Assignment 7

|  |  |  |
| --- | --- | --- |
| **7** | | |
| **Aim:** Write a program that creates a new file, writes some content to it, closes the file, and then reopens it to read and display the content.  Write a program that reads a list of numbers, inserts odd numbers into a file named odd\_numbers.txt, and even numbers into a file named even\_numbers.txt.  Write a program that creates a new file, writes some content to it, closes the file, and then reopens it to read and display the content.  Write a program that reads a text file and prints any 5 words from the file.  Write a program that generates a triangle pattern of 5 rows and saves the pattern into a file named triangle.txt. | | |
| **Code:**  def create\_write\_read\_file():  *"""Create a new file, write content to it, close it, and then reopen to read and display content."""*  *# Create and write to file*  with open("sample.txt", "w") as file:  file.write("Hello, this is some sample content.\n")  file.write("Python file handling is fun and useful!\n")  file.write("This file was created as part of Assignment 7.")    print("File created and content written successfully.")    *# Reopen and read the file*  try:  with open("sample.txt", "r") as file:  content = file.read()  print("\nContent of the file:")  print(content)  except FileNotFoundError:  print("Error: File not found.")  def separate\_odd\_even\_numbers():  *"""Read a list of numbers and insert odd numbers into odd\_numbers.txt and even numbers into even\_numbers.txt."""*  numbers = input("Enter numbers separated by spaces: ").split()    *# Convert inputs to integers*  numbers = [int(num) for num in numbers]    *# Open files for writing*  with open("odd\_numbers.txt", "w") as odd\_file, open("even\_numbers.txt", "w") as even\_file:  for num in numbers:  if num % 2 == 0:  even\_file.write(str(num) + "\n")  else:  odd\_file.write(str(num) + "\n")    print("Numbers have been separated into odd\_numbers.txt and even\_numbers.txt")    *# Read and display the contents of both files*  print("\nContents of odd\_numbers.txt:")  with open("odd\_numbers.txt", "r") as odd\_file:  print(odd\_file.read())    print("Contents of even\_numbers.txt:")  with open("even\_numbers.txt", "r") as even\_file:  print(even\_file.read())  def read\_five\_words():  *"""Read a text file and print any 5 words from the file."""*  filename = input("Enter the filename to read from: ")  try:  with open(filename, "r") as file:  content = file.read()  words = content.split()    print(f"5 words from {filename}:")  for i in range(min(5, len(words))):  print(f"{i+1}: {words[i]}")    if len(words) < 5:  print(f"Note: The file only contains {len(words)} words.")    except FileNotFoundError:  print(f"Error: File '{filename}' not found.")  def generate\_triangle():  *"""Generate a triangle pattern of 5 rows and save to triangle.txt."""*  with open("triangle.txt", "w") as file:  for i in range(1, 6):  pattern = "\* " \* i  file.write(pattern + "\n")    print("Triangle pattern has been saved to triangle.txt")    *# Read and display the content*  print("\nContents of triangle.txt:")  with open("triangle.txt", "r") as file:  print(file.read())  def main():  while True:  print("\n" + "="\*50)  print("File Handling Menu:")  print("1. Create, write, close, reopen and read a file")  print("2. Separate odd and even numbers into files")  print("3. Read and print 5 words from a text file")  print("4. Generate triangle pattern and save to file")  print("5. Exit")  print("="\*50)    choice = input("\nEnter your choice (1-5): ")    if choice == "1":  create\_write\_read\_file()  elif choice == "2":  separate\_odd\_even\_numbers()  elif choice == "3":  read\_five\_words()  elif choice == "4":  generate\_triangle()  elif choice == "5":  print("Exiting program. Goodbye!")  break  else:  print("Invalid choice. Please try again.")  if \_\_name\_\_ == "\_\_main\_\_":  print("File Handling Operations - Assignment 7")  main()  **Output Screenshot:** | | |
| **Conclusion/Summary:**  The file handling operations implemented in this assignment demonstrate several important concepts in Python file I/O:  Basic File Operations: The program showcases the fundamental operations of creating, writing to, closing, and reading from files using the with statement, which ensures proper resource management.  Error Handling: Try-except blocks are used to gracefully handle potential errors such as FileNotFoundError, making the program more robust and user-friendly.  File Processing Logic: The assignment demonstrates practical applications of file handling, such as categorizing data (odd/even numbers) and storing formatted output (triangle pattern).  User Interaction: The menu-driven interface provides a clean way for users to interact with the program and choose which file operation to perform.  String Manipulation: The code demonstrates various string operations like splitting input strings, formatting output, and creating patterns.  These file handling techniques are essential skills for data processing, configuration management, logging, and many other real-world programming tasks. The program structure also follows good coding practices with well-documented functions and a modular design that separates concerns.  From a practical standpoint, this assignment provides a foundation for more complex file operations that might be encountered in data analysis, system administration, or application development. | | |
| **Student Signature & Date** | **Marks:** | **Evaluator Signature & Date** |