

1. The `isDigit`, `isLetter`, and `isLetterOrDigit` methods are members of this class.
  1. `String`
  2. `Char`
  - 3. `Character`**
  4. `StringBuilder`
2. This method converts a character to uppercase.
  1. `makeUpperCase`
  - 2. `toUpperCase`**
  3. `isUpperCase`
  4. `upperCase`
3. This `String` class method returns true if the calling string's length is 0 or it contains only whitespace characters.
  1. `isEmpty`
  - 2. `isBlank`**
  3. `isNull`
  4. `isVoid`
4. This `String` class method returns true if the calling string's length is 0, but returns false if the calling string contains only whitespace characters.
  - 1. `isEmpty`**
  2. `isBlank`
  3. `isNull`
  4. `isVoid`

5. This String class method returns true if the calling String object contains a specified substring.

1. matches
2. isOwnerOf
3. substring
4. contains

6. The startsWith, endsWith, and regionMatches methods are members of this class.

1. String
2. Char
3. Character
4. Wrapper

7. The indexOf and lastIndexOf methods are members of this class.

1. String
2. Integer
3. Character
4. Wrapper

8. The substring, getChars, and toCharArray methods are members of this class.

1. String
2. Float
3. Character
4. Wrapper

9. This String class method performs the same operation as the + operator when used on strings.

1. add
2. join

**3. concat**

4. plus

10. This String class method adds or takes away spaces from the beginning of each line in a multiline string.

1. pad

2. spacing

**3. indent**

4. margin

11. This String class method returns a string that contains the contents of the calling String object repeated a specified number of times.

1. replicate

2. duplicate

3. copy\_n

**4. repeat**

12. The String class has several overloaded versions of a method that accepts a value of any primitive data type as its argument and returns a string representation of the value. The name of the method is \_\_\_\_\_.

1. stringValue

**2. valueOf**

3. getString

4. valToString

13. If you do not pass an argument to the `StringBuilder` constructor, the object will have enough memory to store this many characters.

1. 16

2. 1

3. 256

4. Unlimited

14. This is one of the methods that are common to both the `String` and `StringBuilder` classes.

1. append

2. insert

3. delete

4. length

15. To change the value of a specific character in a `StringBuilder` object, use this method.

1. changeCharAt

2. setCharAt

3. setChar

4. change

16. To delete a specific character in a `StringBuilder` object, use this method.

1. deleteCharAt

2. removeCharAt

3. removeChar

4. expunge

17. The character that separates tokens in a string is known as a \_\_\_\_\_.

1. separator
2. tokenizer
- 3. delimiter**
4. terminator

18. This String method breaks a string into tokens.

1. break
2. tokenize
3. getTokens
- 4. split**

19. This method converts a string to an int and returns the int value.

1. int.ParseInt
- 2. Integer.parseInt**
3. Integer.valueOf
4. Integer.castInt

20. These static final variables are members of the numeric wrapper classes and hold the minimum and maximum values for a particular data type.

- 1. MIN\_VALUE and MAX\_VALUE**
2. MIN and MAX
3. MINIMUM and MAXIMUM
4. LOWEST and HIGHEST

21. **True or False:** Character testing methods, such as isLetter, accept strings as arguments and test each character in the string.

22. **True or False:** If the toUpperCase method's argument is already uppercase, it is returned as is, with no changes.

23. **True or False:** If toLowerCase method's argument is already lowercase, it will be inadvertently converted to uppercase.
24. **True or False:** The startsWith and endsWith methods are case-sensitive.
25. **True or False:** There are two versions of the regionMatches method: one that is case-sensitive and one that can be case-insensitive.
26. **True or False:** The indexOf and lastIndexOf methods can find characters, but cannot find substrings.
27. **True or False:** The String class's replace method can replace individual characters, but cannot replace substrings.
28. **True or False:** The StringBuilder class's replace method can replace individual characters, but cannot replace substrings.
29. **True or False:** You can use the = operator to assign a string to a StringBuilder object.

## Algorithm Workbench

5.

```
package Assignment4.Algorithm_WorkBench;

public class Main {
    public static void main(String[] args) {
        System.out.println(comChecker(string: ".com"));
    }

    public static boolean comChecker(String string){
        if(string.endsWith(suffix: ".com")){
            return true;
        }
        return false;
    }
}
```

6.

```
package Assignment4.Algorithm_WorkBench;

public class Main {

    Run | Debug
    public static void main(String[] args) {

        System.out.println(comChecker(string: "HelloWorld.com"));

    }

    public static boolean comChecker(String string){

        if(string.toLowerCase().endsWith(suffix: ".com")){
            return true;
        }
        return false;
    }

}
```

9.

```
package Assignment4.Algorithm_WorkBench;

public class Split {
    Run | Debug
    public static void main(String[] args) {

        String string = "cookies>milk>fudge:cake:ice cream";
        String[] words = string.split(regex: "[>:]");

        StringBuilder newString = new StringBuilder();

        for(int i = 0; i < words.length; i++){
            if[i < words.length-1]){
                newString.append(words[i]);
                newString.append(str: ", ");
            }
            else{
                newString.append(words[i]);
            }
        }

        System.out.println(newString);

    }
}
```

10.

```
package Assignment4.Algorithm_WorkBench;

public class toInt {
    Run | Debug
    public static void main(String[] args) {

        double d = 2436;

        if(d < Integer.MAX_VALUE){
            int j = (int)d;
        }

    }
}
```

13.

AXYAXYAXY