Test: get_file_info() function's ability to return array with file info while ignoring .exe, .py, .pyd, and .dll files

Input:

- Run fileSynchronizer.py in directory with:
  - testDir (directory)
  - file
  - file1.txt
  - fileSynchronizer.py (itself)
  - hi.pdf
  - testexe.exe
  - yaya.dll
  - yo.pyd

Expected output:

{'name': file, 'mtime': [mtime]}

{'name': file1.txt, 'mtime': [mtime]}

{'name': hi.pdf, 'mtime': [mtime]}

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA3\test1>filesynchronizer.py
{'name': 'file', 'mtime': 1553702191}
{'name': 'file1.txt', 'mtime': 1553702191}
{'name': 'hi.pdf', 'mtime': 1553702191}
```

*Print statements added to print elements of file_info array

Result: Pass

Test: get_next_available_port() function

Input:

- Run fileSynchronizer.py with added print statements and an extra call to get_next_available_port(8000) at end:

```
#get free port
synchronizer_port = get_next_available_port(8000)
print(synchronizer_port) # REMOVE ME
synchronizer_thread = FileSynchronizer(tracker_ip,tracker_port,synchronizer_port)
print(get_next_available_port(8000)) # REMOVE ME
synchronizer_thread.start()
```

Expected output: Prints next two available ports

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA3\test1>filesynchronizer.py 127.0.0.1 8080
8000
8001
```

Result: Pass

Test: Establish TCP connection

Input:

- Run tracker.exe with command line arguments: 127.0.0.1 8080
- Run fileSynchronizer.py with command line arguments: 127.0.0.1 8080

Expected output: “Client connected”

Output:

```
C:\Users\drumm\Files\Coding\GitLab\salayog\PA3\test1>filesynchronizer.py 127.0.0.1 8080
Waiting for connections on port 8000
('connect to:127.0.0.1', 8080)
```

Result: Pass

Test: Copy new files to peers

Input:

- Folder with fileSynchronizer.py and file1.txt (content: “Hello world”)
- Folder with fileSynchronizer.py and file2.txt (content: “Goodbye world”)
- Run tracker.exe with command line arguments: 127.0.0.1 8080
- Run both copies of fileSynchronizer.py with command line arguments: 127.0.0.1 8080

Expected output: Both folders possess file1.txt and file2.txt (with their respective contents)

Output: Both folders possess file1.txt and file2.txt (with their respective contents)

Result: Pass

Test: Update files held by peers to most recent versions

Input:

- Folder with fileSynchronizer.py, file1.txt (content: “Hello world”), and file2.txt (empty text file)
    - file1.txt is most recently modified version
- Folder with fileSynchronizer.py, file2.txt (content: “Goodbye world”), and file1.txt (empty text file)
    - file2.txt is most recently modified version

- Run tracker.exe with command line arguments: 127.0.0.1 8080
- Run both copies of fileSynchronizer.py with command line arguments: 127.0.0.1 8080

Expected output: Both folders possess file1.txt (with content: “Hello world”) and file2.txt (with content: “Goodbye world”)

Output: Both folders possess file1.txt and file2.txt (with their respective contents)

Result: Pass