap between consecutive prime numbers in a given range [a, b].**
Test Input: Two integers a and b ($1 \leq a < b \leq 1000$)
Test Output: Largest gap between consecutive primes
(Hint: Uses For Loop, Range Function, Boolean Expressions, Nested Conditionals)
Sample Input: 10 30
Sample Output: 4 **(1)**

12. **Write a function to calculate the digital root of a number. Digital root is obtained by repeatedly summing digits until a single digit is obtained.**
Test Input: Integer n ($1 \leq n \leq 10^9$)
Test Output: Digital root of n
(Hint: Uses For Loop, Boolean Expressions, Conditional Execution )
Sample Input: 9875
Sample Output: 2
(Solution : $9+8+7+5 = 29 \Rightarrow 2+9=11 \Rightarrow 2$) **(1)**

13. **Given a lowercase string s of length between 1 and 100, verify if it follows one of four defined vowel-consonant patterns. Print "YES" if the string matches any one of the following patterns, otherwise print "NO"**
Pattern A: Start with a vowel, alternate vowel/consonant strictly
Example: abubu → YES

Rule: Even indices must be vowels, odd indices consonants
Pattern B: Start with a consonant, alternate consonant/vowel strictly
Example: bobobo → YES
Rule: Even indices must be consonants, odd indices vowels
Pattern C: All vowels only (must be length $\geq 2$)
Example: aeiou → YES
Rule: All characters must be vowels
Pattern D: All consonants only (must be length $\geq 2$)
Example: bcdfg → YES
Rule: All characters must be consonants
Test Input: String s ($1 \leq \text{len(s)} \leq 100$, only lowercase letters)
Test Output: "YES" if any pattern is followed, "NO" otherwise
(Hint: Uses For Loop, Range Function, In and Not in Operators, Chained Conditionals)
Sample Input: abubu
Sample Output: YES **(1)**

14. **Count how many Fibonacci numbers exist in a given array. A number is Fibonacci if it appears in the sequence: 0, 1, 1, 2, 3, 5, 8, 13...**
Test Input: Integer n followed by n integers ($1 \leq n \leq 1000$, $0 \leq arr[i] \leq 10^6$)
Test Output: Count of Fibonacci numbers in array
(Hint: Uses For Loop, Boolean Expressions, Conditional Execution, Precedence of Operators)
Sample Input: 6 1 4 8 13 21 25
Sample Output: 4
(Solution: 1, 8, 13, 21 belong to the fibonacci sequence.) **(1)**

15. **Check if a sequence of brackets is valid. Valid means every opening bracket has a corresponding closing bracket in correct order.**
**Types: (), [], {}**
Test Input: String s containing only brackets ($1 \leq \text{len(s)} \leq 1000$)
Test Output: "VALID" if sequence is valid, "INVALID" otherwise
(Hint: Uses For Loop, In and Not in Operators, Conditional Execution, Logical Operators)
Sample Input: {[()]}
Sample Output: VALID **(1)**

## Practice Questions

(Not Evaluated)

1. **Create two lists of numbers, say X, Y.**
   **- Add Y as the last element of X**
   **- Add each element of Y as an element of X**
   (HINT: Difference between append and extend functions of lists)

2. **A number is lucky if the sum of its even-positioned digits equals the sum of odd-positioned digits (1-indexed). Find all lucky numbers in the range [L, R].**
   Test Input: Two integers L and R ($1 \leq L \leq R \leq 10000$)
   Test Output: Count of lucky numbers
   (Hint: Uses For Loop, Range Function, Boolean Expressions, Logical Operators)
   Sample Input: 10 20
   Sample Output: 11
   Solution : THE NUMBER IS 11 as 1=1

3. **Find all peak elements in an array. A peak element is greater than or equal to its neighbors. For corner elements, consider only existing neighbors.**
   Test Input: Integer n followed by n integers ($1 \leq n \leq 1000$)
   Test Output: Indices of all peak elements (0-indexed)
   (Hint: Uses For Loop, Range Function, Logical Operators, Chained Conditionals )
   Sample Input: 5 1 3 2 4 1
   Sample Output: 1 3
   Solution: As $1 < 3 > 2$, $2 < 4 > 1$ so indices of 3 and 4 are 1 and 3.