CS 106A, Lecture 4
Introduction to Java

suggested reading:
Java Ch. 1, 2.1-2.4, 3.1-3.4
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
Announcements

• Assignment 1 is out on the website!
  – You must fill out the Honor Code form on the assignment page.

• Roomba example
  – Do on your own if you want!
  – I will post a YouTube video sometime over the weekend about using the debugger

• “Extra” slides: treat these as required reading
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
Bye, Karel!

I will miss you.

Enjoy Java!

See you on the midterm 😞
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
Our First Java Program

Hello, world!
import acm.program.*;

public class HelloWorld extends ConsoleProgram {
    public void run() {
        println("Hello, world!");
    }
}
import acm.program.*;

public class Name extends ConsoleProgram {
  public void run() {
    statements;
  }
}

• Unlike Karel, many programs produce their behavior as text.
• **console**: Text box into which the behavior is displayed.
  – **output**: Messages displayed by the program.
  – **input**: Data read by the program that the user types.
public class Hello extends ConsoleProgram {
    public void run() {
        println("Hello, world!");
        println();
        println("This program produces");
        println("four lines of output");
    }
}
Console Output: println

• A statement that prints a line of output on the console, and goes to the next line.
  – pronounced "print-linn"

• Two ways to use println:
  
  • println("text");
    • Prints the given message as output, and goes to the next line.
    • A message is called a string; it starts/ends with a " quote character.
    • The quotes do not appear in the output.
    • A string may not contain a " character.

  • println();
    Prints a blank line of output.
Console I/O

- `println` allows output text to the user via the console
  - Output is the “O” in “I/O”
- We can also get input from the user via the console!
Console I/O

• println allows out to output text to the user via the console
  – Output is the “O” in “I/O”

• We can also get input from the user via the console!
  – Before we can receive input from the user, we need a way to store it...
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
Variables = Boxes
Variables = Boxes

My computer has space for about a billion boxes
Making a new Variable

```c
int myVariable;
```
Making a new Variable

```c
int myVariable;
```
Variable Types

`int`  – an integer number

`double`  – a decimal number

`char`  – a single character

`boolean`  – true or false
How much does a movie cost?

double – answer is decimal number

Coco
YouTube Movies
Family · 2017 · PG · English

In Disney•Pixar's vibrant tale of family, fun and adventure, an aspiring young musician named Miguel (voice of newcomer Anthony ...

Actors: Anthony Gonzalez, Gael García Bernal, Benjamin Bratt
Directors: Lee Unkrich, Adrian Molina

CC
How many Teslas in Palo Alto?

int – answer is an integer
How much vs. how many
• **variable declaration**: Sets aside memory for storing a value.
  – Variables must be declared before they can be used.

• Syntax:

  \[
  \text{type name;}
  \]

  int zipcode;

  double myGPA;
Assignment

Existing variable name: myVariable

Value: 2

myVariable = 2;
Assignment

• **assignment**: Stores a value into a variable.
  – The value can be an expression; the variable stores its result.

• Syntax:

  \[ \text{name} = \text{expression}; \]

  ```
  int zipcode;
  zipcode = 90210;
  double myGPA;
  myGPA = 1.0 + 2.25;
  ```
Assignment

// integer values
int num;
num = 5;

// real values
double fraction;
fraction = 0.2;

// a single letter
char letter;
letter = ‘c’;

// true or false
boolean isLove;
isLove = true;
• A variable can be declared and initialized in one statement.
  – This is probably the most commonly used declaration syntax.

• Syntax:

  $\text{type name} = \text{expression};$

  double tempF = 98.6;

  int x = (12 / 2) + 3;
`=` Means Assignment

• Assignment uses =, but it is not an algebraic equation.

  = means, "store the value at right in the variable at left"

• The right side expression is evaluated first,
  and then its result is stored in the variable at left.

