

COMP 110/L Lecture 17

Kyle Dewey

Outline

- `char type`
- **Methods on `String`**
 - `length`
 - `charAt`
- **Two-dimensional arrays**

char **Type**

char Type

Represents a single character

char Type

Represents a single character

'a'

(Single quotes denote char)

char Type

Represents a single character

'a'

(Single quotes denote char)

'b'

char Type

Represents a single character

'a'

(Single quotes denote char)

'b'

'\n'

char Type

Represents a single character

'a'

(Single quotes denote char)

'b'

'\n'

(newline)

-Newline character (still a single character)

char Type

Represents a single character

'a'

(Single quotes denote char)

'b'

'\n'

(newline)

'\t'

char Type

Represents a single character

'a'

(Single quotes denote char)

'b'

'\n'

(newline)

'\t'

(tab)

-Again, still a single character

`char` With String Concatenation

Works just like it does with `int`

char With String Concatenation

Works just like it does with `int`

```
'h' + "ello"
```

char With String Concatenation

Works just like it does with `int`

```
'h' + "ello"  
"hello"
```

char With String Concatenation

Works just like it does with `int`

```
'h' + "ello"
```

```
"hello"
```

```
"goodbye" + 'e'
```

char With String Concatenation

Works just like it does with `int`

```
'h' + "ello"
```

```
"hello"
```

```
"goodby" + 'e'
```

```
"goodbye"
```

Methods on `String`

length ()

Returns the number of characters in the `String`

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

5

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

5

```
"goodbye".length ()
```

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

5

```
"goodbye".length ()
```

7

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

5

```
"goodbye".length ()
```

7

```
"".length ()
```

length ()

Returns the number of characters in the `String`

```
"hello".length ()
```

5

```
"goodbye".length ()
```

7

```
"".length ()
```

0

Example:

`StringLength.java`

length () vs. length

- `length ()` is defined for Strings
 - `"foo".length ()`
- `length` is defined for arrays
 - `(new int[] {1, 2, 3}).length`

charAt ()

Gets the char at a given 0-indexed position in a String

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt(1)
```

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt (1)
```

```
'o'
```

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt (1)
```

```
'o'
```

```
"what".charAt (3)
```

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt (1)
```

```
'o'
```

```
"what".charAt (3)
```

```
't'
```

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt (1)
```

```
'o'
```

```
"what".charAt (3)
```

```
't'
```

```
"kangaroo".charAt (5)
```

charAt ()

Gets the char at a given 0-indexed position in a String

```
"something".charAt (1)
```

```
'o'
```

```
"what".charAt (3)
```

```
't'
```

```
"kangaroo".charAt (5)
```

```
'r'
```


Example:

`StringCharAt.java`

Two-Dimensional Arrays

Recap: Arrays

You're seen arrays...

Recap: Arrays

You're seen arrays...

```
new int[] {1, 2, 3}
```

Recap: Arrays

You're seen arrays...

```
new int[] {1, 2, 3}
```

```
new String[] {"foo", "bar"}
```

Recap: Arrays

You're seen arrays...

```
new int[] {1, 2, 3}
```

```
new String[] {"foo", "bar"}
```

```
public static int[] baz(int[] input) {  
    ...  
}
```

Arrays of Arrays

...but you can also make arrays of arrays.

AKA multi-dimensional arrays.

-Up until this point, you've only been working with single-dimensional arrays

Arrays of Arrays

...but you can also make arrays of arrays.

AKA multi-dimensional arrays.

```
new int[][] { new int[] {1, 2, 3},  
              new int[] {3, 4, 5},  
              new int[] {6, 7, 8} }
```

-Up until this point, you've only been working with single-dimensional arrays

Arrays of Arrays

...but you can also make arrays of arrays.

AKA multi-dimensional arrays.

```
new int[][] { new int[] {1, 2, 3},  
              new int[] {3, 4, 5},  
              new int[] {6, 7, 8} }
```

```
public static void blah(String[][] in) {  
    ...  
}
```

-Up until this point, you've only been working with single-dimensional arrays

Example:

`PrintAll12D.java`

Example:

```
SetAllToCount.java
```

Example:

Create2DArray.java

Important Points

- Accessing one element: `array[row][column]`
- Inner arrays can be of different lengths
- Edits to arrays persist across method calls
 - This is different for non-arrays, which copy
 - More on this difference later in the course