PROGRAMMING IN PYTHON

HARMAN SETHI
CSE-A
0461770222

1. import os

```
def count_word_frequency(input_file, output_file):
    try:
        with open(input_file, 'r', encoding='utf-8')
        text = file.read()

    word_list = text.split()
    word_freq = {}

    for word in word_list:
        word = word.lower().strip(',.!?;:*')
        word_freq[word] = word_freq.get(word, 0) + 1

    with open(output_file, 'w', encoding='utf-8') as file:
        for word, count in sorted(word_freq.items(), key = lambda x: x[1],
        reverse = True):
            file.write(f"{word}: {count}\n")

    print("Word frequency count saved successfully")

except FileNotFoundError:
    print("Error: The input file does not exist.")

except Exception as e:
    print(f"An error occured: {e}")

count_word_frequency("sample.txt", "word_count.txt")
```

2. import os

```
def list_files_by_extension(folder_path, extension):
    try:
        if not os.path.exists(folder_path):
            print("Error: Folder does not exist")
            return
        files = [f for f in os.listdir(folder_path) if f.endswith(extension)]
```

```
        print(f"Files with .{extension} extension:")
        for file in files:
            print(file)

    except Exception as e:
        print(f"Error: {e}")
list_files_by_extension("test_folder", ".txt")
```

3. import shutil

```
def zip_folder(folder_path, zip_name):
    try:
        shutil.make_archive(zip_name, 'zip', folder_path)
        print(f"Folder '{folder_path}' compressed successfully")
    except Exception as e:
        print(f"Error: {e}")

def extract_zip(zip_path, extract_to):
    try:
        shutil.unpack_archive(zip_path, extract_to, 'zip')
        print(f"Zip file '{zip_path}' extracted successfully")
    except Exception as e:
        print(f"Error: {e}")
zip_folder("test_folder", "compressed_folder")
extract_zip("compressed_folder.zip", "extracted_folder")
```

4. import requests

```
def download_image(url, save_path):
    try:
        response = requests.get(url, stream=True)
        response.raise_for_status()
        with open(save_path, 'wb') as file:
            for chunk in response.iter_content(1024):
                file.write(chunk)
        print(f"Image downloaded successfully: {save_path}")
    except requests.exceptions.RequestException as e:
        print(f"Error: {e}")
```

```
image_url = "https://example.com/sample.jpg"
download_image(image_url, "downloaded_image.jpg")
```

5.
```
import re
def extract_emails(file_path):
    try:
        with open(file_path, 'r', encoding="utf-8") as file:
            content = file.read()

        emails = re.findall(r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}', content)
        print("Extracted Email Addresses:")
        for email in emails:
            print(email)

    except FileNotFoundError:
        print("Error: File not found")
    except Exception as e:
        print(f"Error: {e}")

extract_emails("sample_text.txt")
```

6.
```
import requests
import os

def download_files(url_list, save_folder):
    if not os.path.exists(save_folder):
        os.makedirs(save_folder)

    for i, url in enumerate(url_list):
        try:
            response = requests.get(url, stream=True)
            response.raise_for_status()

            file_name = os.path.join(save_folder, f"file_{i+1}.jpg")

            with open(file_name, 'wb') as file:
                for chunk in response.iter_content(1024):
                    file.write(chunk)
            print(f"Downloaded: {file_name}")
        except requests.exceptions.RequestException as e:
            print(f"Error downloading {url}: {e}")
```

```
urls = ["https://example.com/image1.jpg", "https://example.com/image2.jpg"]
download_files(urls, "downloaded_images")
```