ASSIGNMENT – 5 [Strings, I/O Operations, and File Management]

Subject: CSW2 (CSE 2141)

Name: Arpit Kumar Mohanty

Registration Number: 2341013237

Section: 23412G1

Branch: CSE

Q1. Write a Java program that illustrates the difference between using string literals and the new keyword for creating String objects. Your program should demonstrate the memory usage implications and how string comparison behaves differently in each case.

Solution:

```java
public class Q1_StringComparison {
    Run | Debug
    public static void main(String[] args) {
        String str1 = "Arpit", str2 = "Arpit";
        String str3 = new String(original:"Arpit"), str4 = new String(original:"Arpit");

        System.out.println(x:"Comparing memory locations:");
        System.out.println("str1 == str2 : " + (str1 == str2));
        System.out.println("str1 == str3 : " + (str1 == str3));
        System.out.println("str3 == str4 : " + (str3 == str4));

        System.out.println(x:"\nComparing actual string values:");
        System.out.println("str1.equals(str2) : " + str1.equals(str2));
        System.out.println("str1.equals(str3) : " + str1.equals(str3));
        System.out.println("str3.equals(str4) : " + str3.equals(str4));

        String str5 = str3.intern();
        System.out.println(x:"\nAfter interning str3:");
        System.out.println("str1 == str5 : " + (str1 == str5));

        Runtime runtime = Runtime.getRuntime();
        long beforeMemory = runtime.freeMemory();
        System.out.println("\nMemory before: " + beforeMemory + " bytes");

        for (int i = 0; i < 10000; i++) new String(original:"MemoryTest");

        long afterMemory = runtime.freeMemory();
        System.out.println("Memory after: " + afterMemory + " bytes");
        System.out.println("Memory difference: " + (beforeMemory - afterMemory) + " bytes");
    }
}
```

Output:

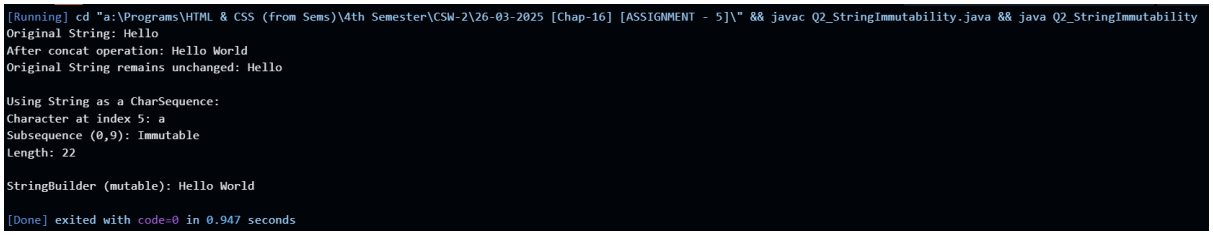
```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03-2025 [Chap-16] [ASSIGNMENT - 5]\" && javac Q1_StringComparison.java && java Q1_StringComparison
Comparing memory locations:
str1 == str2 : true
str1 == str3 : false
str3 == str4 : false

Comparing actual string values:
str1.equals(str2) : true
str1.equals(str3) : true
str3.equals(str4) : true

After interning str3:
str1 == str5 : true

Memory before: 261221040 bytes
Memory after: 260759648 bytes
Memory difference: 461392 bytes

[Done] exited with code=0 in 0.97 seconds
```

Q2. Write a Java program that demonstrates the immutability of the String class and how it implements the CharSequence interface. Your program should illustrate the behaviours that highlight String immutability and its usage as a CharSequence.

