CMPINF0401 Recitation

TUESDAYS 11:00-12:50

MICHAEL BARTLETT

Overview

- Datatypes
- Manipulating Strings
- Arithmetic
- User Input
- Lab 2

Datatypes

- Primitives vs. objects
  - Primitive examples: boolean, byte, short, long, int, double, float, char (all lowercase!)
    - boolean: true/false value
    - char: character (‘a’.,‘x’)
    - short, long, int, double, float, and byte are all numbers of varying lengths
      - Most of the time, you’ll use int and double of these six
      - Only double and float can be decimal values
  - Object examples: String, Scanner, and many… many more (capital first letter!)
- Side note: You always need to put the datatype before the name of your variable
  - i.e.) String myName = Michael;

Strings

- We can make a new String (think: array of char) like this:
  - String myName = “Michael”;
- Strings are immutable!
  - This means that they cannot be change once they’re created
  - Therefore, whenever you want to do something to a string, you have to call a method that is a part of the String class in Java.
  - i.e.) Make myName all uppercase:
    - myName = myName.toUpperCase();

Manipulating Strings

- More common operations are:
  - `str.toUpperCase();` → returns `str` in all uppercase letters
  - `str.toLowerCase();` → returns `str` in all lowercase letters
  - `str.charAt(int num);` → returns a `char` at index `num` in `str`
  - `str.length();` → returns an `int` giving the length of `str`
- As seen on the previous slide, we have to set our String equal to the method call in order to update its value since all these methods are **returning** a String.
- You can concatenate Strings also to form one string using the "+" operator
  - String firstName = "Michael";
  - String lastName = "Bartlett";
  - String fullName = firstName + " " + lastname;

String Methods

```
String str = "Coding is FUN!";
String upper = str.toUpperCase();
String lower = str.toLowerCase();
char secondLetter = str.charAt(1);
int strLength = str.length();
```

```
Coding is FUN!
CODING IS FUN!
coding is fun!
o
14
```

- Note: string indexing starts at 0!
  - `str.charAt(0);` would return ‘C’ while `str.charAt(14);` gives an error

Manipulating Numeric Data: Operations and Operator Precedence

Operator	Purpose	Example	Equivalent
+=	Addition	x += 2	x = x + 2
-=	Subtraction	x -= 2	x = x - 2
/=	Division	x /= 2	x = x / 2
*=	Multiplication	x *= 2	x = x * 2
%=	Modulus	x %= 2	x = x % 2

- There are shorthand ways of using these operations!

More Operations and Operator Precedence

Level	Operator	Description	Associativity

16	`[]`
	`.`
	`()`
		access array element
		access object member
		parentheses
			left to right

15	`++` `--`	unary post-increment unary post-decrement	not associative

14	`++`
	`--`
	`+`
	`-`
	`!`
	`~`
		unary pre-increment
		unary pre-decrement
		unary plus
		unary minus
		unary logical NOT
		unary bitwise NOT
			right to left

13	`()` `new`	cast object creation	right to left
12	`* / %`	multiplicative	left to right
11	`+ -` `+`	additive string concatenation	left to right
10	`<< >>` `>>>`	shift	left to right

9	`< <=`
	`> >=`
	`instanceof`
		relational	not associative

8	`==` `!=`	equality	left to right
7	`&`	bitwise AND	left to right
6	`^`	bitwise XOR	left to right
5	`|`	bitwise OR	left to right
4	`&&`	logical AND	left to right
3	`||`	logical OR	left to right
2	`?:`	ternary	right to left

A Quick Note About Integer Division

- Integer division and floating point (decimal) division are different!
- If you're dividing two integers, you will end up with a whole number (`int`). Otherwise, you'll end up with a decimal (`double`)

```
public static void main(String[] args) {
  System.out.println(10/3);
  System.out.println(10.0/3);
  System.out.println(10/3.0);
  System.out.println(10.0/3.0);
}
```

A Quick Note About Integer Division

```java
public static void main(String[] args) {
  System.out.println(10/3);
  System.out.println(10.0/3);
  System.out.println(10/3.0);
  System.out.println(10.0/3.0);
}
```

```
3
3.3333333333333335
3.3333333333333335
3.3333333333333335
```

Getting User Input

- To get user input we need to use a Scanner.
  - To initialize a Scanner, we make a variable to "hold" our scanner and then make a new Scanner using System.in
    - Scanner keyboard = new Scanner(System.in)
  - Once you have a Scanner, you can get input from the user and set the input equal to a variable:
    - String name = keyboard.next() // .next takes the next token as a String
    - int num = keyboard.nextInt() // Takes the user input as an int
      - If an int isn't entered the program will crash since it's expecting an int. This exception could be handled though.
    - More methods: https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html

Lab 2

- Assignment:
  - https://canvas.pitt.edu/courses/127916/files/8050342?module_item_id=2735237
- You only need to write two lines of code for this lab:
  - One to set userName equal to the part of the input
  - Another to set income equal to the other part of the input
- Hint: When using Scanner methods, you can have the user enter multiple tokens at once and then separate them by calling the correct method:
  - i.e.) We have a Scanner called keyboard and we want to get two Strings from it for separate variables
    - stringOne = keyboard.next()
      - .next() only takes in one token at a time so if a user entered two strings with a space in between, this will only read the first one. .nextLine will read in the entire line
    - stringTwo = keyboard.next()
  - Use this hint for your lab where you need to set userName (String) and income (double) from one prompt for the user.