

# CS 113 – Computer Science I

## Lecture 02 – Data Types, Variables, Expressions, Scanner

Thursday 09/10/2023

# Announcements

HW00 – due last night

- Everyone submitted on gradescope!
- Great job!
- Expect grades back by end of week (hopefully before)

HW01 – due this coming Monday night

Office hours:

- Adams: TBD
- TAs:
  - Sunday – Friday
  - 7:00pm-9:30pm Park 230

# Outline

Review

Reading in data

Data Types

Variables

Expressions

Operators

## A simple java program

```
1 // A java program to print a message
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5         // Prints out message to standard output
6         System.out.println("Hello World!");
7     }
8 }
```

# What are the errors here?

```
public clas SyntaxErrors {  
  
    public static void main(String args) {  
        System.out.println("Hello World");  
  
    }  
}
```

# Types of Errors

- Syntax error
  - didn't follow the rules of the programming languages "grammar"
  - caught by the compiler
- Runtime error:
  - program compiles, but crashes when executing
- Logic error:
  - program compiles and runs but doesn't do what we intended

# Navigating Linux Directory

## Terminal commands

- List files
  - `ls`
- Move directories
  - `cd`
- Print the path to working directory
  - `pwd`
- Compile a java program
  - `javac <java file>`
- Run a java program
  - `java <class name>`

# Folders & Directories

- Computer is structured as a folder-system.
  - Folders (directories) can contain files and other directories
- Organizing programs in directories
- special directories:
  - .. (double dot) - parent directory



# Reading in data

- Way to communicate to our program by passing data to our program
- `System.console().readline();`

# Storing Data

# Data Types

- Way to store information in programs
- `int`: whole numbers
- `double`: numbers with decimal points
- `String`: anything between quotations

# Why have types

- Memory
- Readability and Documentation
- Enforcement of proper operations

# Variables - Holders for values

- `String greeting;`
  - Creates a variable called “greeting” that can store a string

- `int a, b, c;`
  - Creates 3 variables that can store integers

- `a = 3;`


Assignment statement

- `int d = 10;`

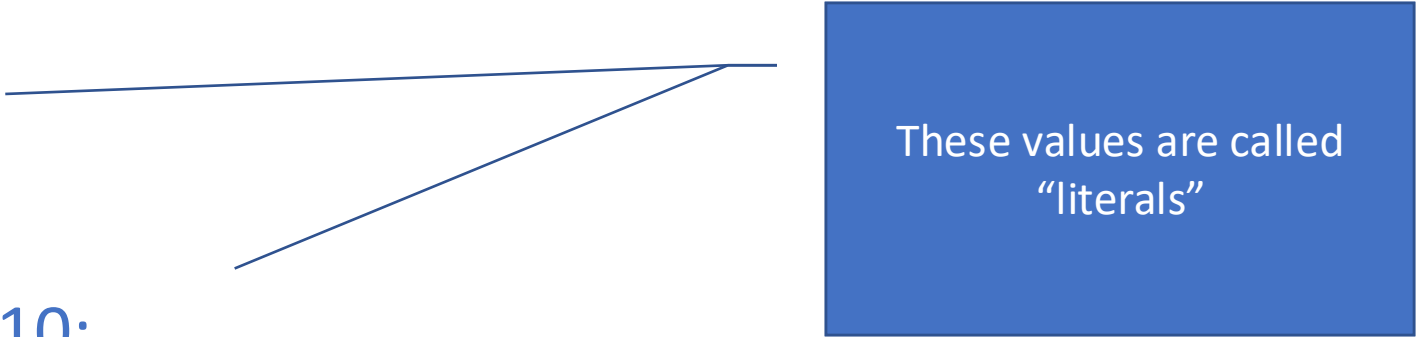
Declaration & Assignment statement  
Best Practice!