Solution:

```java
public class Q2_StringImmutability {
    Run | Debug
    public static void main(String[] args) {
        // Demonstrating immutability
        String str = "Hello";
        System.out.println("Original String: " + str);

        // Attempt to modify the string
        String modifiedStr = str.concat(str:" World");
        System.out.println("After concat operation: " + modifiedStr);
        System.out.println("Original String remains unchanged: " + str);

        // String as a CharSequence
        CharSequence charSeq = "Immutable CharSequence";
        System.out.println(x:"\nUsing String as a CharSequence:");
        System.out.println("Character at index 5: " + charSeq.charAt(index:5));
        System.out.println("Subsequence (0,9): " + charSeq.subSequence(start:0, end:9));
        System.out.println("Length: " + charSeq.length());

        // Demonstrating behavior when using StringBuilder (mutable alternative)
        StringBuilder sb = new StringBuilder(str:"Hello");
        sb.append(str:" World");
        System.out.println("\nStringBuilder (mutable): " + sb);
    }
}
```

Output:



Q3. Write a Java program that uses StringBuffer to construct a simple text editor which can perform the following operations: a. Append a given string to the existing text. b. Insert a given string at a specified index within the existing text. c. Delete a portion of text between two specified indices. d. Reverse the entire text. e. Replace a portion of the text between two specified indices with a given string. Your program should display a menu with options to perform each of the above operations. After each operation, print the current state of the text. Also, display the current capacity and length of the StringBuffer after each operation to showcase its dynamic nature.

Solution:

```java
import java.util.*;
public class Q3_SimpleTextEditor {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StringBuffer text = new StringBuffer(str:"Arpit Kumar");

        while (true) {
            System.out.println("\nCurrent Text: " + text);
            System.out.println("Length: " + text.length() + ", Capacity: " + text.capacity());
            System.out.println(x:"1. Append  2. Insert  3. Delete  4. Reverse  5. Replace  6. Exit");
            System.out.print(s:"Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine(); // Consume newline
            if (choice == 6) break;

            switch (choice) {
                case 1 -> {
                    System.out.print(s:"Enter text to append: ");
                    text.append(sc.nextLine());
                }
                case 2 -> {
                    System.out.print(s:"Enter index: ");
                    int index = sc.nextInt();
                    sc.nextLine();
                    System.out.print(s:"Enter text to insert: ");
                    text.insert(index, sc.nextLine());
                }
                case 3 -> {
                    System.out.print(s:"Enter start and end index: ");
                    text.delete(sc.nextInt(), sc.nextInt());
                }
                case 4 -> text.reverse();
                case 5 -> {
                    System.out.print(s:"Enter start and end index: ");
                    int start = sc.nextInt(), end = sc.nextInt();
                    sc.nextLine();
                    System.out.print(s:"Enter replacement text: ");
                    text.replace(start, end, sc.nextLine());
                }
                default -> System.out.println(x:"Invalid option!");
            }
        }
        sc.close();
    }
}
```

Output:

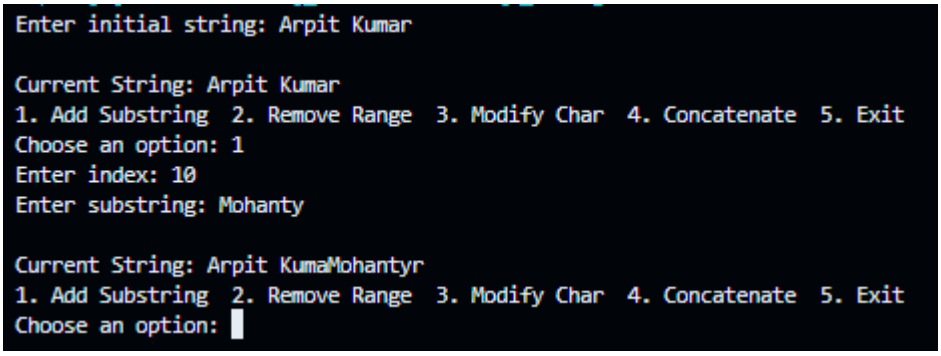
```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\Rec

Current Text: Arpit Kumar
Length: 11, Capacity: 27
1. Append  2. Insert  3. Delete  4. Reverse  5. Replace  6. Exit
Choose an option: 1
Enter text to append:  Mohanty

Current Text: Arpit Kumar Mohanty
Length: 19, Capacity: 27
1. Append  2. Insert  3. Delete  4. Reverse  5. Replace  6. Exit
Choose an option: 4

Current Text: ytnahoM ramuK tiprA
Length: 19, Capacity: 27
1. Append  2. Insert  3. Delete  4. Reverse  5. Replace  6. Exit
Choose an option:
```

Q4. Create a Java program that uses StringBuilder to perform a series of text manipulations on a user-provided string. The program should allow users to: a. Add a substring at a specified position. b. Remove a range of characters from the string. c. Modify a character at a specified index. d. Concatenate another string at the end. e. Display the current string after each operation. The program should repeatedly prompt the user to choose an operation until they decide to exit. After each operation, it should display the modified string, demonstrating the mutable nature of StringBuilder.

