

---

# **Variables and assignments**

## **COMP 102**

**Victoria University of Wellington**

# Variables

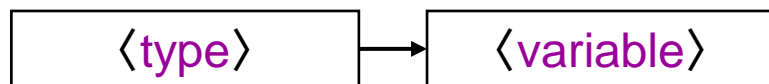
```

/** Convert from fahrenheit to Celsius */
public void doFahrenheitToCelsius(){
    double fahrenheit = UI.askDouble("Fahrenheit:");
    double celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    UI.println(fahrenheit + " F is " + celsius + " C");
}

```

Use a variable whenever you need the computer to remember something temporarily.

- A variable is a place in memory that can hold a value.
  - Must specify the **type** of value that can be put in the variable  
⇒ “Declare” the variable first time it is mentioned.



- Must put a value into a variable before you can use it  
⇒ “assign” to the variable
- Can *use* the value by specifying the variable’s name
- Can change the value in a variable (unlike mathematical variable)

Asking for a place

# Assignment Statements

```

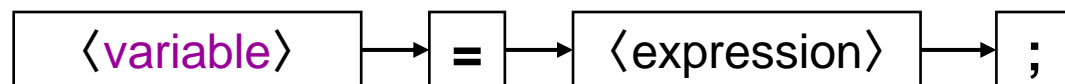
/** Convert from fahrenheit to Celsius */
public void doFahrenheitToCelsius(){
    double fahrenheit = UI.askDouble("Fahrenheit:");
    double celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    UI.println(fahrenheit + "F is " + celsius + " C");
}

```

Putting a value into a variable

- Assignment Statement:

*where*                      =                      *what* ;



*name-of-place*                      =                      *specification-of-value* ;

*formula*                      =                      " Celsius = (Fahrenheit - 32) \*5/9"

Meaning:    Compute the value and put it in the place

# Expressions

```

/** Convert from fahrenheit to Celsius */
public void doFahrenheitToCelsius(){
    double fahrenheit = UI.askDouble("Fahrenheit:");
    double celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    UI.println(fahrenheit + "F is " + celsius + " C");
}

```

//This line is combining these two lines into one

double fahrenheit;

fahrenheit = UI.askDouble("Fahrenheit:");

- Expressions describe how to compute a value.
- Expressions are constructed from
  - values
  - variables
  - operators (+, -, \*, /, etc)
  - method calls that return a value
  - sub-expressions, using (...)
  - ...

+ for Strings: "concatenates" the Strings

# Values / Data

---

There are lots of different kinds ("Types") of values:

- Numbers
  - Integers      **int** (or **long**)      42      -194573203
  - real numbers    **double** (or **float**)    42.0    16.43    6.626e-34
  - ...
- Characters      **char**      'X'    '4'
- Text            **String**      " F -> "
- Colours        **Color**      Color.red    Color.green
- Methods                           this::doFahrenheitToCelsius
- Other Objects
- ...

# Method Calls and variables: a metaphor

Method Definition: Like a pad of worksheets

```
public void doFahrenheitToCelsius(){
    double fahrenheit = UI.askDouble("Fahrenheit:");
    double celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
    UI.println(fahrenheit + " is " + celsius + " C");
}
```

Fahrenheit: 86
86.0F is 30.0C

86 0

Calling a Method:

30 0

```
tempCalc1.fahrenheitToCelsius();
```

- ⇒ get a “copy” of the method worksheet
- ⇒ perform each action in the body.
- ⇒ throw the worksheet away (losing all the information on it)

# Summary of Java program structure

---

- A Class specifies a type of object
  - TemperatureCalculator.class describes TemperatureCalculator objects
- A Class contains a collection of methods
  - Each method is an action the objects can perform.
  - TemperatureCalculator objects can do celsiusToFahrenheit, fahrenheitToCelsius, printFormula
  - If you have an object, you can call its methods on it.
- A method definition contains statements
  - Each statement specifies one step of performing the action
  - Method call statements
  - Declaration and Assignment statements