CSE 11
Accelerated Intro to Programming
Lecture 5

Greg Miranda, Spring 2021

This lecture is being recorded

Announcements

- Quiz 5 due Friday @ 8am
- PA1 due tonight @ 11:59pm → PA1 resubmission
- Survey 2 due Friday @ 11:59pm

Discussion 4pm 5pm

↳ PA2

- String example program

```
class StringExamples {
}
```

- Write a method called firstHalf that:
  - Takes a String and returns a new String that has just the first half of the characters from the input String

(round down)

- When writing a method:
  - Think about what some examples are and what we expect the results to be:
    - We can write these down as fields
    - Then we can easily check if we are right after running the program
  - Examples first – then build up into the implementation
    - Do on paper/whiteboard first – then type them in

Argument	Result
"banana"	"ban"
"hello"	"he" or "hel"
"a"	? "" ↓ empty string

- One of the first things to think about is:
  - What method (or methods) out of the methods we saw on strings is going to be useful here
    - We will be able to accomplish this only with methods we have seen so far

"banana" → "ban"

→ length

→ substring(int s, int e)

0 ?

indexOf

replace

repeat

"a" → ?

- This showed us how to implement a method from a word problem prompt
- We thought through some examples
  - Which helped us to refine our understanding
- We experimented a little bit
  - Figured out we are okay with this empty String result
- This is the process we should use when implementing methods
  - i.e. Programming Assignments

↑

unit testing

examples ⇒ expected results

write implementation

New Data Type

- Previous data types:
  - String
  - int
- Examples
  - boolean b1 = 4 < 5;
  - boolean b2 = 5 < 4;
- New data type:
  - Boolean → boolean
    - Uses different kinds of operators
      - Comparison operators

- String – many different types of strings, infinite # of strings
    - Only limited by how much memory is in our computer
- int – somewhat limited 2^{32}
    - -2,147,483,648 to 2,147,483,647
- Boolean – only two values
    - true / false
        - Represents the answers to yes or no questions
            - 4 < 5
                - Asking the question: is 4 less than 5?

true

String	int	boolean
"a"	-2	true
"Apple"	-1	false
⋮	0	
	1	
	2	
	⋮	

arbitrary # of elements

just 2 values

==

- Many boolean operators
  - boolean b3 = 4 == 4;   //checks for equality
  - boolean b4 = 4 == 5;
    - = is not the same as ==
      - = is used to create or initialize a field definition
        - Assignment operator
  - boolean b5 = 5 ≥ 4;
  - boolean b6 = 5 >= 4;
    - As well as <=
- All of these are ways to compare numbers
  - Gives true/false (yes or no) answers
- What happens if we use it to compare Strings?
  - boolean stringComp = “a” < “b”;

- Useful idea when learning a new feature
  - Ask if it works with other things you've worked with before
- Comparison operators like < and > do not type check
  - Only numeric types work with Java's ~~type checking~~
- What about == on Strings?
  - boolean stringComp = "a" == "a";
    - == does produce an answer on Strings
  - boolean stringComp = "a" == "b";
  - Does produce an answer, but not recommended for Strings
  - We will talk more about comparing Strings for equality in future weeks
  - Only use == for numeric comparisons in this course

int

- Main lesson:
  - 2 new values
    - true/false
  - With new data type boolean
  - New relational and comparison operators that work with booleans

    r c

    r
    - < >
    - <= >=

    c
    - ==
  - Another comparison operator
    - !=
    - boolean b7 = 4 != 5;
    - boolean b8 = 5 != 5;
    - Opposite of the == answer

! Not

!= not equal

Boolean Operations

- What can you do given a boolean?
  - What if we want to ask more than simple questions
    - Are two things true at the same time?
    - Is one of two things true?