• What happens here?

```c
int x = 3;
x = x + 2; // ???
```

x

5
Using Variables

• Use + to print a string and a variable's value on one line.

```java
double temperature = (95 + 71 + 82.7) / 3.0;
println("The average temp. was " + temperature);

int enrolled = 11 + 17 + 4 + 19 + 14;
println("There are " + enrolled + " students.");
```

• Output:

The average temp. was 82.9
There are 65 students.
• Once given a value, a variable can be used in expressions:

```java
int x = 3;
println(5 * x - 1);       // 5 * 3 - 1
```

• You can assign a value more than once:

```java
int x = 3;
println(x + " here");    // 3 here
```
```
int x = 4 + 7;
println("now x is " + x); // now x is 11
```
Assignment and types

• A variable can only store a value of its own type.

```java
int x = 2.5;    // Error: incompatible types
```

• An int value can be stored in a double variable.
  – The value is converted into the equivalent real number.

```java
double myGPA = 4;
```

```java
myGPA 4.0
```
Compiler Errors

• A variable can't be used until it is assigned a value.

```java
int x;
println(x);  // Error: x has no value
```

• You may not declare the same variable twice.

```java
int x;
int x;    // ERROR: x already exists

int y = 3;
int y = 5;  // Error: y already exists
```
// Asks the user for an integer by
// displaying the given message
// and stores it in the variable 'a'
int a = readInt(message);

// Asks the user for a double by
// displaying the given message and
// stores it in the variable 'b'
double b = readDouble(message);
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

n1  n2  total
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}
```java
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}
```

This program adds two numbers.
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

n1  n2  total

This program adds two numbers.
Enter n1:
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

This program adds two numbers.
Enter n1: 17
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

```
<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

This program adds two numbers.
Enter n1: **17**
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is "+ total + ".");
    }
}

n1 n2 total
17

This program adds two numbers.
Enter n1: 17
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is "+ total + ".");
    }
}

n1 n2 total
17

This program adds two numbers.
Enter n1: 17
Enter n2: 25
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

This program adds two numbers.
Enter n1: 17
Enter n2: 25

17 25 total

45
```java
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is "+ total + ".");
    }
}
```

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

This program adds two numbers.
Enter n1: 17
Enter n2: 25
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>42</td>
</tr>
</tbody>
</table>

This program adds two numbers.
Enter n1: 17
Enter n2: 25
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is "+total+".");
    }
}

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>n2</td>
<td>total</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>42</td>
</tr>
</tbody>
</table>
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>42</td>
</tr>
</tbody>
</table>

This program adds two numbers.
Enter n1: 17
Enter n2: 25
The total is 42.
public class Add2Integers extends ConsoleProgram {
    public void run() {
        println("This program adds two numbers.");
        int n1 = readInt("Enter n1: ");
        int n2 = readInt("Enter n2: ");
        int total = n1 + n2;
        println("The total is " + total + ".");
    }
}
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
Expressions

• You can combine literals or variables together into expressions using binary operators:

+ Addition  *  Multiplication
– Subtraction  /  Division
% Remainder or Mod or Modulus
Integer remainder %

- The % operator computes the remainder from integer division.
  
  $14 \div 4 = 3 \text{ remainder } 2$
  
  $218 \div 5 = 43 \text{ remainder } 3$
  
- Applications of % operator:
  
  - Obtain last digit of a number: $230857 \div 10$ is 7
  
  - Obtain last 4 digits: $658236489 \div 10000$ is 6489
  
  - See whether a number is odd: $7 \div 2$ is 1, but $42 \div 2$ is 0
What does this do?

```java
println(1 / 2);
```
What does this do?

```java
println(1 / 2);
```

0!
**Integer division**

- When we divide integers, the quotient is also an integer.
  
  \[
  14 \div 4 \text{ is } 3, \text{ not } 3.5. \quad (\text{Java ALWAYS rounds down.})
  \]

\[
\begin{array}{c}
4 \quad ) \quad 14 \\
\underline{12} \\
\quad 2
\end{array}
\quad \begin{array}{c}
10 \quad ) \quad 45 \\
\underline{40} \\
\quad 5
\end{array}
\quad \begin{array}{c}
27 \quad ) \quad 1425 \\
\underline{135} \\
\quad 75 \\
\underline{54} \\
\quad 21
\end{array}
\]

- More examples:
  - \( 32 \div 5 \) is 6
  - \( 84 \div 10 \) is 8
  - \( 156 \div 100 \) is 1
  
  - Dividing by 0 causes an error when your program runs.
Type Interactions

\textbf{int} and \textbf{int} results in an \textbf{int}

\textbf{double} and \textbf{double} results in a \textbf{double}

\textbf{int} and \textbf{double} results in a \textbf{double}

* The general rule is: operations always return the most expressive type
Convert 100° Celsius temperature to its Fahrenheit equivalent:

double c = 100;
double f = 9 / 5 * c + 32;

The computation consists of evaluating the following expression:

The problem arises from the fact that both 9 and 5 are of type int, which means that the result is also an int.
You can fix this problem by converting the fraction to a double:

```c
  double c = 100;
  double f = 9.0 / 5 * c + 32;
```

The computation now looks like this:
Plan For Today

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt
• Let’s write a ConsoleProgram that calculates the tax, tip and total bill for us at a restaurant.
• The program should ask the user for the subtotal, and then calculate and print out the tax, tip and total.
public class Receipt extends ConsoleProgram {
    public void run() {
        double subtotal = readDouble("Meal cost? $");
        double tax = subtotal * 0.08;
        double tip = subtotal * 0.20;
        double total = subtotal + tax + tip;

        println("Tax: $" + tax);
        println("Tip: $" + tip);
        println("Total: $" + total);
    }
}
Recap

• Announcements
• Bye, Karel!
• Console Programs
• Variables
• Expressions
• Practice: Receipt

Next time: Control flow in Java
• **precedence**: Order in which operators are evaluated.
  
  – Generally operators evaluate left-to-right.
    
    \[ 1 - 2 - 3 \text{ is } (1 - 2) - 3 \text{ which is } -4 \]
    
    – But \( * / \) have a higher level of precedence than \( + - \)
      
      \[ 1 + 3 * 4 \text{ is } 13 \]
      
      \[ 6 + 8 / 2 * 3 \]
      
      \[ 6 + 4 * 3 \]
      
      \[ 6 + 12 \text{ is } 18 \]
      
  – Parentheses can alter order of evaluation, but spacing does not:
    
    \[ (1 + 3) * 4 \text{ is } 16 \]
    
    \[ 1+3 * 4-2 \text{ is } 11 \]
[Extra] String concatenation

• **string concatenation**: Using + between a string and another value to make a longer string.

  - "hello" + 42 is "hello42"
  - 1 + "abc" + 2 is "1abc2"
  - "abc" + 1 + 2 is "abc12"
  - 1 + 2 + "abc" is "3abc"
  - "abc" + 9 * 3 is "abc27"
  - "1" + 1 is "11"
  - 4 - 1 + "abc" is "3abc"

• Use + to print a string and an expression's value together.

  ```java
  println("Average: " + (95.1 + 71.9) / 2);
  ```

  Output:   Average: 83.5
• 5 + 3 / 2 - 4
• 15 / 2.0 + 6
• 1 * 2 + 3 * 5 % 4
• "abc" + 1 + 2
• "abc" + (1 + 2)
[Extra] Practice

• 5 + 3 / 2 - 4 = 2
• 15 / 2.0 + 6 = 13.5
• 1 * 2 + 3 * 5 % 4 = 11
• “abc” + 1 + 2 = "abc12"
• “abc” + (1 + 2) = "abc3"
[Extra] Precedence examples

1 * 2 + 3 * 5 % 4

“abc” + 1 + 2
[Extra] Precedence examples

1 * 2 + 3 * 5 % 4

“abc” + 1 + 2

\[
\begin{align*}
1 & \times 2 + 3 \times 5 \mod 4 \\
& \downarrow \\
2 & + 3 \times 5 \mod 4 \\
& \downarrow \\
2 & + 15 \mod 4 \\
& \downarrow \\
2 & + 3 \\
& \downarrow \\
5 &
\end{align*}
\]
[Extra] Precedence examples

1 * 2 + 3 * 5 % 4
“abc” + 1 + 2

“abc1” + 2

“abc12”
public class HelloWorld extends ConsoleProgram {
    public void run() {
        print("Hello, ");
        print("world!");
    }
}

Same as println, but does not go to the next line.
• **escape sequence**: A special sequence of characters used to represent certain special characters in a string.

  \t tab character
  \n new line character
  " quotation mark character
  \ backslash character

– **Example:**

```java
println("\\hello\\nhow\\tare "\"you\\"?\\\\");
```

– **Output:**

```
\hello
how are "you"?
```