

CS 152
Computer Programming
Fundamentals
Strings

Brooke Chenoweth

University of New Mexico

Spring 2024

String and char

- String is a *class*: `java.lang.String`

```
String s1 = "Hello"; //ok
String s2 = "H";     //ok
String s3 = "";      //empty string is ok
```

- char is a *primitive type*:

```
char c = 'H';        //ok
char c = 'Hello';   //Syntax ERROR
char c = '';         //Syntax ERROR
```

String Concatentation

```
public static void main(String[] args) {  
    String a = "Hello";  
    String b = "World";  
  
    String c = a + b;  
  
    System.out.println(a);  
    System.out.println(c);  
    System.out.println(a + " " + b);  
}
```

Output:

Hello

HelloWorld

Hello World

Overloaded Operator: +

concatenation operator and addition operator

```
public class Hello {  
    public static void main(String[] args) {  
        int a = 5;  
        int b = 7;  
        System.out.println(a + b);  
        System.out.println("Sum is " + a + b);  
        System.out.println("Sum is " + (a + b));  
        System.out.println(a + b + "boo");  
    }  
}
```

12

Sum is 57

Sum is 12

12boo

Comparing Strings for equality

```
String s1 = "CS152";  
String s2 = "CS";  
s2 += 152;  
  
if(s1 == s2) System.out.println("Same");  
if(s1.equals(s2)) System.out.println("Equal");
```

- Use == operator when comparing primitive types
- Use equals method when comparing objects
- Using == on objects will compare if same object in memory, not what you want usually

String methods – length

```
String str1 = "Brooke";  
String str2 = "Chenoweth";  
System.out.println(  
    str1.length() + ", " + str2.length());
```

Output: 6, 9

- Every String object has data and methods.
- In the example, str1 is a String object.
- The data in str1 is "Brooke"
- str1 has the method length() which returns an int equal to the number of characters in its data.

Using a String object's length()

```
public class Hello {  
    public static void main(String[] args) {  
        String a = "Hello";  
        String b = "Everyone";  
        System.out.println(a.length());  
        System.out.println(b +  
            " has a length of " + b.length() +  
            " characters");  
    }  
}
```

5

Everyone has a length of 8 characters

Without space characters, output will run together.

charAt

Every String object has the method:

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

```
String name = "Ralph";  
int index = 1;  
char letter;  
letter = name.charAt(index);  
System.out.println(letter);
```

Output: a

Using a String object's charAt(int i)

```
public class TestClass {  
    public static void main(String[] args) {  
        String a = "Lisa";  
        System.out.println(a.length());  
        System.out.println(a.charAt(1));  
        System.out.println(a.charAt(3));  
        System.out.println(a.charAt(4));  
    }  
}
```

4

i

a

java.lang.StringIndexOutOfBoundsException:
String index out of range: 4

toUpperCase

```
public class Hello {  
    public static void main(String[] args) {  
        String str = "Angela";  
        System.out.println(str); //Angela  
  
        str.toUpperCase();  
        System.out.println(str); //Angela  
  
        str = str.toUpperCase();  
        System.out.println(str); //ANGELA  
    }  
}
```

substring

```
String name1 = "Brooke";  
String name2 = "Chenoweth";  
String name3 = name1.substring(2);  
String name4 = name2.substring(2);  
String name5 = name2.substring(4);  
  
System.out.println(name1);  
System.out.println(name2);  
System.out.println(name3);  
System.out.println(name4);  
System.out.println(name5);
```

Output:

Brooke

Chenoweth

ooke

enoweth

oweth

substring: An Overloaded Method

Two versions of substring:

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

```
public static void main(String[] args) {  
    String name = "Chenoweth";  
  
    System.out.println(name.substring(4));  
    System.out.println(name.substring(4,6));  
}
```

Output:

oweth

ow