# Lab Report on Signal Operations in Signal Processing

## Theory

Signal processing involves manipulating signals to extract information, enhance features, or analyze behavior. Basic operations like addition, multiplication, scaling, and shifting are fundamental in understanding signal behavior in both time and frequency domains.  
  
1. Signal Addition: Combining two or more signals, often used in overlaying or superimposing information.  
 y(t) = x1(t) + x2(t)  
  
2. Signal Multiplication: Multiplying two signals results in a combined signal with modulated characteristics, often used in amplitude modulation.  
 y(t) = x1(t) \* x2(t)  
  
3. Scaling: Modifying the amplitude or duration of a signal.  
 - Amplitude Scaling: Changes the signal's magnitude by a constant.  
 - Time Scaling: Compresses or expands the signal along the time axis.  
 y(t) = k \* x(at)  
  
4. Shifting: Translating the signal along the time axis.  
 - Time Delay: y(t) = x(t - t0)  
 - Time Advance: y(t) = x(t + t0)  
  
These operations help in signal transformation, modulation, and system analysis.

## Source Code

import numpy as np  
import matplotlib.pyplot as plt  
  
# Parameters  
t = np.arange(-10, 10, 0.01) # Time vector  
  
# Define two signals  
x1 = np.sin(2 \* np.pi \* 1 \* t) # Signal 1: Sine wave  
x2 = np.cos(2 \* np.pi \* 0.5 \* t) # Signal 2: Cosine wave  
  
# Signal Addition  
y\_add = x1 + x2  
  
# Signal Multiplication  
y\_mult = x1 \* x2  
  
# Amplitude Scaling  
k = 2 # Scaling factor  
y\_scaled = k \* x1  
  
# Time Shifting  
shift = 2 # Shift value (in seconds)  
y\_shifted = np.sin(2 \* np.pi \* 1 \* (t - shift)) # Delayed signal  
  
# Plot Results  
plt.figure(figsize=(10, 12))  
  
# Original Signals  
plt.subplot(5, 1, 1)  
plt.plot(t, x1, label='x1: Sine Wave', color='b')  
plt.plot(t, x2, label='x2: Cosine Wave', color='r')  
plt.title('Original Signals')  
plt.legend()  
plt.grid()  
  
# Signal Addition  
plt.subplot(5, 1, 2)  
plt.plot(t, y\_add, color='m')  
plt.title('Signal Addition')  
plt.grid()  
  
# Signal Multiplication  
plt.subplot(5, 1, 3)  
plt.plot(t, y\_mult, color='k')  
plt.title('Signal Multiplication')  
plt.grid()  
  
# Amplitude Scaling  
plt.subplot(5, 1, 4)  
plt.plot(t, y\_scaled, color='g')  
plt.title('Amplitude Scaling')  
plt.grid()  
  
# Time Shifting  
plt.subplot(5, 1, 5)  
plt.plot(t, y\_shifted, color='c')  
plt.title('Time Shifting')  
plt.grid()  
  
# Display the plots  
plt.tight\_layout()  
plt.show()

## Output

The Python code generates plots to visualize each operation:  
1. Original Signals: Displays the sine and cosine waves.  
2. Signal Addition: Shows the resultant waveform of superimposed signals.  
3. Signal Multiplication: Displays the product of the two signals.  
4. Amplitude Scaling: Presents the sine wave scaled in amplitude.  
5. Time Shifting: Depicts the delayed sine wave.