CS 113 – Computer Science I

Lecture 02 – Data Types, Variables, Expressions

Thursday 09/07/2023
Announcements

• HW00 – due Monday midnight
  • Survey
  • Growth-mindset
  • Fortune.java
  • Errors
  • *What is CS: Not Computers, not Science*
  • Submit on gradescope

• Office hours:
  • Monday: 2:45-4:00pm
  • Tuesday: 9:30-10:45am (I’ll try to get there by 9:15)
Outline

Review
Reading in data
Data Types
Variables
Expressions
Operators
A simple java program

```java
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 class SyntaxErrors {

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

}
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
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;`

**Declaration statements:** Do not store any value

**Assignment statement**

**Declaration & Assignment statement**

Best Practice!
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;**

What are these (3 & 10) called?
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;`

These values are called “literals”
Properties of Variables

Variables have the following properties:

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

Example:

```java
String greeting;
```
Printing Variables

• `int d = 10;`

• Creates 3 variables that can store integers
Variable Examples

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Variable Examples

- \texttt{int a, b;}

\begin{tabular}{|c|c|c|}
  \hline
  a & b & c \\
  \hline
  - & - & - \\
  \hline
\end{tabular}
Variable Examples

- `int a, b;`

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
</tbody>
</table>
Variable Examples

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

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
</tbody>
</table>
Variable Examples

- `int a, b;`

- `String c = “Coco”;`

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

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

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

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

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>undefined</td>
<td>undefined</td>
<td>&quot;Coco&quot;</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>undefined</td>
<td>&quot;Coco&quot;</td>
</tr>
</tbody>
</table>
### Variable Examples

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

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

• `int a, b;`
• `String c = “Coco”;`
• `a = 3;`
• `b = a;`

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

• int a, b;

• String c = “Coco”;

• a = 3;

• b = a;

• a = 5;

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

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

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
Variable Examples

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

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>undefined</td>
<td>undefined</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>undefined</td>
<td>“Coco”</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>“Coco”</td>
</tr>
</tbody>
</table>
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, ...
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
Order of operations

- $24 + 10 / 2$
- $(24 + 10) / 2$

Operations between floats and ints:
- $1 / 3$
- $1 / 3.0$
String Operators (Textbook: 2.8)

What is the term for combining strings together?
  • Concatenation

What is the concatenation operator?
  • +
Exercise:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;42.64&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 + 3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 - 5 * 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;hot&quot; + &quot;dog&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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