Solution:

```java
import java.util.*;
public class Q4_StringBuilderEditor {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter initial string: ");
        StringBuilder sb = new StringBuilder(sc.nextLine());

        while (true) {
            System.out.println("\nCurrent String: " + sb);
            System.out.println(x:"1. Add Substring  2. Remove Range  3. Modify Char  4. Concatenate  5. Exit");
            System.out.print(s:"Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine(); // Consume newline

            if (choice == 5) break;

            switch (choice) {
                case 1 -> {
                    System.out.print(s:"Enter index: ");
                    int index = sc.nextInt();
                    sc.nextLine();
                    System.out.print(s:"Enter substring: ");
                    sb.insert(index, sc.nextLine());
                }
                case 2 -> {
                    System.out.print(s:"Enter start and end index: ");
                    sb.delete(sc.nextInt(), sc.nextInt());
                }
                case 3 -> {
                    System.out.print(s:"Enter index: ");
                    int index = sc.nextInt();
                    sc.nextLine();
                    System.out.print(s:"Enter new character: ");
                    sb.setCharAt(index, sc.nextLine().charAt(index:0));
                }
                case 4 -> {
                    System.out.print(s:"Enter string to concatenate: ");
                    sb.append(sc.nextLine());
                }
                default -> System.out.println(x:"Invalid option!");
            }
        }
        sc.close();
    }
}
```

Output:

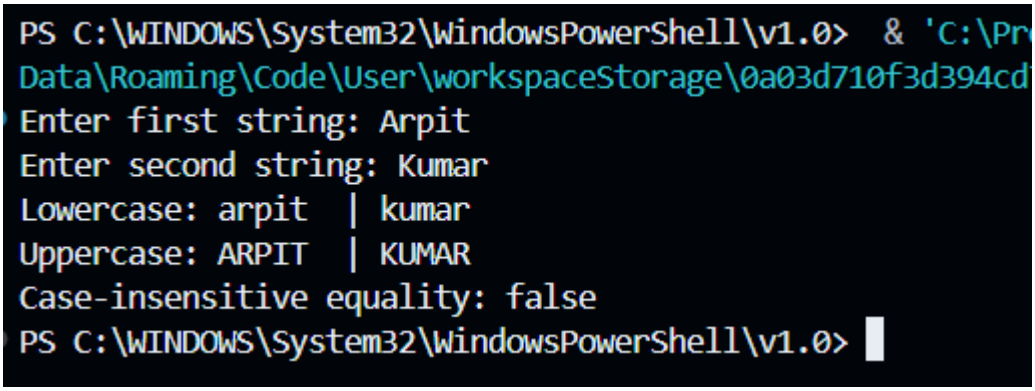


Q5. Case Conversion and Comparison: Prompt the user to input two strings. Convert both strings to lowercase and uppercase. Compare the converted strings to check caseinsensitive equality. Display the converted strings and the result of the comparison.

Solution:

```java
import java.util.Scanner;
public class Q5_CaseConversionComparison {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter first string: ");
        String str1 = sc.nextLine();
        System.out.print(s:"Enter second string: ");
        String str2 = sc.nextLine();

        System.out.println("Lowercase: " + str1.toLowerCase() + " | " + str2.toLowerCase());
        System.out.println("Uppercase: " + str1.toUpperCase() + " | " + str2.toUpperCase());
        System.out.println("Case-insensitive equality: " + str1.equalsIgnoreCase(str2));
        sc.close();
    }
}
```

Output:



Q6. Character Array and Search: Ask for a string from the user. Convert the string to a character array. Prompt the user to enter a character to

search in the string. Find the first and last occurrences of the character. Display the character array and the positions found (if any).

Solution:

```java
import java.util.*;
public class Q6_CharArraySearch {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter a string: ");
        char[] charArray = sc.nextLine().toCharArray();

        System.out.print(s:"Enter character to search: ");
        char ch = sc.next().charAt(index:0);

        String str = new String(charArray);
        int first = str.indexOf(ch), last = str.lastIndexOf(ch);

        System.out.println("Character Array: " + java.util.Arrays.toString(charArray));
        System.out.println(first == -1 ? "Character not found" : "First: " + first + ", Last: " + last);

        sc.close();
    }
}
```

Output:

```
Enter a string: Arpit kumar m
Enter character to search: m
Character Array: [A, r, p, i, t,  , k, u, m, a, r,  , m]
First: 8, Last: 12
PS A:\>
```

Q7. Word Replacement in Sentences: Request a sentence and two words from the user: one to search for and one to replace it with. Find the first occurrence of the search word in the sentence. Replace the word using substring operations and concatenation. Display the original and the modified sentences.

Solution :

```java
import java.util.*;
public class Q7_WordReplacement {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter a sentence: ");
        String sentence = sc.nextLine();

        System.out.print(s:"Enter word to search: ");
        String search = sc.next();
        System.out.print(s:"Enter replacement word: ");
        String replace = sc.next();

        int index = sentence.indexOf(search);
        String modified = (index == -1) ? sentence : sentence.substring(beginIndex:0, index) + replace + sentence.substring(index + search.length());

        System.out.println("Original: " + sentence);
        System.out.println("Modified: " + modified);

        sc.close();
    }
}
```

Output :

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-16] [ASSIGNM
Enter a sentence: Arpit Kumar
Enter word to search: mar
Enter replacement word: moh
Original: Arpit Kumar
Modified: Arpit Kumoh
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q8. Interactive String Explorer: Prompt the user for a string. Display a menu with options to perform various operations: convert to lowercase/uppercase, search for a character/index, or concatenate with another string. Based on user selection, perform the appropriate string operation and show the result.

Solution:

```java
import java.util.*;
public class Q8_StringExplorer {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter a string: ");
        String str = sc.nextLine();

        while (true) {
            System.out.println(x:"\n1. To Lower  2. To Upper  3. Search Char  4. Search Index  5. Concatenate  6. Exit");
            System.out.print(s:"Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine(); // Consume newline

            if (choice == 6) break;

            switch (choice) {
                case 1 -> System.out.println("Lowercase: " + str.toLowerCase());
                case 2 -> System.out.println("Uppercase: " + str.toUpperCase());
                case 3 -> {
                    System.out.print(s:"Enter character: ");
                    char ch = sc.next().charAt(index:0);
                    System.out.println("First Occurrence: " + str.indexOf(ch));
                }
                case 4 -> {
                    System.out.print(s:"Enter substring: ");
                    String sub = sc.nextLine();
                    System.out.println("First Occurrence: " + str.indexOf(sub));
                }
                case 5 -> {
                    System.out.print(s:"Enter string to concatenate: ");
                    str += sc.nextLine();
                    System.out.println("Updated String: " + str);
                }
                default -> System.out.println(x:"Invalid option!");
            }
        }
        sc.close();
    }
}
```

Output :

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openj
rs\Arpit\AppData\Roaming\Code\User\workspaceStorage\0a03d710f3d394cd798c21468000e31f\redi
r'
Enter a string: Arpit Kumar

1. To Lower  2. To Upper  3. Search Char  4. Search Index  5. Concatenate  6. Exit
Choose an option: 2
Uppercase: ARPIT KUMAR

1. To Lower  2. To Upper  3. Search Char  4. Search Index  5. Concatenate  6. Exit
Choose an option: 1
Lowercase: arpit kumar

1. To Lower  2. To Upper  3. Search Char  4. Search Index  5. Concatenate  6. Exit
Choose an option:
```

Q9. Create and Write to a File: Write a Java program that prompts the user for a diary entry, then creates a file named "diary.txt" and writes the current date followed by the user's entry into this file. Ensure the

program checks if the file already exists and informs the user, to avoid overwriting any previous content.

Solution :

```java
import java.io.*;
import java.time.LocalDate;
import java.util.Scanner;
public class Q9_DiaryWriter {
    Run | Debug
    public static void main(String[] args) throws IOException {
        File file = new File(pathname:"A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]\\diary.txt");
        if (file.exists()) {
            System.out.println(x:"File already exists. Append new entries instead.");
            return;
        }

        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter your diary entry: ");
        String entry = sc.nextLine();

        try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {
            writer.write(LocalDate.now() + "\n" + entry);
            System.out.println(x:"Diary entry saved to diary.txt");
        }
        sc.close();
    }
}
```

Output :

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Fil
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\2
Enter your diary entry: This side Arpit, CEO of Diary Pr
Diary entry saved to diary.txt
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q10. Read from a File: Write a Java application that opens the "diary.txt" file created in the previous question and displays its content on the console. The program should handle cases where the file does not exist by displaying an appropriate error message.

Solution:

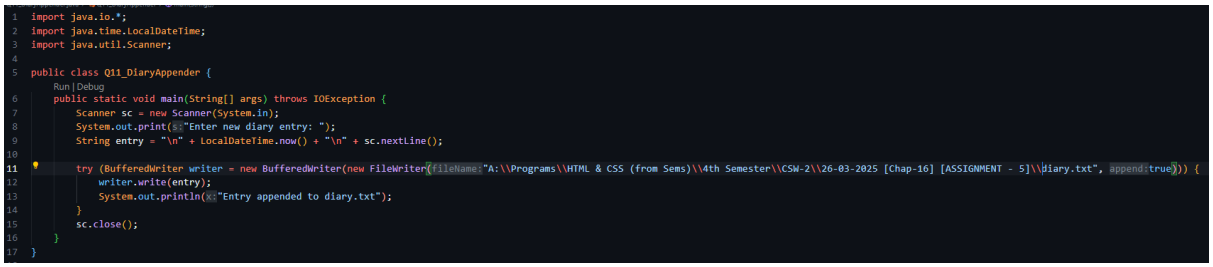
```java
import java.io.*;
public class Q10_DiaryReader {
    Run | Debug
    public static void main(String[] args) {
        File file = new File(pathname:"A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]\\diary.txt");
        if (!file.exists()) {
            System.out.println(x:"Error: File not found.");
            return;
        }

        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
            System.out.println("Diary Content:\n" + reader.lines().reduce(identity:"", (a, b) -> a + "\n" + b));
        } catch (IOException e) {
            System.out.println(x:"Error reading the file.");
        }
    }
}
```

Output :

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21.0.6.0.7-1\bin\java.exe' '-XX:+S
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-16] [ASSIGNMENT - 5]_1e83c963\bin' 'Q10_Diary
Diary Content:

2025-03-30
This side Arpit, CEO of Diary Pr
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

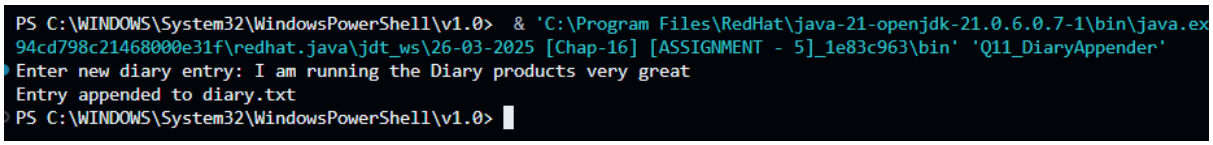
Q11. Append Content to an Existing File: Write a Java program that adds a new diary entry to the "diary.txt" file without overwriting its existing content. The program should ask the user for the new entry and append it to the file along with a timestamp.

Solution:



```java
import java.io.*;
import java.time.LocalDateTime;
import java.util.Scanner;

public class Q11_DiaryAppender {
    Run|Debug
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter new diary entry: ");
        String entry = "\n" + LocalDateTime.now() + "\n" + sc.nextLine();

        try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName:"A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]\\diary.txt", append:true))) {
            writer.write(entry);
            System.out.println(x:"Entry appended to diary.txt");
        }
        sc.close();
    }
}
```

Output:



```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21.0.6.0.7-1\bin\java.ex
94cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-16] [ASSIGNMENT - 5]_1e83c963\bin' 'Q11_DiaryAppender'
Enter new diary entry: I am running the Diary products very great
Entry appended to diary.txt
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q12. List Files and Directories: Write a program in Java that asks the user for a directory path and then lists all files and subdirectories in that directory. If the directory does not exist, the program should inform the user.

Solution :

```java
import java.io.File;
import java.util.Scanner;

public class Q12_ListFilesDirectories {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter directory path: ");
        File dir = new File(sc.nextLine());

        if (!dir.exists() || !dir.isDirectory()) {
            System.out.println(x:"Invalid directory.");
            sc.close();
            return;
        }

        String[] files = dir.list();
        System.out.println(x:"Contents of directory:");
        for (String file : files) System.out.println(file);

        sc.close();
    }
}
```

Output :

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21.0.6.0.7-1\bin\java.e
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-16] [ASSIGNMENT - 5]_1e83c963\bin
Enter directory path: A:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\18-03-2025 ASSIGNMENT-4 (Chap-15)
Contents of directory:
2341013237 Arpit Kumar Mohanty ASSIGNMENT - 4.pdf
AnonymousObject.class
Ans.4 (2).txt
ASSIGNMENT - 4.docx
ASSIGNMENT - 4.pdf
Assignment-4.txt
CSW2_Assignment4_Ch15.pdf
DataObject.class
NullifiedReference.class
Q1.class
Q1.java
Q2.class
Q2.java
Q3.class
Q3.java
Q4.class
Q4.java
Q5.class
Q5.java
Q6$MemoryIntensiveObject.class
Q6.class
Q6_MemoryIntensiveApp.class
Q6_MemoryIntensiveApp.java
Q6_Solution.jpg
Q6_Solution_Output.jpg
Q7.class
Q7.java
ReassigningReference.class
Student.class
UnreachableObject.class
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q13. Delete a Specific File: Write a Java program where the user can enter the name of a file to be deleted from the system. The program should check if the file exists and delete it, providing a confirmation message upon successful deletion or an error message if the file does not exist.

Solution:

```java
import java.io.File;
import java.util.Scanner;

public class Q13_DeleteFile {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter file name to delete: ");
        File file = new File("A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]"+sc.nextLine());

        System.out.println(file.exists() ? (file.delete() ? "File deleted." : "Deletion failed.") : "File not found.");
        sc.close();
    }
}
```

Output:

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-2
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-
Enter file name to delete: no_file.txt
File not found.
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q14. Copy File Content: Write a Java program that copies the content from one file (source) to another (destination). The program should prompt the user for both source and destination file paths and perform the copy operation, ensuring that it doesn't overwrite an existing file without user confirmation

Solution:

```java
import java.io.*;
import java.util.Scanner;

public class Q14_CopyFile {
    Run | Debug
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter source file path: ");
        File src = new File("A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]"+sc.nextLine());
        System.out.print(s:"Enter destination file path: ");
        File dest = new File("A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]"+sc.nextLine());

        if (!src.exists()) {
            System.out.println(x:"Source file not found.");
            sc.close();
            return;
        }
        if (dest.exists()) {
            System.out.print(s:"Destination file exists. Overwrite? (yes/no): ");
            if (!sc.next().equalsIgnoreCase(anotherString:"yes")) {
                sc.close();
                return;
            }
        }

        try (FileInputStream in = new FileInputStream(src); FileOutputStream out = new FileOutputStream(dest)) {
            out.write(in.readAllBytes());
            System.out.println(x:"File copied successfully.");
        }
        sc.close();
    }
}
```

Output:

```
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>  & 'C:\Program Files\RedHat\java-21-openjdk-21.0.6.0.7-1\bin\java.exe' '-XX:+ShowC
paceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap-16] [ASSIGNMENT - 5]_1e83c963\bin' 'Q14_CopyFile
Enter source file path: hello.txt
Enter destination file path: there.txt
Source file not found.
PS C:\WINDOWS\System32\WindowsPowerShell\v1.0>
```

Q15. Rename a File: Develop a Java application that renames a specified file. The program should request the current file name and the new file name from the user, renaming the file accordingly and confirming the action upon completion.

