

# Brainstorm

Given a random input String from the user, how to count

- a) the number of digits in the input String
- b) the number of uppercase letters in the input String
- c) the number of lowercase letters in the input String
- d) the number of spaces in the input String

**Q: How many times  
do we need to repeat  
the process?**

1. Use **charAt()** to get the char from the input String
2. Use if statements to check the char
  - a) **Character.isDigit()**
  - b) **Character.isUpperCase()**
  - c) **Character.isLowerCase()**
  - d) Compare with ' '

# while Loop

zyBook Chap 5.1, 5.2

# while Loops

- **While** the **<condition>** is **true**, executes the **<controlled stmt(s)>**
  - Can be considered as an if statement that's **repeatedly** executed until the **<condition>** is false
- Hence, the **<controlled stmt(s)>** in a while loop can be executed **ZERO** or **MANY** times.

```
while (<condition>) {  
    <controlled stmt(s)>;  
}  
  
<statement(s)>;
```

```

public class MultiplesOfFour {
    public static void main (String[] args) {

        int val = 4;

        /*
         * Check if the condition is true
         * If yes, execute the controlled statements
         *     1) print out val
         *     2) increment val by 4
         *     3) check if the condition is still true
         * If no, skip the while loop
         */
        while (val <= 20){
            System.out.println(val);
            val += 4;
        }

        System.out.println("Done.");
    }
}

```

val = 4

**val <= 20 ?**

True

print out 4

val += 4 // 8

**val <= 20 ?**

True

print out 8

val += 4 // 12

**val <= 20 ?**

True

print out 12

val += 4 // 16

**val <= 20 ?**

True

print out 16

val += 4 // 20

**val <= 20 ?**

True

print out 20

val += 4 // 24

**val <= 20 ?**

False

print out "Done."

Q: What's the exact output?

```
public class WhileExample {  
    public static void main (String[] args) {  
        int x = 1;  
        while (x < 11) {  
            System.out.print(x + " ");  
        }  
        System.out.println("!");  
    }  
}
```

**Infinite Loop**

Output:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 .....

x = 1

x < 11 ?

Yes

print out 1

x < 11 ?

Yes

print out 1

x < 11 ?

Yes

print out 1

...

Q: What's the exact output?

```
public class WhileExample {  
    public static void main (String[] args) {  
        int x = 1;  
        while (x < 11) {  
            System.out.print(x + " ");  
            x += 4;  
        }  
        System.out.println("!");  
    }  
}
```

Output:

**1 5 9 !**

x = 1

x < 11 ?

Yes

print out 1

x += 4 // 5

x < 11 ?

Yes

print out 5

x += 4 // 9

x < 11 ?

Yes

print out 9

x += 4 // 13

x < 11 ?

No

print out "!"

**Q:** For each point in the code, choose  
(**A**)lways true, (**N**)ever true, or (**S**)ometimes true

	$y < x$	$y == 0$	$count > 0$
Point A	<b>S</b>	<b>S</b>	<b>N</b>
Point B	<b>A</b>	<b>S</b>	<b>S</b>
Point C	<b>A</b>	<b>A</b>	<b>A</b>
Point D	<b>S</b>	<b>S</b>	<b>S</b>
Point E	<b>N</b>	<b>S</b>	<b>S</b>

```
public static int mystery (Scanner input, int x) {  
    int y = input.nextInt();  
    int count = 0;  
    // Point A  
    while (y < x) {  
        // Point B  
        if (y == 0) {  
            count++;  
            // Point C  
        }  
        y = input.nextInt();  
        // Point D  
    }  
    // Point E  
    return count;  
}
```

**Q:** What's wrong with the following code? How to fix it?

```

/*
 * Write a method called printNum that prints each number
 * from 1 to a given maximum, with each number separated by a comma.
 * Sample output for printNum(5): 1, 2, 3, 4, 5
 */
public static void printNum(int max) {
    int val = 1;
    while (val <= max){
        System.out.print(val + ", ");
        ++val;
    }
}

```

**Fencepost Problem**

**Incorrect Output:**  
 1, 2, 3, 4, 5,



# Three ways to address Fencepost Problem

```
public static void printNumbers(int max) {  
    int val = 1;  
    while (val < max) {  
        System.out.print(val + ", ");  
        ++val;  
    }  
    System.out.print(val);  
}
```

```
public static void printNumbers(int max) {  
    int val = 1;  
    System.out.print(val);  
    while (val < max) {  
        ++val;  
        System.out.print(", " + val);  
    }  
}
```

```
public static void printNum(int max) {  
    int val = 1;  
    while (val <= max) {  
        System.out.print(val);  
        if (val < max) { // Checks the boundary  
            System.out.print(", ");  
        }  
        ++val;  
    }  
}
```

# Coding Practice

```
$ javac CountChar.java  
$ java CountChar  
Enter a String: Spring23 - CS 1101  
The input String "Spring23 - CS 1101" contains 6 digits,  
3 uppercase letters, 5 lowercase letters, and 3 spaces.
```

Write a program that

- prompts the user for an input String (one or multiple tokens)
- counts and prints
  - a) the number of digits in the input String
  - b) the number of uppercase letters in the input String
  - c) the number of lowercase letters in the input String
  - d) the number of spaces in the input String

# Sample Solution

```
import java.util.Scanner;

public class CountChar {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a String: ");
        String str = input.nextLine(); // Use nextLine() since the input could be multiple tokens

        int numDigit = 0, numUpper = 0, numLower = 0, numSpace = 0;
        int index = 0; // Set index to 0 as the String index starts at 0

        // Make sure the index is within the valid range of the input String
        while(index < str.length()){

            char temp = str.charAt(index);

            if(Character.isDigit(temp)) {
                numDigit++;
            } else if (Character.isUpperCase(temp)) {
                numUpper++;
            } else if (Character.isLowerCase(temp)) {
                numLower++;
            } else if (temp == ' '){
                numSpace++;
            }
            index++;
        }

        System.out.println("The input String \"" + str + "\" contains " +
            numDigit + " digits, " + numUpper + " uppercase letters, " +
            numLower + " lowercase letters, and " + numSpace + " spaces.");
    }
}
```