

# CS 113 – Computer Science I

## Lecture 11 – Review & Loops

Tuesday 10/22/2024

# Announcements

- HW05
  - Due Tuesday after fall break
- HW06:
  - Optional, due Monday night
- No lab today
- Office hours:
  - Adam's Tuesday (today) 2:40-4:45pm Thursday 2:40-4:00pm

# Agenda

- **Review**
- While Loops
- For Loops
- Arrays of Arrays

# Midterm – Thursday 10/24

In class, closed book

Terminal commands, vim, directory structure  
variables (int, double, char, bool, string, array)

Errors

Expressions

Methods

Frame diagrams

Conditionals

Recursion

Practice exam is on course website

# Review topics

Types of Errors

Boolean Operators

Conditionals (if-else)

Stack Diagrams

Arrays

# Errors

compilation, runtime, or logic?

Example.java:3: error: incompatible types:  
String cannot be converted to int

```
int x = 100 + "s";
```

# Errors

compilation, runtime, or logic?

```
Exception in thread "main"  
java.lang.ArithmeticException: / by zero  
    at Example.main(Example.java:3)
```

# Errors

compilation, runtime, or logic?

```
Example.java:3: error: cannot find symbol
    System.ouy.println("Hello, World!");
           ^
```

symbol: variable ouy

location: class System



# Errors

compilation, runtime, or logic?

```
Exception in thread "main"  
java.util.IllegalFormatException: d !=  
java.lang.String
```

# Errors

what's a logic error?

# Variables

1. How do I declare a variable?
1. How do I assign a value to a variable?

# Variables

assignment or declaration?

- `int x;`
- `x = foo();`
- `int y = 99;`

# Types

1. List some types
1. What is a primitive type?
  - a. list some
1. What is an object (reference type?)
  - a. list some
1. What are the differences?

# Objects

Strings and arrays are **NOT** primitives

They are objects

Explains why we can't use "==" to compare Strings

"==" checks if two objects are the same  
not if the two values are the same

# Types

types.java

# Methods

methods.java



# Methods - scope

Scope.java

# Strings

- concatenation
- length
- substring

Strings.java

# Truth tables

`int temp = 68;`

`double val = 10.5;`

`boolean raining = true;`

Expression	Value
<code>temp &gt; 80</code>	
<code>val != 5.6</code>	
<code>val &gt;= 10.1</code>	
<code>raining == true</code>	
<code>!raining</code>	
<code>raining == false</code>	
10/22/2024	CS 131 – Fall '23 - Lecture 11

# If statements

If.java

# Arrays

- equality
- length
- indexing
- AOB
- is an array a primitive or object?
- default values

Arrays.java

# Initializing empty arrays

```
int[] nums = new int[3];  
    [0, 0, 0]
```

```
String[] strs = new String[3];  
    [null, null, null]
```

# Coding Question

Write a method `count` which takes a `String` and a `char` and returns an `int` indicating how many times the character appears in the `String`

# Agenda

- Review
- **While Loops**
- For Loops
- Arrays of Arrays



# Exercise

Suppose we wanted to ask the user for 6 numbers (int) and output their sum?

# Loops

- Easy way to repeat some computation
- Two kinds of loops:
  - While
  - For
- Loops repeat block of code until the condition becomes false

# While loop

While a condition is true, run a block of code

```
while(condition) {  
    //run the code in this block  
}
```

# Example: While Loop

```
int val = 0;
int sum = 0;

int count = 0;
while (count < 6) {
    System.out.print("Enter a number: ");
    val = sc.nextInt();
    sum = sum + val;
    count = count + 1;
}
System.out.println("The sum is "+sum);
```

# Tracing Loops

```
int sum = 1;
int count = 0;
while (count < 3) {
    sum = sum + 2;
    count = count + 1;
}
```

Iteration	Count < 6	count	sum

# Tracing Loops

```
int sum = 1;
int count = 0;
while (count < 3) {
    sum = sum + 2;
    count = count + 1;
}
```

Iteration	Count < 6	count	sum
0	T	0	1
1	T	1	3
2	T	2	5
3	T	3	7

# Exercise: Tracing loops

```
int sum = 10;  
int count = 0;  
while (count < 6) {  
    sum = sum - 1;  
    count = count + 2;  
}
```

Iteration	Count < 6	count	sum

# Exercise: Tracing loops

```
int sum = 10;  
int count = 0;  
while (count < 6) {  
    sum = sum - 1;  
    count = count + 2;  
}
```

Iteration	Count < 6	count	sum
0	T	0	10
1	T	2	9
2	T	4	8
3	T	6	7
4	F		



# Accumulator pattern

Idea: Repeatedly update a variable (typically in a loop)

Pattern:

1. Initialize accumulator variable
2. Loop until done
  1. Update the accumulator variable

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

`sum = sum + 2`

`count = count + 1`

`count = count - 1`

`product = product * 2`

`divisor = divisor / 2`

`message = message + "lol!"`

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

sum = sum + 2	
count = count + 1	
count = count - 1	
product = product * 2	
divisor = divisor / 2	
message = message + " lol"	

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

sum = sum + 2	sum += 2
count = count + 1	
count = count - 1	
product = product * 2	
divisor = divisor / 2	
message = message + " lol"	

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

sum = sum + 2	sum += 2
count = count + 1	count += 1
count = count - 1	
product = product * 2	
divisor = divisor / 2	
message = message + " lol"	

