

Task 1: String Operations

Write a method that takes two strings, concatenates them, reverses the result, and then extracts the middle substring of the given length. Ensure your method handles edge cases, such as an empty string or a substring length larger than the concatenated string.

Code-:

```
package com.wipro.assignment;
public class StringOperations {
    public static String
extractMiddleSubstring(String str1,
String str2, int length) {
    // Concatenate the two strings
    String concatenated =
str1.concat(str2);

    // Reverse the concatenated
string
    StringBuilder reversed = new
StringBuilder(concatenated).reverse();

    // Get the length of the reversed
string
    int reversedLength =
reversed.length();
```

```
        // If the length of the reversed
string is smaller than the specified
length, return null
        if (reversedLength < length) {
            return null;
        }

        // Calculate the starting index
of the middle substring
        int startIndex = (reversedLength
- length) / 2;

        // Extract the middle substring
        return
reversed.substring(startIndex, startIndex
+ length);
    }

    public static void main(String[]
args) {
        String str1 = "Hello";
        String str2 = "World";
        int length = 5;

        String result =
extractMiddleSubstring(str1, str2,
length);
    }
}
```

```

        if (result != null) {
            System.out.println("Middle
substring of length " + length + ": " +
result);
        } else {
            System.out.println("Substring
length is larger than the concatenated
string.");
        }
    }
}

```

Output: -

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project structure with packages like 'com.wipro.assignment' and 'com.wipro'. The main editor window shows the source code for 'StringOperations.java'. The code includes package declarations, class declarations, and static methods for string operations. The console window on the right shows the output of the program: 'Middle substring of length 5: roWol'.

```

1 package com.wipro.assignment;
2 public class StringOperations
3 {
4     // Concatenate the two strings
5     String concatenated = "roWol";
6
7     // Reverse the concatenated string
8     String reverse = new StringBuilder(concatenated).reverse().toString();
9
10    // Get the length of the reversed string
11    int reversedLength = reverse.length();
12
13    // If the length of the reversed string is less than the original string
14    if (reversedLength < concatenated.length())
15    {
16        return null;
17    }
18
19    // Calculate the start index of the middle substring
20    int startIndex = (concatenated.length() - reversedLength) / 2;

```

Console Output: Middle substring of length 5: roWol