Java Coding Challenges - Chapter 2

Challenge 1: Method Creation

Create methods for common operations:

```
public class MathUtils {
    // TODO: Create a method called 'square' that takes an int and returns
its square

    // TODO: Create a method called 'isEven' that takes an int and returns
true if even

    // TODO: Create a method called 'getMax' that takes two ints and returns
the larger one

    public static void main(String[] args) {
        // Test your methods here
        System.out.println(square(5));        // Should print 25
        System.out.println(isEven(4));        // Should print true
        System.out.println(getMax(10, 7));    // Should print 10
    }
}
```

Challenge 2: Array Operations

Create methods that work with arrays:

```
public class ArrayUtils {
    // TODO: Create a method 'printArray' that prints all elements in an int
array

    // TODO: Create a method 'findSum' that returns the sum of all elements

    // TODO: Create a method 'findAverage' that returns the average as a
double

    public static void main(String[] args) {
        int[] numbers = {5, 10, 15, 20, 25};

        printArray(numbers);                  // Should print each number
        System.out.println(findSum(numbers));     // Should print 75
        System.out.println(findAverage(numbers)); // Should print 15.0
    }
}
```

Challenge 3: String Array Processing

Work with arrays of strings:

```
public class StringProcessor {
    // TODO: Create a method 'countWords' that counts how many strings are in
the array

    // TODO: Create a method 'findLongest' that returns the longest string

    // TODO: Create a method 'hasWord' that checks if a specific word exists
in the array

    public static void main(String[] args) {
        String[] words = {"java", "programming", "fun", "coding"};

        System.out.println(countWords(words));           // Should print 4
        System.out.println(findLongest(words));          // Should print
"programming"
        System.out.println(hasWord(words, "java"));      // Should print true
    }
}
```

Challenge 4: Grade Calculator

Create a complete grade management system:

```
public class GradeCalculator {
    // TODO: Create a method 'calculateGrade' that takes an array of scores
    // and returns the average as a double

    // TODO: Create a method 'getLetterGrade' that takes a double average
    // and returns a char: A (90+), B (80+), C (70+), D (60+), F (<60)

    // TODO: Create a method 'passedClass' that takes an average
    // and returns true if >= 60

    public static void main(String[] args) {
        int[] scores = {85, 92, 78, 96, 89};

        double average = calculateGrade(scores);
        char letter = getLetterGrade(average);
        boolean passed = passedClass(average);

        System.out.println("Average: " + average);
        System.out.println("Letter Grade: " + letter);
        System.out.println("Passed: " + passed);
    }
}
```

Solutions

Challenge 1 Solutions:

```
public static int square(int num) {
    return num * num;
}
```

```
public static boolean isEven(int num) {
    return num % 2 == 0;
}

public static int getMax(int a, int b) {
    if (a > b) {
        return a;
    } else {
        return b;
    }
    // Alternative: return (a > b) ? a : b;
}
```

Challenge 2 Solutions:

```
public static void printArray(int[] arr) {
    for (int num : arr) {
        System.out.println(num);
    }
}

public static int findSum(int[] arr) {
    int sum = 0;
    for (int num : arr) {
        sum += num;
    }
    return sum;
}

public static double findAverage(int[] arr) {
    return (double) findSum(arr) / arr.length;
}
```

Challenge 3 Solutions:

```
public static int countWords(String[] words) {
    return words.length;
}

public static String findLongest(String[] words) {
    String longest = words[0];
    for (String word : words) {
        if (word.length() > longest.length()) {
            longest = word;
        }
    }
    return longest;
}

public static boolean hasWord(String[] words, String target) {
    for (String word : words) {
        if (word.equals(target)) {
            return true;
        }
```

```
    }
    return false;
}
```

Challenge 4 Solutions:

```
public static double calculateGrade(int[] scores) {
    int sum = 0;
    for (int score : scores) {
        sum += score;
    }
    return (double) sum / scores.length;
}

public static char getLetterGrade(double average) {
    if (average >= 90) return 'A';
    else if (average >= 80) return 'B';
    else if (average >= 70) return 'C';
    else if (average >= 60) return 'D';
    else return 'F';
}

public static boolean passedClass(double average) {
    return average >= 60;
}
```