Task 7: Basic Exception Handling

Write a program that attempts to divide by zero, catches the ArithmeticException, and provides a custom error message.

```
public class task7 {

public class Main {

    public static void main(String[] args) {

int dividend = 5;          int divisor = 0;

try {

        int result = dividend / divisor;

        System.out.println("Result: " + result);

      } catch (ArithmeticException e) {

        System.out.println("Error: " + e.getMessage());

      }

    }

  }

}
```

Output:

```
"Error: Cannot divide by zero!"
```

Task 2: List interface

Implement a method that takes a List as an argument and removes every second element from the list, then prints the resulting list.

```
import java.util.ArrayList; import

java.util.List; public class

listinterface {     public static
```

```
void main(String[] args) {
List<Integer> numbers = new
ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
numbers.add(4);
numbers.add(5);
numbers.add(6);
numbers.add(7);
numbers.add(8);
numbers.add(9);
numbers.add(10);

        List<Integer> result = removeEverySecondElement(numbers);
        System.out.println("Original List: " + numbers);
        System.out.println("Result List: " + result);
    }

    public static List<Integer> removeEverySecondElement(List<Integer> list) {
List<Integer> result = new ArrayList<>();
        for (int i = 0; i < list.size(); i++) {
         if (i % 2 == 0) {
result.add(list.get(i));
         }
}
      return result;
   }
}
```

Output:

```
Original List: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Result List: [1, 3, 5, 7, 9]
```

Task 3: Set interface

Write a program that reads words from a String variable into a Set and prints out the number of unique words, demonstrating the unique property of sets.

```
import java.util.HashSet; import

java.util.Set;


public class setinterface {    public static

void main(String[] args) {

      String input = "This is a sample sentence for testing unique words in Java";

Set<String> uniqueWords = new HashSet<>();
```

```
        // Add words from the input string to the set
for (String word : input.split("\\s+")) {
uniqueWords.add(word);

        }


        // Print the number of unique words
        System.out.println("Number of unique words: " + uniqueWords.size());


        // Print the unique words
        System.out.println("Unique words:");
for (String word : uniqueWords) {
            System.out.println(word);
        }
    }
}
```

Output:

```
Number of unique words: 11 Unique
words:
sentence
a
Java
in testing
unique
```

```
for
words This
is
sample
```

Task 4: Map interface

Create a Java class that uses a Map to store the frequency of each word that appears in a given string. import java.util.HashMap; import java.util.Map;

```
public class mapinterface {

    public static Map<String, Integer> countWordFrequency(String text) {
        Map<String, Integer> frequencyMap = new HashMap<>();

        // Split the text into words
        String[] words = text.split("\\s+");

        // Count the frequency of each word
for (String word : words) {             //
Remove any punctuation marks
            word = word.replaceAll("[^a-zA-Z]", "").toLowerCase();

            // If the word is not empty after removing punctuation
if (!word.isEmpty()) {
```

```
                frequencyMap.put(word, frequencyMap.getOrDefault(word, 0) + 1);
            }
        }

        return frequencyMap;
    }

    public static void main(String[] args) {
        String text = "This is a sample text, with some words. This text will be
used to demonstrate the word frequency counter.";
        Map<String, Integer> frequencyMap = countWordFrequency(text);

        // Print the word frequencies         for (Map.Entry<String, Integer>
entry : frequencyMap.entrySet()) {
System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
} Output:
a: 1
some: 1
be: 1
will: 1 this:
2
words: 1
```

```
is: 1

used: 1 counter:

1 sample: 1

frequency: 1

the: 1

with: 1

text: 2 to:

1

demonstrate: 1 word:

1
```