Declaration  
statements:  
Do not store any value

# Variables - Holders for values

- String greeting;
    - Creates a variable called “greeting” that can store a string
  - int a, b, c;
    - Creates 3 variables that can store integers
  - a = 3;
  - int d = 10;
- 

# Variables - Holders for values

- `String greeting;`
    - Creates a variable called “greeting” that can store a string
  - `int a, b, c;`
    - Creates 3 variables that can store integers
  - `a = 3;`
  - `int d = 10;`
- 

# Properties of Variables

Variables have the following properties:

- Names
- Type of Data
- Location
  - Where on the computer the variable is stored

Example:

String greeting;

Name

Type of data



# Printing Variables

- `int d = 10;`
- Creates 3 variables that can store integers

# Variable Examples

a	b	c
-	-	-

# Variable Examples

- `int a, b;`

a	b	c
-	-	-

# Variable Examples

- `int a, b;`

a	b	c
undefined	undefined	-

# Variable Examples

- `int a, b;`
- `String c = "Coco";`

a	b	c
undefined	undefined	-

# Variable Examples

- `int a, b;`
- `String c = "Coco";`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"



# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`
- `b = a;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`
- `b = a;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"
3	3	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`
- `b = a;`
- `a = 5;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"
3	3	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`
- `b = a;`
- `a = 5;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"
3	3	"Coco"

# Variable Examples

- `int a, b;`
- `String c = "Coco";`
- `a = 3;`
- `b = a;`
- `a = 5;`

a	b	c
undefined	undefined	-
undefined	undefined	"Coco"
3	undefined	"Coco"
5	3	"Coco"

# Rules for naming variables

- Case sensitive
- Can't:
  - start with a number
  - Contain special characters: \*, +, -, /, %, \$, #, etc.
  - No spaces
  - Special words:
    - String, int, main, for, while, ...

# Exercise: Miles to Kilometers

Write a program called `MilesToKMs.java` that asks a user for miles and then prints out the distance in kilometers

- `java MilesToKMs`  
50 miles is 80 kilometers

# Converting Types (Numbers)

- Double to integer:
  - `(int) 3.14;`
  - `int a = (int) 3.14; //` Store the converted double in a var
- Storing an integer as a double:
  - `double b = 6;`



# Converting Types (Strings & Numbers)

- Integer to String
  - `int a = 23;`
  - `String numMajors = String.valueOf(a);`
- String to integer
  - `int x = Integer.parseInt("40");`
- String to double
  - `double a = Double.parseDouble("40.11");`

# Operators & Expressions

- Examples of operators:

- $+$ ,  $-$ ,  $/$ ,  $*$ ,  $\%$

- Expression

- $55 + c$



Operator

Operands

# Order of operations

- $24 + 10 / 2;$
- $(24 + 10) / 2;$
- Operations between floats and ints:
  - $1 / 3$
  - $1 / 3.0$

# Exercise:

Expression	Value	Data Type
-4		
3.76		
"42.64"		
10 + 3.3		
9 - 5 * 1		
"hot" + "dog"		

# String Operators (Textbook: 2.8)

What is the term for combining strings together?

- Concatenation

What is the concatenation operator?

- +

# Exercise: Miles to Kilometers

Write a program called `MilesToKMs.java` that asks a user for miles and then prints out the distance in kilometers

- `java MilesToKMs`  
50 miles is 80 kilometers

# Reading in Data

# Scanner class

Another way for reading in data

```
Scanner sc = new Scanner(System.in);
```

System.in specifies we are reading from user input

What type is “sc”? Is it an int, double, or string?

It's a Scanner type.



# Using Scanner object

Javadocs:

<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>

Reading in an integer:

```
nextInt();
```

Reading in a string:

```
nextLine();
```

# Formatting Strings

<code>%d</code>	Integer in base 10 (“decimal”)	12345
<code>%,d</code>	Integer with comma separators	12,345
<code>%08d</code>	Padded with zeros, at least 8 digits wide	00012345
<code>%f</code>	Floating-point number	6.789000
<code>%.2f</code>	Rounded to 2 decimal places	6.79
<code>%s</code>	String of characters	"Hello"
<code>%x</code>	Integer in base 16 (“hexadecimal”)	bc614e

Table 3.1: Example format specifiers

# Review

- 1.How do you print in Java?
- 2.How do you read input?
- 3.What does a declaration statement do?
- 4.What does an assignment statement do?
- 5.Give me an example of an illegal variable name.
- 6.Give me an example of an operator.