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

sum = sum + 2	sum += 2
count = count + 1	count += 1
count = count - 1	count -= 1
product = product * 2	
divisor = divisor / 2	
message = message + " lol"	

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

<code>sum = sum + 2</code>	<code>sum += 2</code>
<code>count = count + 1</code>	<code>count += 1</code>
<code>count = count - 1</code>	<code>count -= 1</code>
<code>product = product * 2</code>	<code>product *= 2</code>
<code>divisor = divisor / 2</code>	
<code>message = message + " lol"</code>	

# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

<code>sum = sum + 2</code>	<code>sum += 2</code>
<code>count = count + 1</code>	<code>count += 1</code>
<code>count = count - 1</code>	<code>count -= 1</code>
<code>product = product * 2</code>	<code>product *= 2</code>
<code>divisor = divisor / 2</code>	<code>divisor /= 2</code>
<code>message = message + " lol"</code>	



# Convenience syntax: Assignment

Because updating variable values is so common, language such as Java provide shorthand syntax for it

- Analogy: contractions in English

<code>sum = sum + 2</code>	<code>sum += 2</code>
<code>count = count + 1</code>	<code>count += 1</code>
<code>count = count - 1</code>	<code>count -= 1</code>
<code>product = product * 2</code>	<code>product *= 2</code>
<code>divisor = divisor / 2</code>	<code>divisor /= 2</code>
<code>message = message + " lol"</code>	<code>message += " lol"</code>

# Exercise: Write a program that computes powers of 2

Write a program, LoopPow2.java, that computes powers of twos. For example,

```
$ java LoopPow2
Enter an exponent: 0
2 to the power of 0 is 1

$ java LoopPow
Enter an exponent: 1
2 to the power of 1 is 2

$ java LoopPow
Enter an exponent: 4
2 to the power of 4 is 16
```

# Agenda

- While Loops
- **For Loops**
- Arrays of Arrays

# Example: For Loop



```
int val = 0;
String valStr = "";
int sum = 0;

for (int count = 0; count < 6; count = count +1) {
    System.out.print("Enter a number: ");
    valStr = System.console().readLine();
    val = Integer.parseInt(valStr);
    sum = sum + val;
}
System.out.println("The sum is "+sum);
```

# Example: For Loop

initialize

condition

update

```
for (int count = 0; count < 6; count = count + 1) {  
  
}
```

# Exercise: Tracing loops

```
String pattern = "";  
for (int i = 0; i < 3; i++) {  
    pattern = pattern + "*";  
}  
System.out.println(pattern);
```

Iteration	i < 3	i	pattern

# Exercise: Tracing loops

```
String pattern = "";  
for (int i = 0; i < 3; i++) {  
    pattern = pattern + "*";  
}  
System.out.println(pattern);
```

Iteration	$i < 3$	$i$	pattern
0	T	0	""
1	T	1	"*"
2	T	2	"**"
3	F	3	"***"

# Exercise: LoopPattern.java

```
$ java LoopPattern
```

```
Enter a length: 5
```

```
*_*_*
```

```
$ java LoopPattern
```

```
Enter a length: 10
```

```
*_*_*_*_*_*
```

```
$ java LoopPattern
```

```
Enter a length: 0
```

```
$ java LoopPattern
```

```
Enter a length: 1
```

```
*
```



# Exercise: Nested loops

```
$ java Square
```

```
Enter a size: 5
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
$ java Square
```

```
Enter a size: 1
```

```
*
```

```
$ java Square
```

```
Enter a size: 0
```

# Iterating through an array

Write a method called `printArray` that takes in an array of integers and prints out the values in each array:

*`printArray({1,2,3,4}) -> "1 2 3 4"`*

# While vs For loop

Use a for loop when we know the number of iterations we want

Use a while loop when we don't know the number of iterations before hand

# Agenda

- Announcements
- While Loops
- For Loops
- **Arrays of Arrays**

# Arrays of Arrays

`int[] array1` is an array of ints

`String[] array2` is an array of Strings

What is `int[][] array3`?

An array of integer arrays

What is `String[][] array4`?

An array of String arrays

# 2D array example

What does `int[][] array = new int[4][3]` look like?

# 2D array example

What does `int[][] array = new int[4][3]` look like?

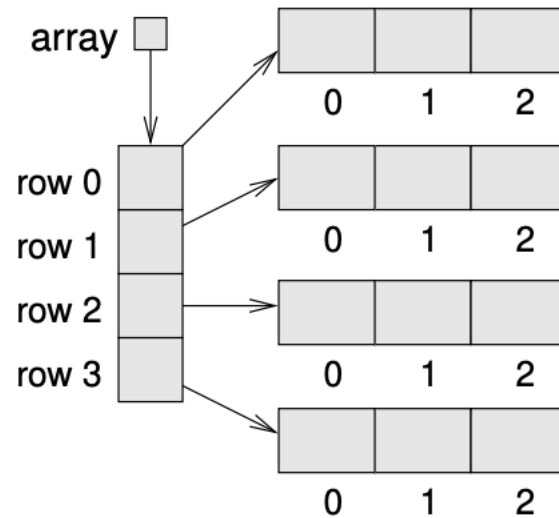


Figure 15.3: Storing rows and columns with a 2D array.

# 2D Array

Useful for representing a:

- Grid
- Boardgame
- Matrix
- Table
- ...



# Traversing through a 2D array

What type of loop should we use?

if we know the length, then a for loop

Pseudocode/algorithm:

for array in 2D array:

for item in array: