Print a file

Problem

Prompt the user for a file name

```
    Enter a file name:
```

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

Compile Time Error

Error message: unreported exception BlaBlaBlaException; must be caught or declared to be thrown

Compile Time Error

Error message: unreported exception BlaBlaBlaException; must be caught or declared to be thrown

Quick fix: add throws BlaBlaBlaException to the header of the main method

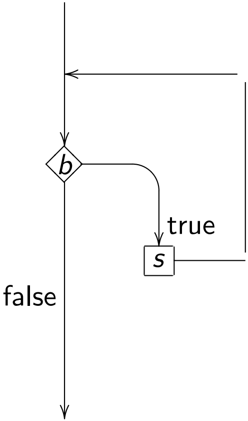
Compile Time Error

Error message: unreported exception BlaBlaBlaException; must be caught or declared to be thrown

Quick fix: add throws BlaBlaBlaException to the header of the main method

Proper solution: will be discussed in Chapter 11

While statement



While statement

Syntax

```
while (b)
{
    s;
}
```

Code conventions:

- while should be followed by a space and
- the body should be indented.

For and while loops

Theorem

Every for-loop can be expressed as a while-loop.

Proof.

```
for ($s_1$; $b$; $s_2$)
{
    $s_3$;
}
```

can be expressed as

```
{
    $s_1$;
    while ($b$)
    {
        $s_3$;
        $s_2$;
    }
}
```

For and while loops

Theorem

Every while-loop can be expressed as a for-loop.

Print a triangle

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, . . . , a line with $n-1$ $*$'s, and a line with n $*$'s.

Reprompt

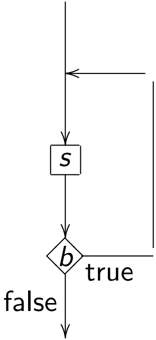
Problem

Prompt the user for a positive integer

```
    Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, . . . , a line with $n - 1$ $*$'s, and a line with n $*$'s. Reprompt the user if they enter a non-positive integer.

Do statement



Do statement

Syntax

```
do
{
    s;
}
while (b);
```

Code conventions:

- while should be followed by a space and
- the body should be indented.

For and do Loops

Theorem

Every for-loop can be expressed as a do-loop.

Theorem

Every do-loop can be expressed as a for-loop.

Question

So which loop should we use?

For and do Loops

Theorem

Every for-loop can be expressed as a do-loop.

Theorem

Every do-loop can be expressed as a for-loop.

Question

So which loop should we use?

Answer

It is a matter of taste. If you know the number of iterations in advance, a for-loop may be most appropriate. If the loop has to be executed at least once, a do-loop may be most appropriate.

Prime

Exercise

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print

n `is prime`

if n is prime and

n `is not prime`

otherwise.

Prime

The New York Times

New Method Said to Solve Key Problem in Math

By SARAH ROBINSON

Three Indian computer scientists have solved a longstanding mathematics problem by devising a way for a computer to tell quickly and definitively whether a number is prime – that is, whether it is evenly divisible only by itself and 1.

New York Times, August 8, 2002

Strings and Loops

CSE 5910

www.eecs.yorku.ca/course/5910

Strings

Strings are immutable objects.

The state of an immutable object cannot be changed.

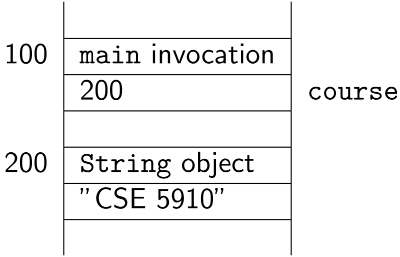
The `String` API does not contain any mutators.

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

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

Strings

String course = new String(”CSE 5910”);



100

main invocation

200

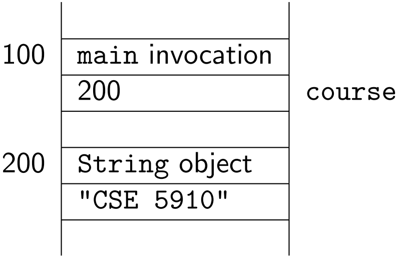
course

200

String object

”CSE 5910”

Strings



100 main invocation

200 course

200 String object

"CSE 5910"

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.

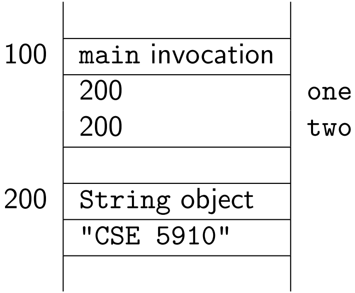
Strings are immutable

> *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

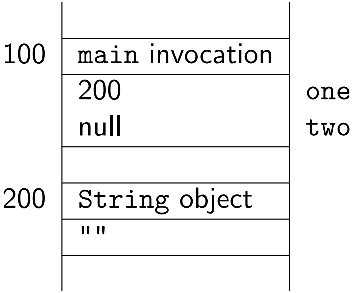
String one = ”CSE 5910”;
String two = ”CSE” + ” ” + ”5910”;



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

The empty string versus null

String one = ””;
String two = null;



If only I had known . . .

Problem

Print, for example,

```
If I had bought ibm shares on 01/15/68
and sold them on 01/16/93,
I would have made a 1599.88% loss
```

where `ibm` is provided as a command line argument.

Let's start with something simpler

Problem

Print the first line of the file gts.csv.

Files with extension .csv usually contain comma separated values.

Let's start with something simpler

Problem

Print all but the first line of the file gts.csv.

Let’s start with something simpler

Problem

Print all but the first line of the file gts.csv, where each value is separated by a tab instead of a comma.

replaceAll method

```
public String replaceAll(String pattern,
                         String replacement)
```

Replaces each substring of this string that matches the given pattern with the given replacement.

StringTokenizer class

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

Let’s start with something simpler

Problem

For all but the first line of the file gts.csv, convert the first value to a `Date` object and the second and third value to values of type `double`.

SimpleDateFormat class

```
public SimpleDateFormat(String pattern)
```

Initializes this SimpleDateFormat using the given pattern.

```
public Date parse(String source) throws ParseException
```

Parses text the given string to produce a date.