|  |
| --- |
| **Scripting**  CW\_KRSIT\_H  Lab Seven |

# Notes

Remember-

No curly brackets, indent your code

Functions are declared with **def**

Add a colon after a function

Use # to add comments

Comment your code, explain what each function does, what is its inputs, what is its outputs

Any errors you encountered when creating these programs and subsequently fixed paste them as an error log at the bottom of your code and write a description of why they occurred and how you fixed them.

Call all your methods from the main method and test with an assortment of values

# The task

## Overview

The objective of this lab is to apply the sysadmin concepts we discussed in the last lecture using Python scripting. You will create a script that performs an incremental backup of files, compresses them into a ZIP archive, and generates a detailed log file that includes system resource usage and file operations.

This exercise will help you:

* Practice file handling operations.
* Work with system resource monitoring and logging.
* Compare working with Python vs working with other scripting languages (such as bash and batch that we used last year for analogous tasks).

# Instructions:

1. **Copy Files by Extension**:
   * Write a script that:
     + Takes two command-line arguments:
       1. The folder to search for files.
       2. The file extension to copy (e.g., .txt).
     + Copies all files with the specified extension from the given folder to a new directory named backup.
2. **Incremental Backup**:
   * As the script starts, it should scan the backup folder to check which files already exist.
   * For each file in the source folder:
     + **If the file does not exist in the backup folder**: Copy it and log it as a **new file**.
     + **If the file exists but has been modified** (based on file modification time): Overwrite it in the backup folder and log it as a **modified file**.
     + **If the file exists and has not been modified**, log that it was **not copied**.
3. **Create a Compressed Archive**:
   * After copying the files, compress the backup folder into a .zip file.
   * The zip file should have a file name that clearly indicates the date and time in which the zip was created.
4. **Generate a Detailed Log**:
   * Write a log file in the backup directory that includes:
     + The date-time in which the scripts start (You can add an entry such as “Backup Script started”
     + CPU usage percentage.
     + Memory usage percentage.
     + Disk space usage (total, used, free).
     + A timestamp (date and time) for each log entry.
     + The operating system (Windows/Linux/MacOS)
     + Information about the file copying process:
       1. For each file processed, log:
          - File name.
          - Whether it was **new**, **modified**, or **not copied**.
   * Append new entries to the log file every time the script is run.