Java Programming 2 Variables, identifiers, scope

Mary Ellen Foster

MaryEllen.Foster@glasgow.ac.uk

Semester 1 2020/2021

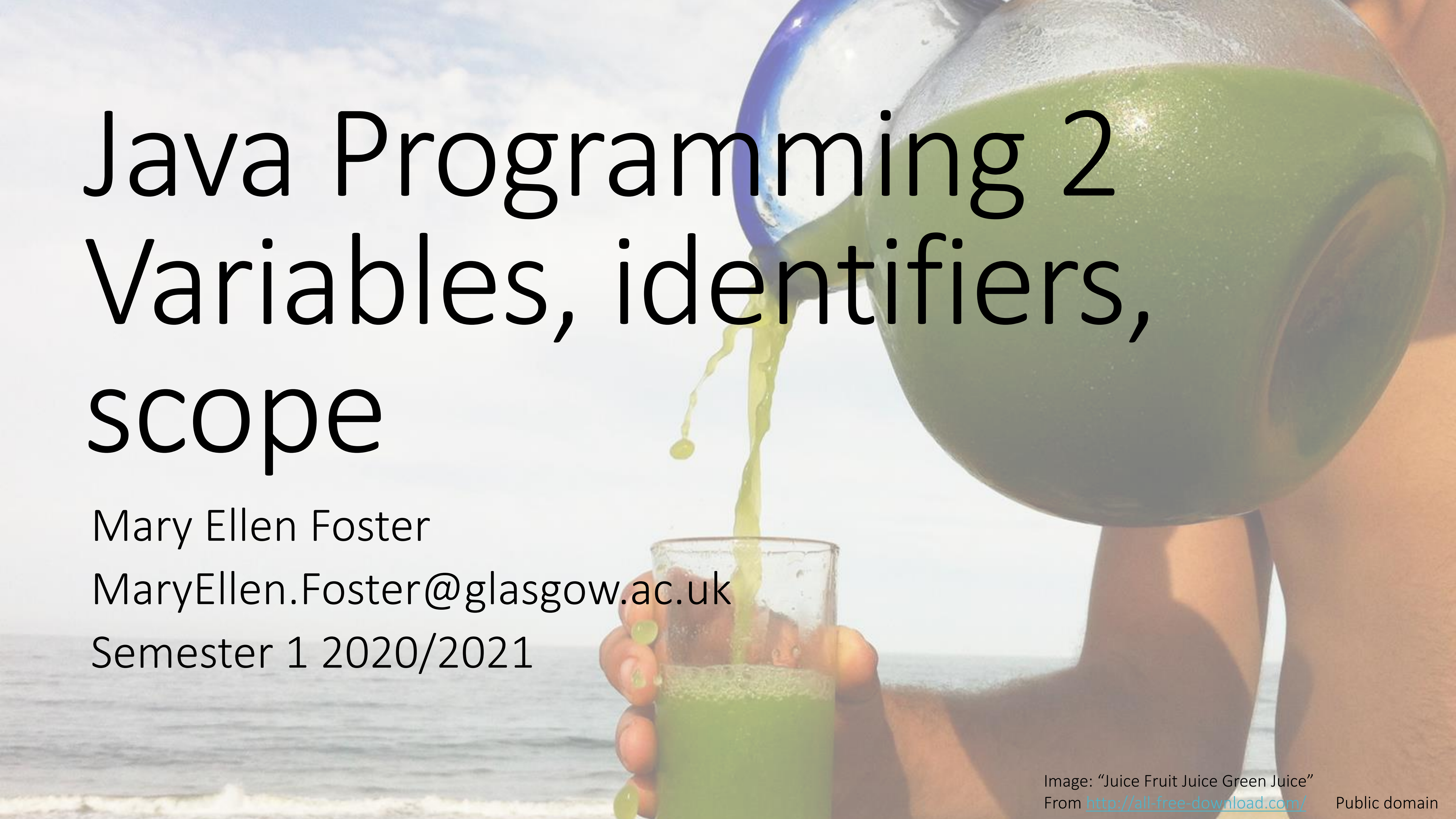


Image: “Juice Fruit Juice Green Juice”
From http://all-free-download.com/ Public domain

What is a variable?*

A **container** that stores information

Useful for the **computer**: data can be referenced and manipulated in the program

Useful for the **programmer**: Allows you to label data with a descriptive name to make it readable

*loosely adapted from https://launchschool.com/books/ruby/read/variables

Variables in Java

Unlike Python, Java is **statically typed**

So types of all variables must be declared

- Local variables:

```
int i;
```

- Class fields:

```
class C { boolean b; }
```

- Method parameter and return values

```
public float getValue (long l) { … }
```

Declaring and initialising variables

Declaration: setting aside a memory location for a variable of this type, and associating the name with the memory location

Initialisation: giving a variable its initial value

Variable can be declared and initialised in one statement, or separately

```
int i;
i = 5;
int j = 4, k = 6, m;
```

