Lesson 4-2: Strings and I/O

Computer Science 46A: Introduction to Programming

San José State University

Announcements

- Homework #4 is posted
- Lab 4 is Friday (2/21)
  - Please show up on time! Lab Instructors report lots of late attendees
- Office hours cancelled today

Learning Objectives

By the end of this lesson, you should be able to:

1. Use a Scanner methods to take in String inputs from the user
2. Use String methods to partition Strings
3. Format numbers in strings with different levels of numerical precision

Strings with Scanner Objects

Strings with Scanner objects

- Recall: we create Scanner objects the same we use other classes:

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

- To ask for a String using a Scanner object we use `nextLine()`:

```
String strInput = myObj.nextLine();
```

- To ask for just one word of a String using a Scanner object we use `next()`:

```
String wordInput = myObj.next();
```

Example: ScannerStrings

Compare and contrast:

- The nextLine method will read in the entire line provided
- The next method will just read in the first word and save the rest of the words for subsequent scanner method calls

```
This is a line
nextLine method: This is a line
word
word method: word
```

Output of ScannerStrings for a line and a word

Poll Everywhere

What is the output of `ScannerStrings` when we provide the following two inputs:

```
This is a line
This is a line
```

A)
```
This is a line
  This is a line
```

B)
```
This is a line
   This
```

C) This is a line
```
java.util.InputMismatchException
```

PollEv.com/narayanbalasubramanian644

Poll Everywhere

What is the output of `ScannerStrings` when we swap the order of the nextLine and next methods and provide the following two inputs:

```
This
This is a line
```

A)
```
This
 This is a line
```

B)
```
This
```

C)
```
This
java.util.InputMismatchException
```

PollEv.com/narayanbalasubramanian644

More Strings Methods!
(and a couple formatting tips)

Quick Tip: Formatting Strings with Quotes

- All Strings start and end with a double quotation mark “”
  - What happens if you want to have a quotation mark in your String?
- To use quotes in your String, you need to use the escape character
  - In Java, the escape character is a backslash (\)
- Example:

```
System.out.print(“ This \“word\” has quotes”)
Output: This “word” has quotes
```

- Recall from Homework 1: We also used this same approach when using Unicode in Strings (e.g. \u00f1 printed the character ñ)

Quick Tip: Formatting Strings with Lines

- In previous classes, we've used the `println` function to start the next output on a new line
- We can also use the escape character (\) with the letter n to start a new line
- Example:

```
System.out.println("Line 1")
System.out.print("Line 2")
```

is the same as:

```
System.out.print("Line 1\nLine 2")
```

Recall from Lecture 2-2: Five Common `String` Methods

- `public int length()`
- `public String toUpperCase()`
- `public String toLowerCase()`
- `public String replace(char oldChar, char newChar)`
- `public String replace(String oldString, String newString)`
- `public char charAt(int index)`

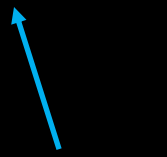
Today:
Four More Common `String` Methods

- `public String trim()`
- `public boolean contains(String str)`
- `public String substring(int beginIndex)`
- `public String substring(int beginIndex, int endIndex)`
- `public int indexOf(char ch)`
- `public int indexOf(char ch, int fromIndex)`
- `public int indexOf(String str)`
- `public int indexOf(String str, int fromIndex)`

String Method: trim()

- The trim() method is used to remove “white space” at the beginning and end of a string
  - White space refers to characters such as tabs and spaces which do not contain any visible characters
- Example:

```
String example = “  This line has white space    ”
System.out.print(example.trim());
```



White space

- Output:

```
“This line has white space”
```

See example `TrimContains` in lec04-2b

String Method: contains()

- The contains() method can be used to tell you whether or not a string contains another string
- Example:

```
String smartPeople = “Einstein, Newton, Tyson”
System.out.println(smartPeople.contains(“Tyson”));
System.out.print(smartPeople.contains(“Wood”));
```

- Output:

```
true
false
```

Output is boolean!

