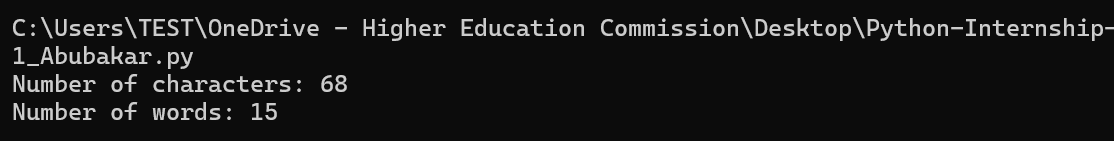
**Python Assignment Week 03 – Abu Bakar**

**TASK 01**

**CODE**

|  |
| --- |
| def count\_characters\_and\_words(file\_path: str) -> None:      try:          with open(file\_path, 'r') as file:              content = file.read()              num\_characters = len(content)              num\_words = len(content.split())              print(f"Number of characters: {num\_characters}")              print(f"Number of words: {num\_words}")      except FileNotFoundError:          print("The file was not found.")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  count\_characters\_and\_words('task1.txt') |

**RESULT**

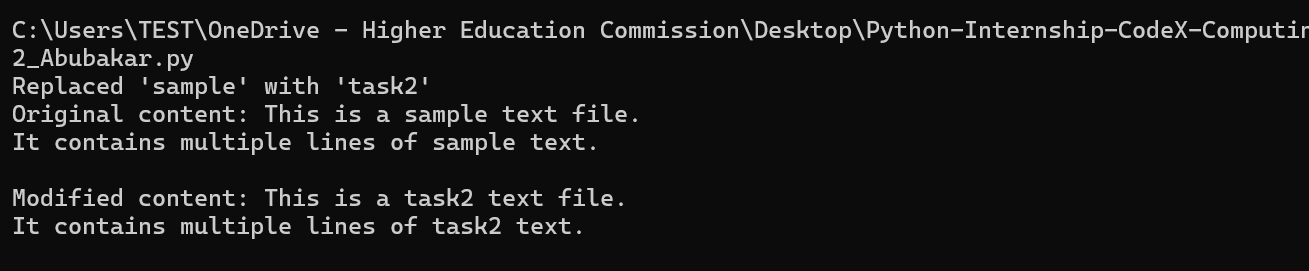
****

**TASK 02**

**CODE**

|  |
| --- |
| def search\_and\_replace(file\_path: str, search\_word: str, replace\_word: str) -> None:      try:          with open(file\_path, 'r') as file:              content = file.read()          modified\_content = content.replace(search\_word, replace\_word)          with open(file\_path, 'w') as file:              file.write(modified\_content)          print(f"Replaced '{search\_word}' with '{replace\_word}'")          print("Original content:", content)          print("Modified content:", modified\_content)      except FileNotFoundError:          print("The file was not found.")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  search\_and\_replace('task2.txt', 'sample', 'task2') |

**RESULT**

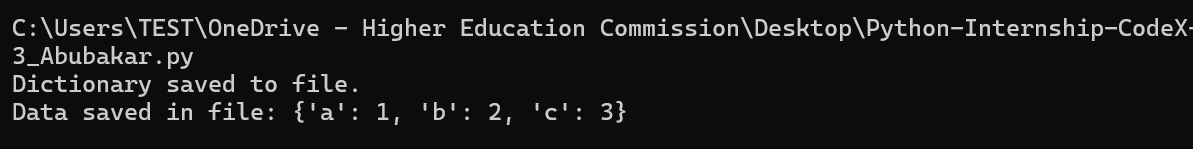
****

**TASK 03**

**CODE**

|  |
| --- |
| def lists\_to\_dict(list1: list, list2: list, file\_path: str) -> None:      try:          if len(list1) != len(list2):              raise ValueError("Lists must have the same number of elements.")          dictionary = {list1[i]: list2[i] for i in range(len(list1))}          with open(file\_path, 'w') as file:              file.write(str(dictionary))          print("Dictionary saved to file.")                  print(f"Data saved in file: {dictionary}")      except ValueError as ve:          print(ve)      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  lists\_to\_dict(['a', 'b', 'c'], [1, 2, 3], 'dictionary.txt') |

**RESULT**

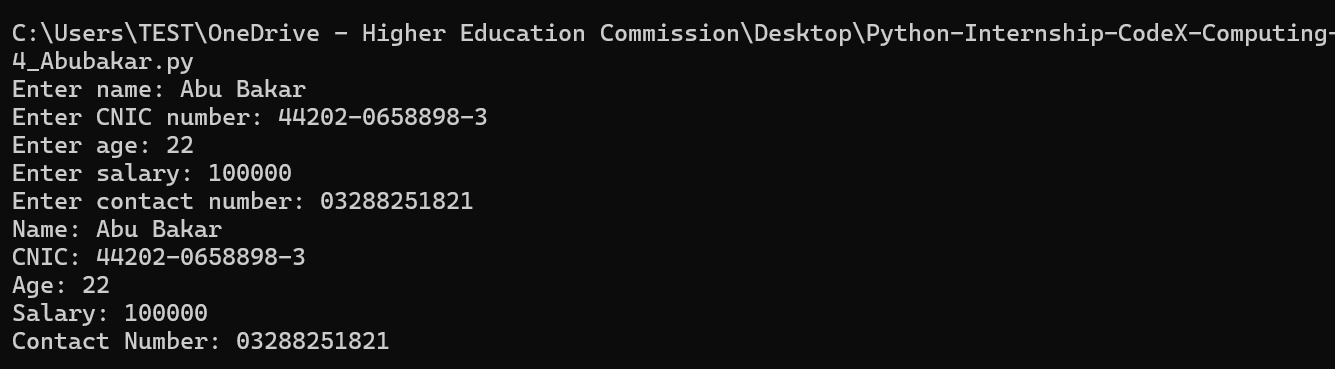
****

**TASK 04**

**CODE**

|  |
| --- |
| def save\_employee\_biodata(file\_path: str) -> None:      try:          name = input("Enter name: ")          cnic = input("Enter CNIC number: ")          age = input("Enter age: ")          salary = input("Enter salary: ")          with open(file\_path, 'w') as file:              file.write(f"Name: {name}\nCNIC: {cnic}\nAge: {age}\nSalary: {salary}\n")          contact\_number = input("Enter contact number: ")          with open(file\_path, 'a') as file:              file.write(f"Contact Number: {contact\_number}\n")          with open(file\_path, 'r') as file:              print(file.read())      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  save\_employee\_biodata('task4\_employee\_biodata.txt') |

**RESULT**

****

**TASK 05**

**CODE**

|  |
| --- |
| import json  def save\_and\_find\_max\_age(dictionary: dict, file\_path: str) -> None:      try:          with open(file\_path, 'w') as file:              json.dump(dictionary, file)          with open(file\_path, 'r') as file:              data = json.load(file)          max\_age = max(data.values())          max\_age\_names = [name for name, age in data.items() if age == max\_age]          print(f"Person(s) with the maximum age ({max\_age}): {', '.join(max\_age\_names)}")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  dictionary = {'Ali': 23, 'Saad': 24, 'Salman': 15, 'Shams': 25, 'Sadiq': 46, 'Hammad': 23}  save\_and\_find\_max\_age(dictionary, 'task5\_ages.json') |

**RESULT**

**A black background with white text

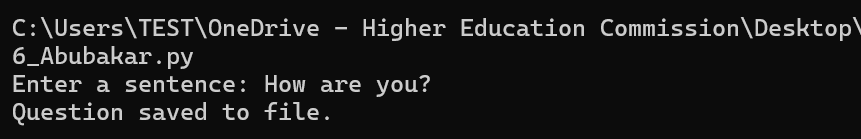
AI-generated content may be incorrect.**

**TASK 06**

**CODE**

|  |
| --- |
| def write\_questions\_to\_file(file\_path: str) -> None:      try:          sentence = input("Enter a sentence: ")          if sentence.endswith('?'):              with open(file\_path, 'a') as file:                  file.write(sentence + '\n')              print("Question saved to file.")          else:              print("The sentence is not a question.")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  write\_questions\_to\_file('task6\_questions.txt') |

**RESULT**

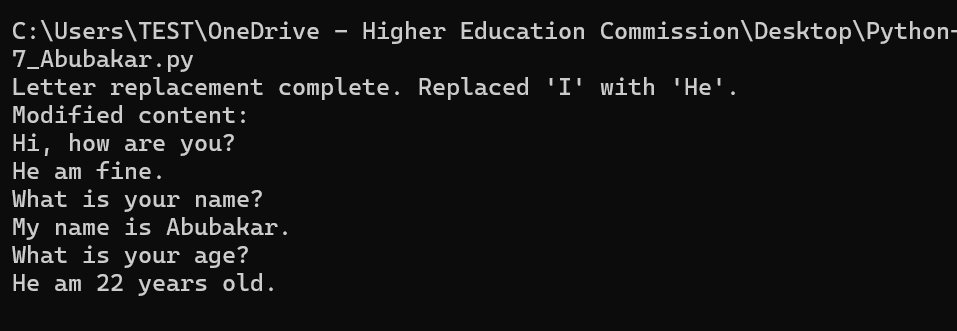
****

**TASK 07**

**CODE**

|  |
| --- |
| def replace\_letter\_in\_file(file\_path: str, old\_letter: str, new\_letter: str) -> None:      try:          with open(file\_path, 'r') as file:              content = file.read()          modified\_content = content.replace(old\_letter, new\_letter)          with open(file\_path, 'w') as file:              file.write(modified\_content)          print(f"Letter replacement complete. Replaced '{old\_letter}' with '{new\_letter}'.")          print(f"Modified content:\n{modified\_content}")      except FileNotFoundError:          print("The file was not found.")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  replace\_letter\_in\_file('task7\_replacement\_needed.txt', 'I', 'He') |

**RESULT**

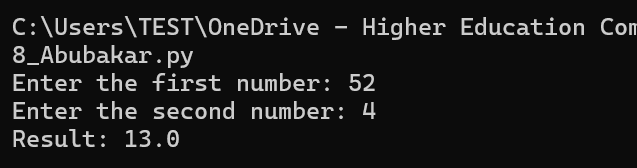
****

**TASK 08**

**CODE**

|  |
| --- |
| def divide\_numbers() -> None:      try:          num1 = int(input("Enter the first number: "))          num2 = int(input("Enter the second number: "))          result = num1 / num2          print(f"Result: {result}")      except ZeroDivisionError:          print("Error: Cannot divide by zero.")      except ValueError:          print("Error: Invalid input. Please enter numbers only.")      except Exception as e:          print(f"An error occurred: {e}")  # Example usage  divide\_numbers() |

**RESULT**

****