Java Assignment-4

Topic:Arrays and String functions

Objective:

These assignments will help beginners to intermediates to practice and understand various concepts related to arrays and string functions in Java.

Instructions:

1.Complete the code and save it in the form of codeshare link
2.The document needs to be converted into pdf including the codeshare links containing the code
3.The screenshot of the output for each program should be pasted below the codeshare link.
4.Upload your work in respective directories under your name.
5.Time limit:60 mins

Questions:

1.Create an array of integers and use a for loop to print out each element of the array.

Code:

```
public class ArrayExample {
    public static void main(String[] args) {
        // Create an array of integers
        int[] numbers = {1, 2, 3, 4, 5};

        // Use a for loop to print out each element of the array
        for (int i = 0; i < numbers.length; i++) {
            System.out.println("Element at index " + i + ": " + numbers[i]);
        }
    }
}
```

Output:
Element at index 0: 1
Element at index 1: 2
Element at index 2: 3
Element at index 3: 4
Element at index 4: 5

2.Create an array of strings and use a for-each loop to print out each element of the array.

Code:

```
public class StringArrayExample {
    public static void main(String[] args) {
        // Create an array of strings
        String[] names = {"Alice", "Bob", "Charlie", "David", "Eve"};

        // Use a for-each loop to print out each element of the array
        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

Output:
Alice
Bob
Charlie
David
Eve

3.Create an array of doubles and use a while loop to print out each element of the array.

Code:

```
public class DoubleArrayExample {
    public static void main(String[] args) {
        // Create an array of doubles
        double[] values = {1.1, 2.2, 3.3, 4.4, 5.5};

        // Use a while loop to print out each element of the array
        int index = 0;
        while (index < values.length) {
            System.out.println("Element at index " + index + ": " +
values[index]);
            index++;
        }
    }
}
```

Output:

```
Element at index 0: 1.1
Element at index 1: 2.2
Element at index 2: 3.3
Element at index 3: 4.4
Element at index 4: 5.5
```

4.Create an array of characters and use a do-while loop to print out each element of the array.

Code:

```
public class CharArrayExample {
    public static void main(String[] args) {
        // Create an array of characters
        char[] characters = {'A', 'B', 'C', 'D', 'E'};
```

```
        // Use a do-while loop to print out each element of the array
        int index = 0;
        do {
            System.out.println("Element at index " + index + ": " +
characters[index]);
            index++;
        } while (index < characters.length);
    }
}
```

Output:
Element at index 0: A
Element at index 1: B
Element at index 2: C
Element at index 3: D
Element at index 4: E

5.Create an array of integers and use the Arrays class method sort() to sort the array in ascending order.

Code:

```
import java.util.Arrays;

public class ArraySortingExample {
    public static void main(String[] args) {
        // Create an array of integers
        int[] numbers = {5, 2, 8, 1, 3, 9, 4};

        // Use the Arrays.sort() method to sort the array in ascending order
        Arrays.sort(numbers);

        // Print the sorted array
        System.out.println("Sorted array in ascending order:");
        for (int number : numbers) {
            System.out.print(number + " ");
        }
```

```
    }
}
```
Output:
Sorted array in ascending order:
1 2 3 4 5 8 9

6.Create an array of strings and use the Arrays class method binarySearch() to find the index of a specific string in the array.

Code:
```
import java.util.Arrays;

public class BinarySearchExample {
    public static void main(String[] args) {
        // Create an array of strings
        String[] names = {"Alice", "Bob", "Charlie", "David",
"Eve"};

        // Sort the array (required for binary search)
        Arrays.sort(names);

        // String to search for
        String target = "Charlie";

        // Use the Arrays.binarySearch() method to find the index
of the target string
        int index = Arrays.binarySearch(names, target);

        if (index >= 0) {
            System.out.println(target + " found at index " + index);
        } else {
            System.out.println(target + " not found in the array");
        }
    }
}
```

Output:
Charlie found at index 2

7.Create a string and use the String class method split() to split the string into an array of substrings.

Code:

```
public class StringSplitExample {
    public static void main(String[] args) {
        // Create a string
        String text = "Hello,World,Java,Programming";

        // Use the String.split() method to split the string into an array of substrings
        String[] parts = text.split(",");

        // Print the resulting array of substrings
        for (String part : parts) {
            System.out.println(part);
        }
    }
}
```

Output:
Hello
World
Java
Programming

8.Create a string and use the String class method replace() to replace a specific substring in the string with a new substring.

Code:

```
public class StringReplaceExample {
    public static void main(String[] args) {
        // Create a string
        String text = "Hello, world!";

        // Use the String.replace() method to replace a specific substring
        String newText = text.replace("world", "Java");

        // Print the updated string
        System.out.println(newText);
    }
}
```

Output:

```
Hello, Java!
```

9.Create a string and use the String class method substring() to extract a portion of the string.

Code:

```
public class SubstringExample {
    public static void main(String[] args) {
        // Create a string
        String text = "This is an example string.";

        // Use the String.substring() method to extract a portion of the string
        String extracted = text.substring(5, 13);

        // Print the extracted portion
        System.out.println(extracted);
```

```
    }
}
```

Output:
is an ex

10.Create a string and use the String class method length() to find the length of the string.

Code:

```
public class StringLengthExample {
    public static void main(String[] args) {
        // Create a string
        String text = "This is a sample string.";

        // Use the String.length() method to find the length of the string
        int length = text.length();

        // Print the length of the string
        System.out.println("The length of the string is: " + length);
    }
}
```

Output:
The length of the string is: 24

.