Solution:

```java
import java.io.*;
import java.util.Scanner;

public class Q15_RenameFile {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String directoryPath = "A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]"; // Set your desired directory path

        System.out.print(s:"Enter current file name: ");
        File oldFile = new File(directoryPath + sc.nextLine());

        if (!oldFile.exists()) {
            System.out.println(x:"File not found in the specified directory.");
            sc.close();
            return;
        }

        System.out.print(s:"Enter new file name: ");
        File newFile = new File(directoryPath + sc.nextLine());

        System.out.println(oldFile.renameTo(newFile) ? "File renamed successfully." : "Rename failed.");
        sc.close();
    }
}
```

Output:

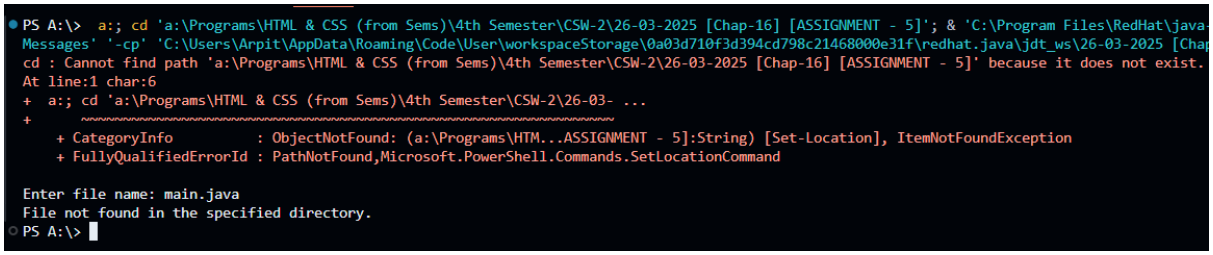
```
PS A:\>  a:; cd 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03-2025 [Chap-16] [ASSIGNMENT - 5]'; & 'C:\Program Files
Messages' '-cp' 'C:\Users\Arpit\AppData\Roaming\Code\User\workspaceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-
cd : Cannot find path 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03-2025 [Chap-16] [ASSIGNMENT - 5]' because it doe
At line:1 char:6
+  a:; cd 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03- ...
+       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (a:\Programs\HTM...ASSIGNMENT - 5]:String) [Set-Location], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

Enter current file name: diary.txt
File not found in the specified directory.
PS A:\>
```

Q16. Display File Metadata: Create a Java program that displays metadata of a specified file. The user should be able to input the file name, and the program should output the file size, last modified date, and other available attributes.

Solution:

```java
import java.io.*;
import java.util.Scanner;
import java.nio.file.*;
import java.nio.file.attribute.*;

public class Q16_FileMetadata {
    Run | Debug
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        String directoryPath = "A:\\Programs\\HTML & CSS (from Sems)\\4th Semester\\CSW-2\\26-03-2025 [Chap-16] [ASSIGNMENT - 5]"; // Set your desired directory path

        System.out.print(s:"Enter file name: ");
        File file = new File(directoryPath + sc.nextLine());

        if (!file.exists()) {
            System.out.println(x:"File not found in the specified directory.");
            sc.close();
            return;
        }

        BasicFileAttributes attr = Files.readAttributes(file.toPath(), type:BasicFileAttributes.class);
        System.out.println("Size: " + file.length() + " bytes");
        System.out.println("Last Modified: " + Files.getLastModifiedTime(file.toPath()));
        System.out.println("Created: " + attr.creationTime());
        System.out.println("Is Directory: " + attr.isDirectory());

        sc.close();
    }
}
```

Output:


PS A:\> a:; cd 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03-2025 [Chap-16] [ASSIGNMENT - 5]'; & 'C:\Program Files\RedHat\java
Messages' '-cp' 'C:\Users\Arpit\AppData\Roaming\Code\User\workspaceStorage\0a03d710f3d394cd798c21468000e31f\redhat.java\jdt_ws\26-03-2025 [Chap
cd : Cannot find path 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03-2025 [Chap-16] [ASSIGNMENT - 5]' because it does not exist.
At line:1 char:6
+ a:; cd 'a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\26-03- ...
+ ~~~~~~
+ CategoryInfo : ObjectNotFound: (a:\Programs\HTM...ASSIGNMENT - 5]:String) [Set-Location], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand
Enter file name: main.java
File not found in the specified directory.
PS A:\>