Variable must have a value before it is used

```
boolean b;
System.out.println (”Value of b: ” + b);
```

Error! (OK in jshell though)

Value can be changed after initialisation

… but must have the correct type

Comparing Python and Java (from quiz)

JAVA

```
int a;
a = 1;
a = 10.5;
System.out.println (a);
```

PYTHON

```
gap = 3
gap = “bye”
print (gap)
```

Static typing in practice

Variables can only be given values that are compatible with their type

```
int i;
i = 5;
i = 5.0;
i = “hello”
```

Variables cannot be redeclared with a different type

```
String s = “hello”;
int s;
```

Identifier names in Java

An **identifier** is the label for a named Java entity

- Class, field, method, parameter, local variable, …

There are both **rules** and **conventions** for identifiers

- Rules: follow them or the *compiler* will be unhappy
- Conventions: follow them or ***your tutors and fellow programmers*** will be unhappy

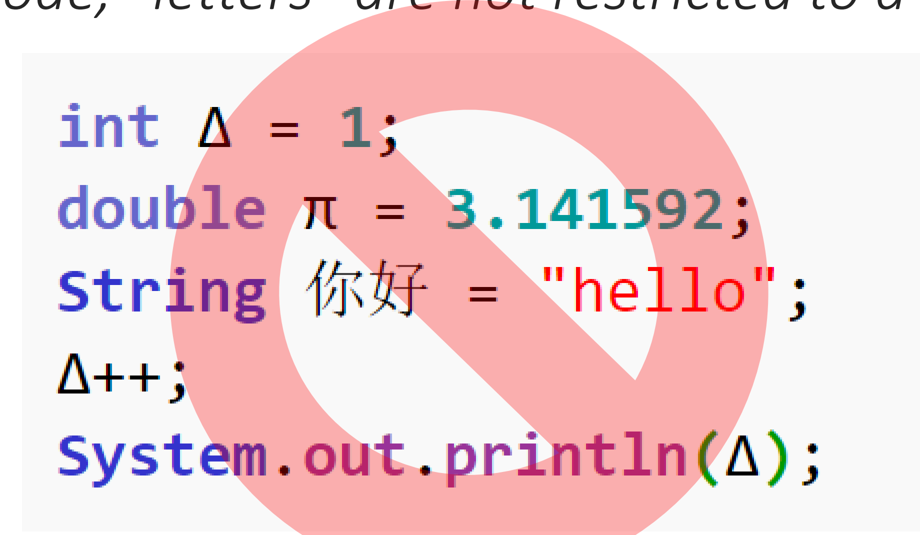
Identifier rules

Must begin with a letter or an underscore (“_”)

Must be at least one character long

Other characters may be letters, numbers, or underscores

Since Java supports Unicode, “letters” are not restricted to a-z and A-Z



```
int Δ = 1;
double π = 3.141592;
String 你好 = "hello";
Δ++;
System.out.println(Δ);
```

Please do not do this!

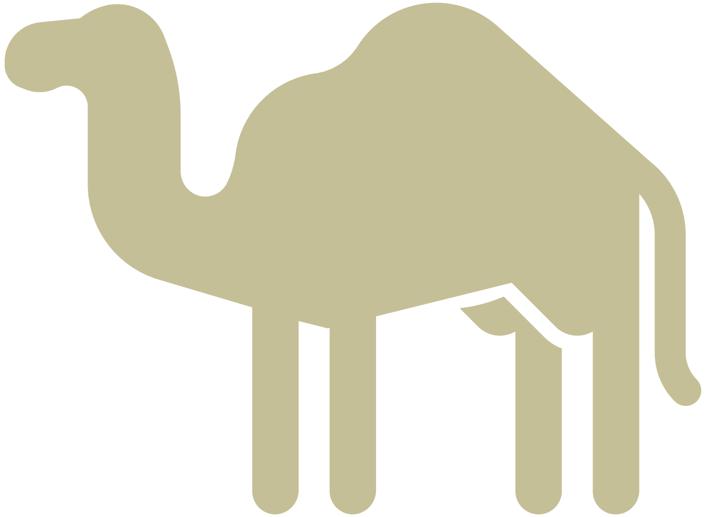
http://rosettacode.org/wiki/Unicode_variable_names#Java

Identifier conventions

Standards and guidelines can differ on the details

Near-universal conventions

- Classes start with a capital letter
- Other identifiers start with a lower-case letter
- Multi-word identifiers use “CamelCase”
  - *`ArrayIndexOutOfBoundsException`*
- Constant values are written in ALL CAPS
  - *`Math.PI`*, *`Double.POSITIVE_INFINITY`*



Sharing the same name

- No two local variables or method parameters can have the same name in a given scope
  - Local variables can have the same name as method parameters (watch out later on!)
- Variables can have the same name as a package, type, method, field, or statement label

```
class a {
  a a(a a) {
    a:
    for(;;) {
      if (a.a(a) == a)
        break a;
    }
    return a;
  }
}
```

Please do not do this either!

Java Language Specification, 2nd edition, p. 113