

## Assignment 1 Overview

- Get started early!
- Read assignment specifications VERY CAREFULLY! It is often assignment on 1 that people get caught for silly mistakes that end up costing them lots.
- Read over the marking scheme before you start and after you complete the assignment – at the outset it's not too late to change a fundamental design issue, and at the end you might catch something small that will have a significant impact.

### Useful Tools & Reminders:

a) `TextInput` – recall:

```
public String readLine()  
           throws IOException
```

Read the rest of the line from the input, including the end-of-line character(s), advancing to the next line of input. Does not return any tabs and spaces at the beginning of the line.

**Returns:**

the entire string except for the end-of-line character(s).

**Throws:**

IOException - if the file cursor is already at the end of the file.

b) `TextOutput`

```
public TextOutput(String fileName,  
                  boolean append)
```

Construct a new `TextOutput` object which prints to the named file.

**Parameters:**

`fileName` - the name of the file where output should be placed.

`append` - append to an existing file if true; create a new file or overwrite an existing file if false

c) `String`

```
public char charAt(int index)
```

Returns the character at the specified index. An index ranges from 0 to `length()` - 1. The first character of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

**Parameters:**

`index` - the index of the character.

**Returns:**

the character at the specified index of this string. The first character is at index 0.

**Throws:**

IndexOutOfBoundsException - if the `index` argument is negative or not less than the length of this string.

## substring

```
public String substring(int beginIndex)
```

Returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.

Examples:

```
"unhappy".substring(2) returns "happy"  
"Harbison".substring(3) returns "bison"  
"emptiness".substring(9) returns "" (an empty string)
```

### Parameters:

beginIndex - the beginning index, inclusive.

### Returns:

the specified substring.

### Throws:

IndexOutOfBoundsException - if beginIndex is negative or larger than the length of this String object.

## substring

```
public String substring(int beginIndex,  
                        int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.

Examples:

```
"hamburger".substring(4, 8) returns "urge"  
"smiles".substring(1, 5) returns "mile"
```

### Parameters:

beginIndex - the beginning index, inclusive.

endIndex - the ending index, exclusive.

### Returns:

the specified substring.

### Throws:

IndexOutOfBoundsException - if the beginIndex is negative, or endIndex is larger than the length of this String object, or beginIndex is larger than endIndex.

## trim

```
public String trim()
```

Removes white space from both ends of this string.

If this `String` object represents an empty character sequence, then a reference to this `String` object is returned.

This method may be used to trim whitespace from the beginning and end of a string; in fact, it trims all ASCII control characters as well.

**Returns:**

this string, with white space removed from the front and end.