Strings and Loops

CSE 5910

moodle.yorku.ca

Strings

Strings are immutable objects.

The state of an immutable object cannot be changed.

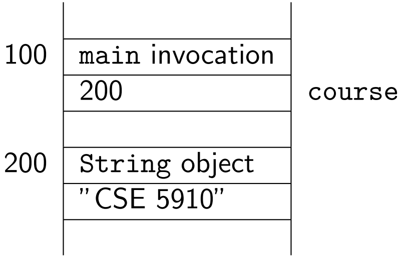
The `String` API does not contain any mutators.

The `StringBuffer` class provides mutable strings. [1]

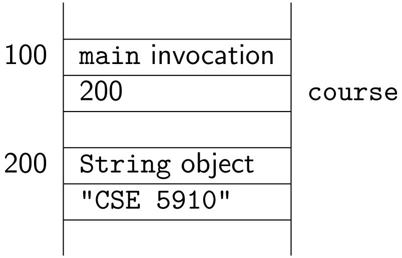
[1] We will come back to the `StringBuffer` class later.

Strings

$$\text{String course} = \text{new String("CSE 5910");}$$



Strings



String reference: `course`
String object: object at address 200
String literal: `"CSE 5910"`

Strings are everywhere

Instead of

```
String course = new String("CSE 5910");
```

we are allowed to write

```
String course = "CSE 5910";
```

Although in most cases you may think of `"CSE 5910"` and `new String("CSE 5910")` as synonyms, they are not always equivalent.[2]

[2]Hardly ever will this difference impact your app.

Strings are immutable

According to the Java Language Specification,

> *Strings that are the values of constant expressions are "interned" so as to share unique instances*

James Gosling, Bill Joy, Guy L. Steele Jr. and Gilad Bracha. The Java Language Specification. Third edition. Addison-Wesley. 2005.

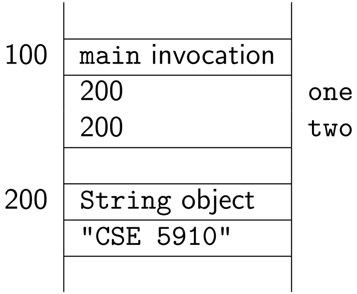
Strings are immutable

> *Strings that are the values of constant expressions are "interned" so as to share unique instances*

These constant expressions are built from String literals and the binary operator $+$.

Strings are immutable

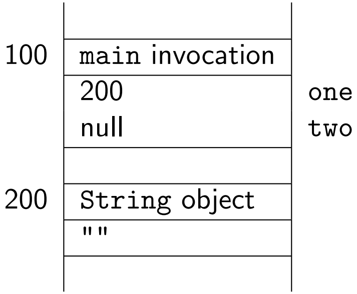
String one = "CSE 5910";
String two = "CSE" + " " + "5910";



This saves memory. Why can one and two refer to the same String object?

The empty string versus `null`

String one = "";
String two = null;



My mom . . .

. . . tends to write long sentences. Long sentences are in general more difficult to comprehend.

Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3): 221-233, June 1948.

Long sentences

Problem

Prompt the user for a file name by printing

```
    Enter file name:
```

so that the file name is entered by the user on the same line as the prompt.

You may assume that the file consists of sentences.

Print the last word of every sentence, each on a separate line.

Long sentences

```
String line = ...
Scanner lineInput = new Scanner(line);
while (lineInput.hasNext())
{
    String token = lineInput.next();
    ...
}
```

Long sentences

```
String line = ...
StringTokenizer tokenizer = new StringTokenizer(line);
while (tokenizer.hasMoreTokens())
{
    String token = tokenizer.nextToken();
     ...
}
```

Long sentences

Problem

Prompt the user for a file name by printing

```
    Enter file name:
```

so that the file name is entered by the user on the same line as the prompt.

You may assume that the file consists of sentences.

Print the number of words for every sentence, each on a separate line.

Long sentences

Problem

Prompt the user for a file name by printing

```
    Enter file name:
```

so that the file name is entered by the user on the same line as the prompt.

You may assume that the file consists of sentences.

Print those sentences that have more than 35 words, each on a separate line.[a]

[a]The New Yorker of October 26, 1946 has on average 20 words per sentence.

The StringBuffer class

StringBuffers are mutable objects.

The method

```
    public StringBuffer append(String s)
```

adds the String `s` to the end of the StringBuffer.

