**Date: 06/09/2025 Timing: 9:30 am to 4:15 pm**

**Long Hour Coding Exam**

**Task:** Design and Implementation of a Python Package signal\_ICT\_StudentName\_EnrollmentNo for Signal Generation and Operations.

**Problem Statement:**

You are required to design a custom Python package named signal\_ICT\_StudentName\_EnrollmentNo that demonstrates fundamental concepts of Signals and Systems. The package must be modular, containing three separate modules:

1. **unitary\_signals.py**

Implement the following functions:

* + - unit\_step(n) – Generates a unit step signal.
    - unit\_impulse(n) – Generates a unit impulse signal.
    - ramp\_signal(n) – Generates a ramp signal.

Each function should return a NumPy array and plot the signal using matplotlib.

1. **trigonometric\_signals.py**

Implement the following functions:

* + - sine\_wave(A, f, phi, t) – Generates a sine wave with amplitude A, frequency f, phase phi, and time vector t.
    - cosine\_wave(A, f, phi, t) – Generates a cosine wave with similar parameters.
    - exponential\_signal(A, a, t) – Generates an exponential signal.

1. **operations.py**

Implement the following signal operations:

* + - time\_shift(signal, k) – Shifts the signal by k units.
    - time\_scale(signal, k) – Scales the time axis of the signal by factor k.
    - signal\_addition(signal1, signal2) – Performs addition of two signals.
    - signal\_multiplication(signal1, signal2) – Performs point-wise multiplication of two signals.

**Main Script (main.py)**

* Import the above modules from the package.
* Demonstrate the following tasks:
  1. Generate and plot a unit step signal and a unit impulse signal of length 20.
  2. Generate a sine wave of amplitude 2, frequency 5 Hz, phase 0, over t = 0 to 1 sec.
  3. Perform time shifting on the sine wave by +5 units and plot both original and shifted signals.
  4. Perform addition of the unit step and ramp signal and plot the result.
  5. Multiply a sine and cosine wave of same frequency and plot the result.

**Expected Deliverables:**

1. Folder structure of the package:
2. signal\_ICT\_StudentName\_EnrollmentNo /
3. \_\_init\_\_.py
4. unitary\_signals.py
5. trigonometric\_signals.py
6. operations.py
7. main.py
8. Well-documented Python code with function definitions and comments.
9. Proper use of NumPy (for signal arrays) and Matplotlib (for plotting).
10. At least 3 plots showing signals and operations as per requirements.

**Student Submission Requirements:**

* The Wheel file (.whl) and source distribution (.tar.gz) inside a dist/ folder.
* A README.md explaining package modules, installation, and usage.
* A screenshot/PDF report showing:
  1. Successful local installation from wheel.
  2. Successful upload to TestPyPI.
  3. Successful installation from TestPyPI.
* GitHub repo link.

Reference Link: <https://www.youtube.com/watch?v=9Ii34WheBOA>

**Evaluation Criteria rubric (20 Marks):**

* Package Design & Modularity (5 Marks) – Proper structure with three modules.
* Correct Implementation of Signals (5 Marks) – Unitary & trigonometric signals.
* Correct Implementation of Operations (5 Marks) – Scaling, shifting, addition, multiplication.
* Main Script Demonstration (3 Marks) – Calling and plotting from all modules.
* Code Quality & Documentation (2 Marks) – Comments, readability, efficiency.

**Student Name: Enrollment No:**

**Division: 3EK1/3EK2/3EK3**