- Combining booleans into another boolean
  - boolean and1 = true && true; //true
  - boolean and2 = true && false;
  - boolean and3 = false && true;
  - boolean and4 = false && false;

&& and

both sides must be true

  - boolean or1 = true || true; // true
  - boolean or2 = true || false; // true
  - boolean or3 = false || true; // true
  - boolean or4 = false || false;

|| or

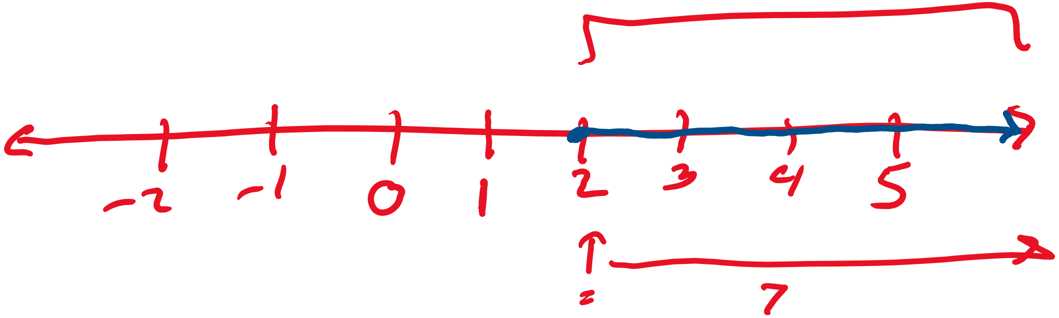
only 1 side must be true

N >= 2

Methods with Booleans

2 →

- Number line 1



- Problem:
  - Write a method that takes a number and returns true if it's in the region in our number line example
  - Examples:

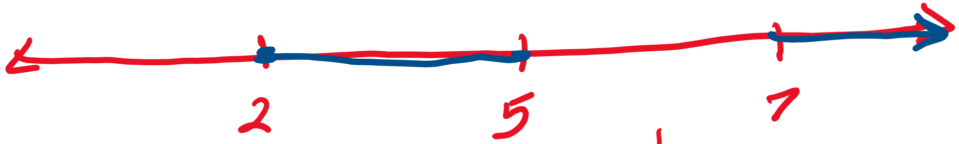
Argument	Expect
2	true
5	true
-1	False

```
boolean numberLine2(int number) {
  return (number > 5) && (number < 7) || (number < 2);
}
```

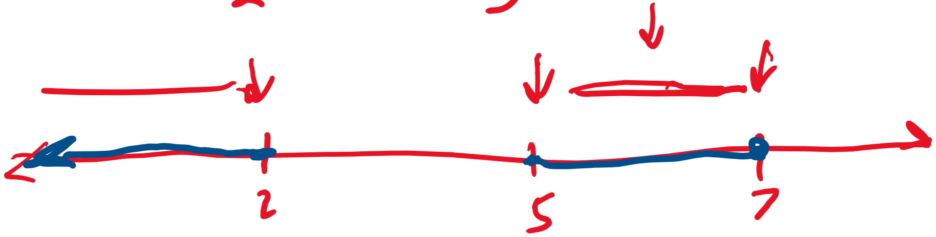
- What does this number line look like?

&& before ||

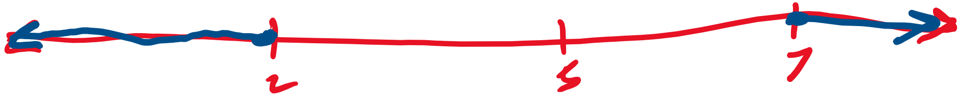
A



16 B



C



More Complicated Questions with Methods

- Write the method to calculate absolute value that takes a number and returns the negation if it's less than 0, or that number otherwise
- Examples:
  - int abs1 = this.absolute(-2);   //should produce 2
  - int abs2 = this.absolute(4);   //should produce 4

```
int absolute(int number) {

}
```

- Important comparison we need to do here
  - Is the number less than 0?
    - number < 0
  - Don't want to return true or false, we want to return the right number
- New Java syntax:
  - if statement

stop

Weekly Pay Problem

- weeklyPay: takes a number of hours worked and an hourly rate, and returns the pay with overtime (over 40 hours) counting as double the rate
- Examples: