

LEC 03

CSE 121

Strings, char, and Variables

Questions during Class?

Raise hand or send here

sli.do #cse121



BEFORE WE START

*Talk to your neighbors:**What is your favorite emoji?* 😊**Respond on sli.do!**Music: [🌿 CSE 121 26au Lecture Tunes 🌿](#)**Instructor:** Miya Natsuhara and Anjali Pal

TAs:	Samrutha	Tamsyn	Owen	Jennie
	William	Cedric	Jameson	Anisha
	Samvedha	Anum	Paul	TJ
	Syreeta	Navya	Gurbax	Ivory
	Ryan	Gene	Tony	Sarah W
	Anant	Margot	Lucas	Avery
	Henry	Sarah K	Alex	
	Reese	Ethan	Jesse	
	Savannah	Evelyn	Matthew	

Agenda (1/3)

- **Announcements, Reminders (now)**
- Casting, More Variables and Operators!
- Strings and Characters Review
 - code example!

Announcements, Reminders

- Programming Project 0 is out, due Tuesday 10/13
- Computer Quiz 0 will take place from Wednesday 10/14 - Wednesday 10/21
 - Sign up for a computer quiz time slot through Canvas!
 - Wide variety of time slots – let us know if you *none* of them work for your schedule!
 - Quizzes will generally be taken on your own devices – if you don't have a laptop/surface, [please let us know](#) so we can make sure to provide a computer for you to use!
 - Ed posts: [Computer Quiz Scheduling](#), **Computer Quiz Logistics**

Agenda (2/3)

- Announcements, Reminders
- C0 Reflection Recap
- **Casting, More Variables and Operators! (now)**
- Strings and Characters Review
 - code example!

PCM: Variables

- Recall: Variables allow us to give a name to a specific value
 - 3 parts: declaration, initialization, usage

- Example:

```
String bestBoy = "gumball";  
System.out.println(bestBoy);
```

- Declaration: `int num;`
- Initialization: `num = 30;`
- Or all in one line: `int num = 30;`

PCM: Casting

- Java will do some type conversions for us
 - E.g., `int` to `double`, `double` to `String`, `int` to `String`
- **BUT** some conversions Java won't do for us...
 - Nonsensical conversions (e.g., `"Gumball"` to `int`)
 - Conversions that are **"lossy"** (e.g., `double` to `int`)
 - We can ask Java to **typecast** for us

```
double num = 8.83;  
int numInt = (int) num;
```

New: Manipulating Variables

They're made to be manipulated, modified, and re-used!

```
int myFavoriteNumber = 7;  
int tripleFavNum = myFavoriteNumber * 3;  
myFavoriteNumber = myFavoriteNumber + 3;
```



Note! This doesn't really make any mathematical sense. That's because in Java, = is *assignment*, not equality!

New Operator: +=

```
myFavoriteNumber = myFavoriteNumber + 3;
```

This pattern is so common, we have a shorthand for it!

```
myFavoriteNumber += 3;
```

This works for both numeric addition and string concatenation!

More Shorthand Operators

The shorthands `--`, `*=`, `/=`, and `%=` exist too!

```
myFavoriteNumber /= 3;
```

Should this work for integers? Doubles? Strings?

Even Shorter Shorthands

There are even shorter operators for “incrementing” and “decrementing”!

```
myFavoriteNumber++; // myFavoriteNumber += 1;  
myFavoriteNumber--; // myFavoriteNumber -= 1;
```

Should this work for integers? Doubles? Strings?



Practice: Think

[sli.do](#)[#cse121](#)

What values do a, b, and c hold after this code is executed?

```
int a = 10;  
int b = 30;  
int c = a + b;  
c -= 10;  
a = b + 5;  
b /= 2;
```

A. 10, 30, 40

B. 35, 15, 30

C. 35, 15.5, 30

D. 20, 15, 30



Practice: Pair

sli.do

#cse121

What values do a, b, and c hold after this code is executed?

```
int a = 10;  
int b = 30;  
int c = a + b;  
c -= 10;  
a = b + 5;  
b /= 2;
```

A. 10, 30, 40

B. 35, 15, 30

C. 35, 15.5, 30

D. 20, 15, 30

Agenda (3/3)

- Announcements, Reminders
- Typecasting
- Casting, More Variables and Operators!
- **Strings and Characters Review (now)**
 - code example!

PCM: Strings & chars

- Recall: String literals are a sequence of characters that are *strung* together, begin and end with `""`
 - Use zero-based indexing
- A `char` represents a single character
 - Begin and end with single quotes (`' '`)
 - Strings are made up of chars!

g	u	m	b	a	l	l
0	1	2	3	4	5	6

```
char letter = 'g';  
char anotherLetter = 'b';
```

PCM: String Methods

Usage: `<string_variable>.<method>(...)`

Method	Description
<code>length()</code>	Returns the length of the string.
<code>charAt(i)</code>	Returns the character at index <i>i</i> of the string
<code>indexOf(s)</code>	Returns the index of the first occurrence of <i>s</i> in the string; returns -1 if <i>s</i> doesn't appear in the string
<code>substring(i, j)</code> or <code>substring(i)</code>	Returns the characters in this string from <i>i</i> (inclusive) to <i>j</i> (exclusive); if <i>j</i> is omitted, goes until the end of the string
<code>contains(s)</code>	Returns whether or not the string contains <i>s</i>
<code>equals(s)</code>	Returns whether or not the string is equal to <i>s</i> (case-sensitive)
<code>equalsIgnoreCase(s)</code>	Returns whether or not the string is equal to <i>s</i> ignoring case
<code>toUpperCase()</code>	Returns an uppercase version of the string
<code>toLowerCase()</code>	Returns a lowercase version of the string



Practice: Think

sli.do

#cse121

Suppose `s` contains the String "bubble gum".

Which statement would result in `s` containing "Gumball" instead?

b	u	b	b	l	e		g	u	m
0	1	2	3	4	5	6	7	8	9

A. `s.substring(7) + "ball";`

B. `s = s.substring(7, 9) + "ball";`

C. `s = s.charAt(7).toUpperCase() + "ball";`

D. `s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";`



Practice: Pair

sli.do

#cse121

Suppose `s` contains the String "bubble gum".

Which statement would result in `s` containing "Gumball" instead?

b	u	b	b	l	e		g	u	m
0	1	2	3	4	5	6	7	8	9

A. `s.substring(7) + "ball";`

B. `s = s.substring(7, 9) + "ball";`

C. `s = s.charAt(7).toUpperCase() + "ball";`

D. `s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";`