The method returns a reference to the StringBuffer itself. Although this is not needed (why?), it is convenient.

From `String` to `StringBuffer` and back

$$\begin{tikzcd} \texttt{String} \arrow[r, bend left, "\texttt{new StringBuffer}(\cdot)"] & \texttt{StringBuffer} \arrow[l, bend left, "\cdot\texttt{.toString()}"] \end{tikzcd}$$

Command-line arguments

Question

How does the user provide command-line arguments?

Command-line arguments

Question

How does the user provide command-line arguments?

Answer

```
java LongSentences "book.txt"
```

Command-line arguments

Question

How does the user provide command-line arguments?

Answer

```
java LongSentences "book.txt"
```

Question

How does the client get the command-line arguments?

Command-line arguments

Question

How does the user provide command-line arguments?

Answer

```
java LongSentences "book.txt"
```

Question

How does the client get the command-line arguments?

Answer

As the parameter of the `main` method.

Command-line arguments

Question

How does the user provide command-line arguments?

Answer

```
java LongSentences "book.txt"
```

Question

How does the client get the command-line arguments?

Answer

As the parameter of the `main` method.

Question

What is the type of the parameter of the `main` method.

Command-line arguments

Question

How does the user provide command-line arguments?

Answer

```
java LongSentences "book.txt"
```

Question

How does the client get the command-line arguments?

Answer

As the parameter of the `main` method.

Question

What is the type of the parameter of the `main` method.

Answer

`String[]`: an array of `String`s.

Command-line arguments

```
public static void main(String[] args)
```

Question

How does the client get the first command-line argument?

Command-line arguments

public static void main(String[] args)

Question

How does the client get the first command-line argument?

Answer

`args[0]`

Command-line arguments

public static void main(String[] args)

Question

How does the client get the first command-line argument?

Answer

`args[0]`

Question

How does the client get the second command-line argument?

Command-line arguments

public static void main(String[] args)

Question

How does the client get the first command-line argument?

Answer

`args[0]`

Question

How does the client get the second command-line argument?

Answer

`args[1]`

Command-line arguments

```
public static void main(String[] args)
```

Question

How does the client get the number of command-line arguments?

Command-line arguments

```
public static void main(String[] args)
```

Question

How does the client get the number of command-line arguments?

Answer

`args.length`

Command-line arguments

Problem

The file name is provided as a command-line argument.

You may assume that the file consists of sentences.

Print those sentences that have more than 35 words, each on a separate line.

If the user does not provide a command-line argument, print

```
    Use:  java LongSentences <file name>
```

sochi.ru
2014

source: http://www.sochi2014.com

Medal standing

Problem

Print the HTML of the web page of the Sochi Olympics containing the medal standings.

HTML

HTML is short for HyperText Markup Language. It classifies the information of a web page. A web browser uses HTML, together with some other data such as Cascading Style Sheets (CSS), to present the information.

```
<html>
<head>
<title>My page</title>
</head>
<body>
<p>Welcome to my page.</p>
</body>
</html>
```

HTML5 (in combination with CSS3) is Turing complete.

HTML was proposed by Tim Berners-Lee in 1990.

Tim Berners-Lee

Sir Timothy Berners-Lee (born 8 June 1955), is a British computer scientist. He is best known as the inventor of the World Wide Web. He is a professor at MIT.

source: Knight Foundation

Medal standing

Problem

Print those lines of the HTML of the web page of the Sochi Olympics that represent the medal table.

Medal standing

Problem

Prompt the user "Enter the name of a country: " and print "has won g gold, s silver and b bronze medals."

Medal standing

Problem

Prompt the user "Enter the name of a country: " and print "has won g gold, s silver and b bronze medals." Reprompt the user if the country cannot be found in the medal table.

Medal standing

Problem

Make the app more robust by allowing the user to use lower- and uppercase when entering the country's name. For example, Canada, canada, CaNaDa, etc should all be acceptable.

Medal standing

Exercise

Make the app more robust by trying to correct typos. In particular, try to correct the situation where the user did not type one of the characters. For example, Canaa, Canad, etc should all be acceptable.

Screen Scraping

Reading the HTML from a URL and extracting information from it is an example of screen scraping.

A robust alternative to screen scraping is web services.

Reading week

- Next week is reading week.
- There will be no lecture.
- There will be no office hours.