See example `TrimContains` in lec04-2b

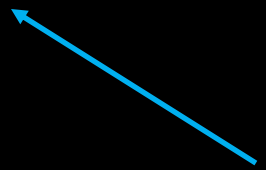
String Method: substring()

- The substring method can be used to output a subset of a given string
- The substring method is *overloaded*
- Method 1: Pass beginning and ending indices

```
substring(int beginIndex, int endIndex)
```

- Method 2: Pass only a beginning index

```
substring(int beginIndex)
```



When only one input argument is given, the endIndex is the end of the string

See example `SubStrProg` in lec04-2b

Poll Everywhere

```
String statement = “Today is Thursday”;
```

Which of the following returns “Thur”

```
A) statement.substring(7, 11)
B) statement.substring(8, 12)
C) statement.substring(9, 13)
D) statement.substring(10, 14)
```

PollEv.com/narayanbalasubramanian644

Java Indexing Starts from 0

```
String statement = “Today is Thursday”;
```

T	o	d	a	y		i	s		T	h	u	r	s	d	a	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Remember that in Java, we start counting indices with the number 0

String Method: indexOf()

- The indexOf method returns where a string or character is located in a given string
- The indexOf method is *overloaded* with 4 different options
- Method 1: Pass a string or a character

  `indexOf(String str)` or `indexOf(char ch)`

- Method 2: Pass a string or a character, AND a “from” index

  `indexOf(String str, int fromIndex)` or

  `indexOf(char ch, int fromIndex)`

When the fromIndex is given, we only search the string after this index

See example `IndexOfAndFirstWord` in lec04-2b

Using String Methods Together: Finding the First Word

- We can use String methods in combination to achieve a desired effect
- Example: Return the first word in a String

```
String example = “First Word”;
```

  - Step 1: Find the index of the first space using the `indexOf` method

```
int firstSpaceIndex = example.indexOf(‘ ’);
```

  - Step 2: Use the index in the substring method to get the first word:

```
String firstWord = example.substring(0,firstSpaceIndex);
```

See example `IndexOfAndFirstWord` in lec04-2b

Formatting String Output With Numbers

Formatting Strings with ints

- In previous lessons, we have used the + operator to mix `Strings`, `ints`, and `doubles`, in `System.out.print()` statements
- Example:

```
int myInteger = 5;
System.out.print(“My number is ” + myInteger + “.”)
```

- We can alternatively use the printf function for formatting output
- Syntax for integers is with %d:

```
System.out.printf(“%d”,[int variable])
```

- Example above re-written with `printf`:

```
System.out.printf(“My number is %d.”, myInteger)
```

Formatting Strings with doubles

- Recall: double type numbers contain approximately 15 digits
- Often, we only care about the first few digits
- We can use the printf function to round output for us
- Syntax for doubles is %f with x representing the number of decimals:

```
System.out.printf("%.xf",[double variable])
```

- Example:

```
System.out.printf("%.2f",3.1415)
Output: 3.14
```

See example `FormattedOutput` in lec04-2c

Participation Exercise 4-2a: StringProg.java

Goal: Write a program to practice new string methods

```
Enter multiple words on one line separated by single spaces:    This is a test
Enter a single word: a
The original line: "   This is a test  "
The original word: "a"
The line with spaces trimmed from both ends: "This is a test"
The word with spaces trimmed from both ends: "a"
The line contains the word: true
The index of word in line: 8
First word of line: "This"
Second word of line: "is"
```

Output of StringProg.java

Codecheck Link: HERE and on Canvas

Participation Exercise 4-2b: DiscountPrice.java

Goal: Write a program to calculate the discount price on an item

```
Enter the product name: iPhone 11
Enter the price: 749.99
Enter a discount percentage: 0.1
The price of iPhone 11 is $749.99
The discount price of iPhone 11 is $674.99
```

Output of DiscountPrice.java

Codecheck Link: HERE and on Canvas