

Data Types and Operators

Variables, Data Types & Arithmetic Operators

Produced
by:

Department of Computing and Mathematics



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Topics list

- Variables.
- Java's Primitive Data Types.
- Arithmetic operators and Order of Evaluation.

Variables

In Programming, variables:

- are created (defined) in your programs.
- are used to store data (whose value can change over time).
- have a data type.
- have a name.
- are a VERY important programming concept.

Variable names...

- Are case-sensitive.
- Can be any length you choose.
- Must not be a **keyword or reserved word** e.g. int, while, etc.
- Cannot contain white spaces (i.e. space bar value).
- Should use full words instead of abbreviations e.g. **ratio** and **gear** is better than **r** and **g**.
- If the name consists of:
 - only one word, spell that word in all lowercase letters e.g. **gear**.
 - more than one word, capitalise the first letter of each subsequent word e.g. **engineSize** and **currentGear**.

Assignment Statement

- Values are stored in variables via assignment statements:

Syntax	<code>variable = expression;</code>
Example	<code>currentGear = 4;</code>

- A variable stores a single value, so any previous value is lost.
- Assignment statements work by taking the value of what appears on the right-hand side of the operator and copying that value into a variable on the left-hand side.

Topics list

- Variables.
- Java's Primitive Data Types.
- Arithmetic operators and Order of Evaluation.

Data Types

- In Java, when we define a variable, we **have** to give it a data type.
- The data type defines the **kinds of values** (data) that can be stored in the variable e.g.
 - - 456
 - 2
 - 45.7897
 - I Love Programming
 - S
 - true
- The data type also determines the operations that may be performed on it.

Java's Primitive Data Types

- Java has eight primitive data types i.e. predefined types in the language:
 - Four whole number data types: **byte**, **short**, **int**, **long**
 - Two decimal number data types: **float** and **double**
 - A single character data type: **char**
 - A true/false data type: **boolean**

Java's Primitive Data Types

- We will cover three of these:
 - int, float, boolean.
- A primitive type is highlighted red when it is typed into the PDE e.g.

```
int a;
```

```
float number;
```

```
boolean flag;
```

Java's Primitive Data Types (whole numbers)

Type	Minimum value (inclusive)	Maximum value (inclusive)	Default value
int	-2,147,483,648	2,147,483,647	0
float	<i>Beyond the scope of this course.</i>		0.0f
boolean	Holds either true or false and is typically used as a flag.		false

Declaring variables of an **int** type

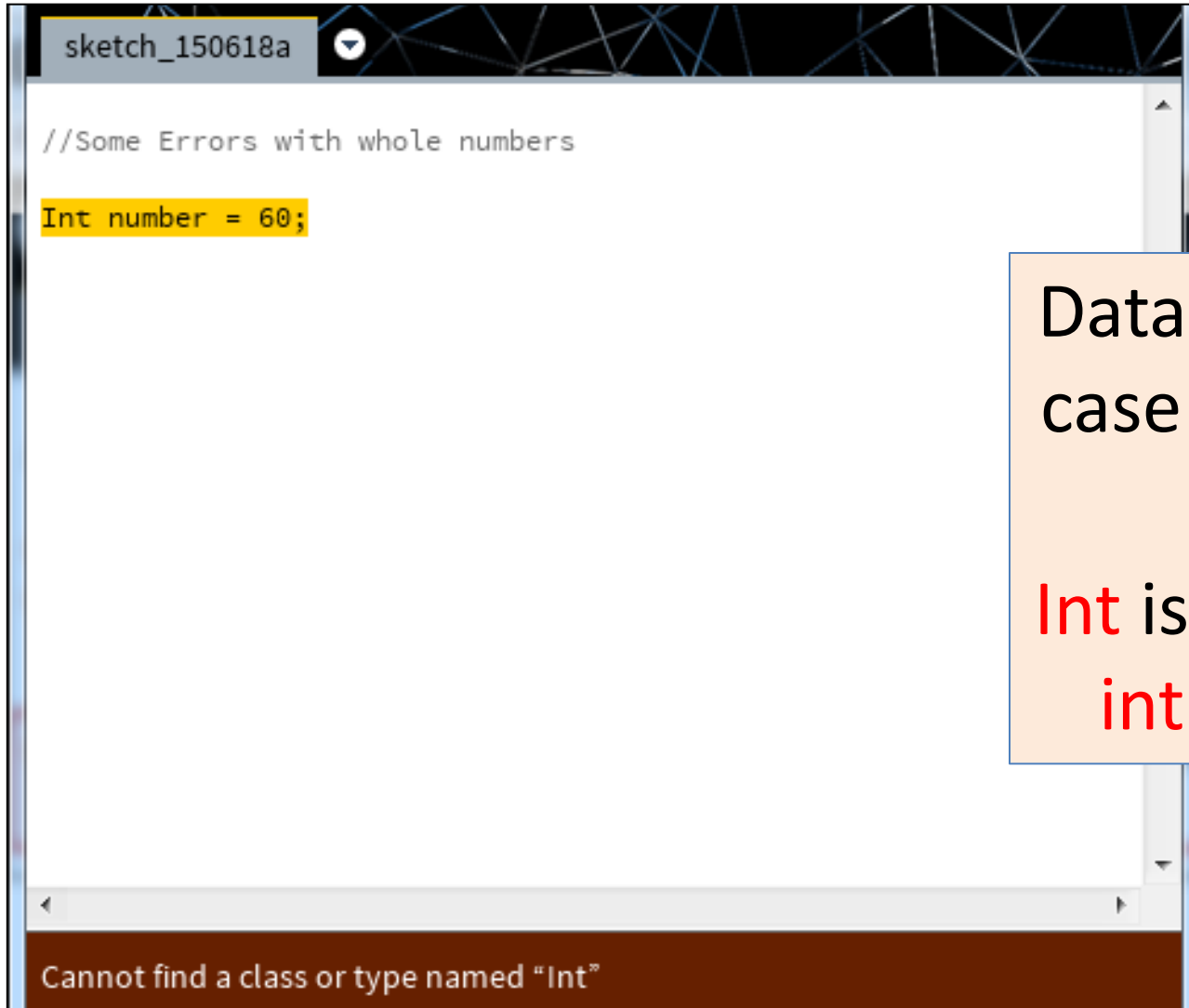
```
int secondNumber;           //declares a variable secondNumber of type int

firstNumber = 40;           //assign a value of 40 to firstNumber
secondNumber = 70;          //assign a value of 70 to secondNumber

int thirdNumber = 80;       //you can declare a variable and assign a value
                             //on one line.

int x, y, z;                //multiple variables of the same type can be defined
                             //on one line.
```

Declaring variables of an **int** type – some errors



The screenshot shows an IDE window titled "sketch_150618a". The code editor contains the following text:

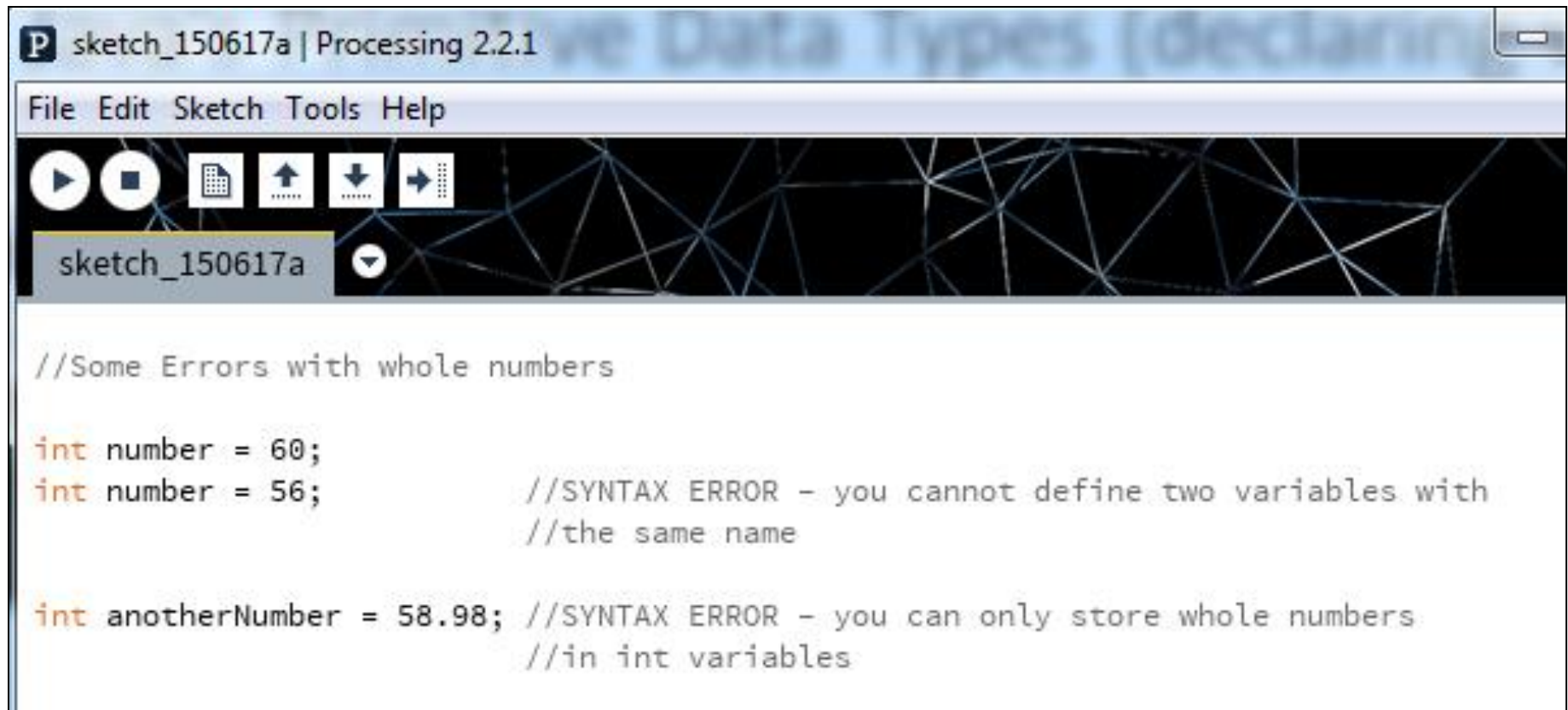
```
//Some Errors with whole numbers  
Int number = 60;
```

The line `Int number = 60;` is highlighted in yellow. At the bottom of the IDE, a brown error message bar displays the text: "Cannot find a class or type named 'Int'".

Data types are
case sensitive.

Int is not valid.
int is valid.

Declaring variables of an **int** type – some errors



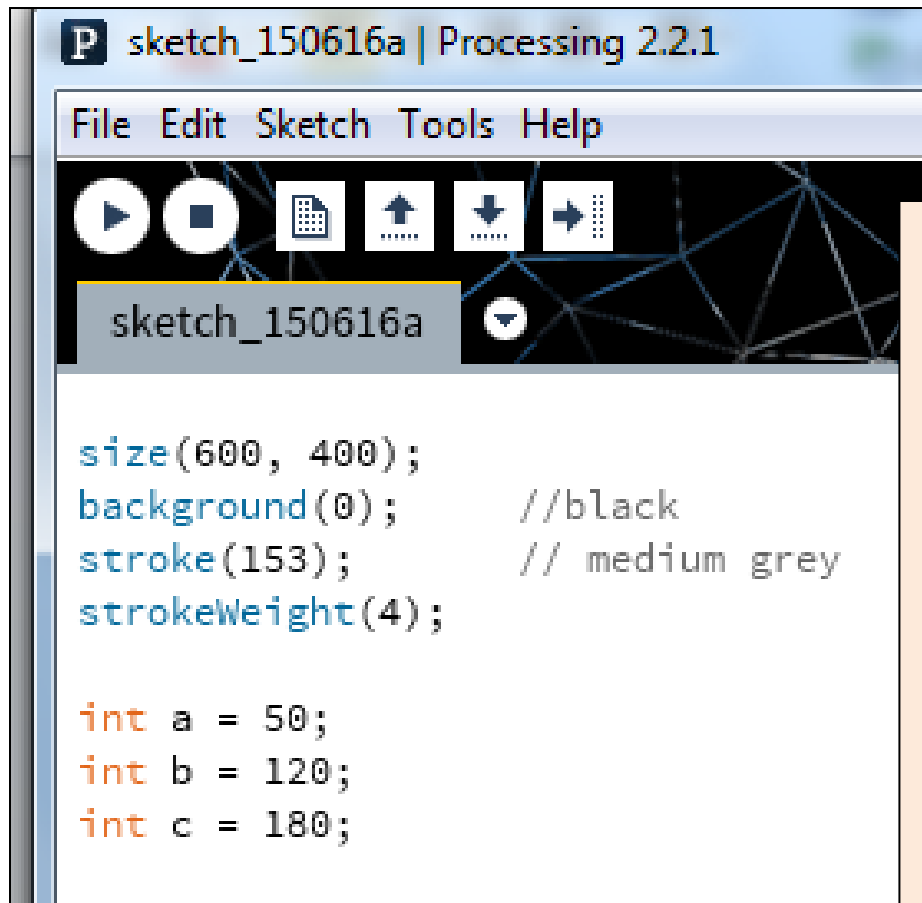
The screenshot shows the Processing IDE interface. The title bar reads "P sketch_150617a | Processing 2.2.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for running, stopping, opening, saving, and other functions. The sketch name "sketch_150617a" is displayed in a dropdown menu. The main code area contains the following text:

```
//Some Errors with whole numbers

int number = 60;
int number = 56;           //SYNTAX ERROR - you cannot define two variables with
                           //the same name

int anotherNumber = 58.98; //SYNTAX ERROR - you can only store whole numbers
                           //in int variables
```

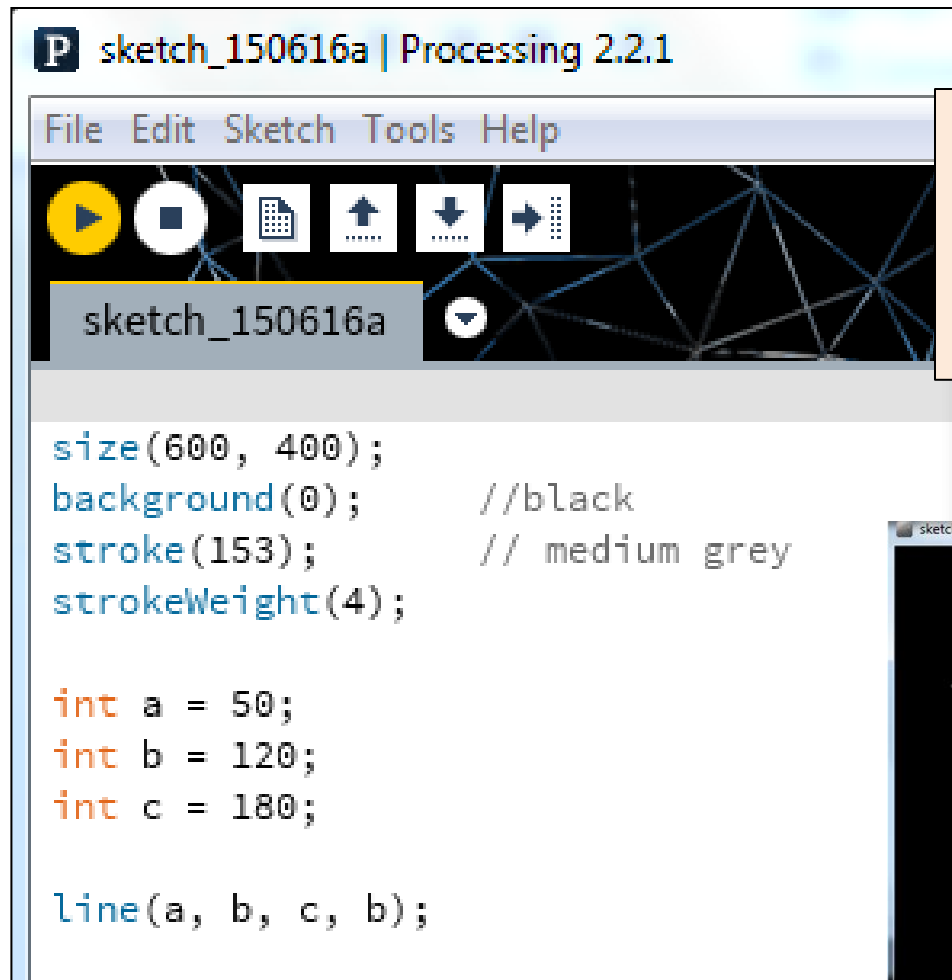
Java's Primitive Data Types: int Example 4.1



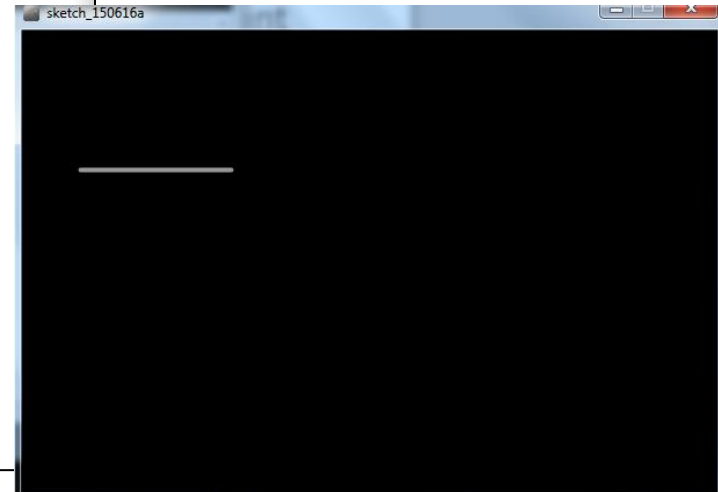
In this example, we have:

- defined three variables (a, b and c)
- that can hold whole numbers (int).
- and are set with a starting value.

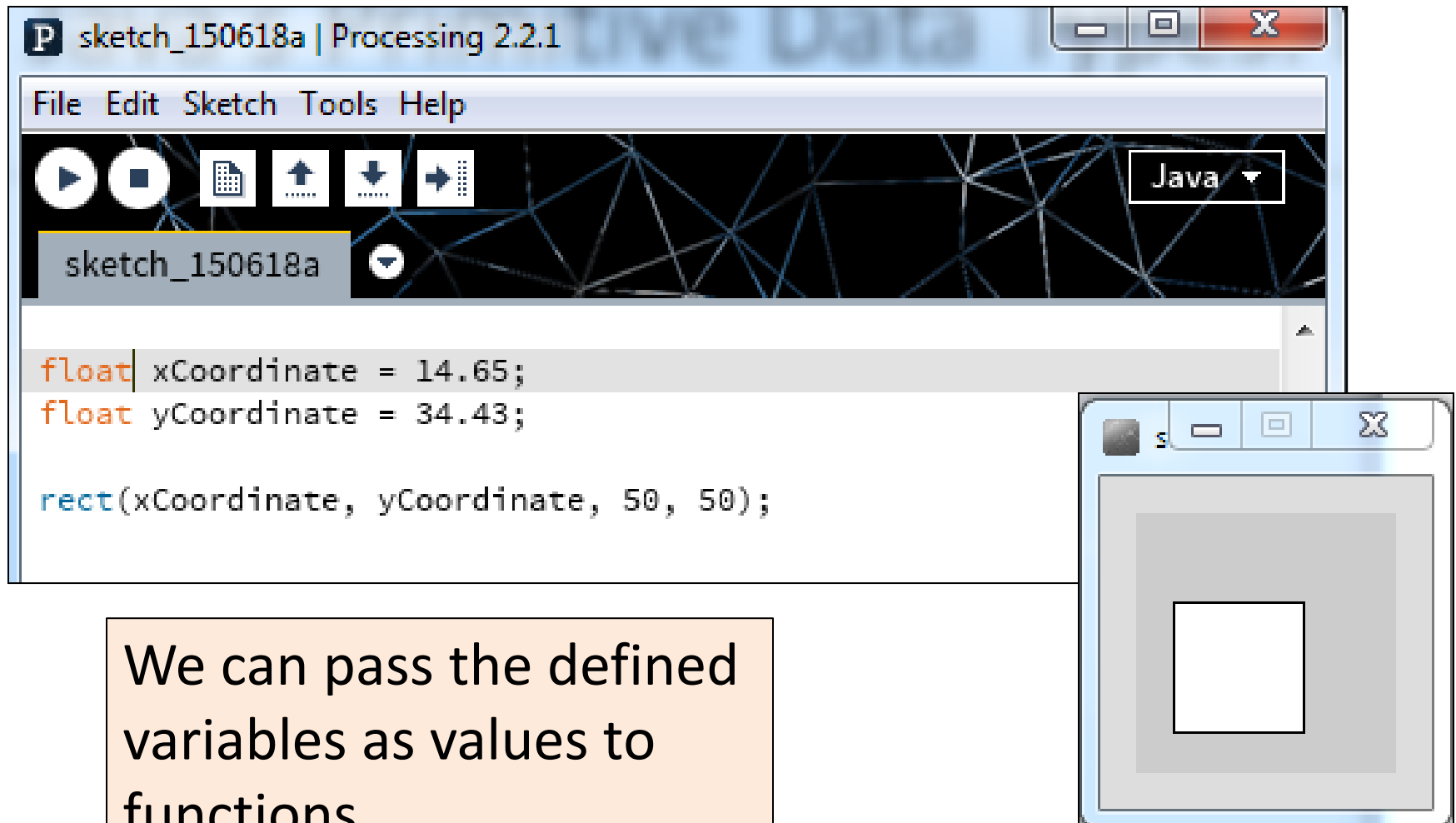
Java's Primitive Data Types: int Example 4.2



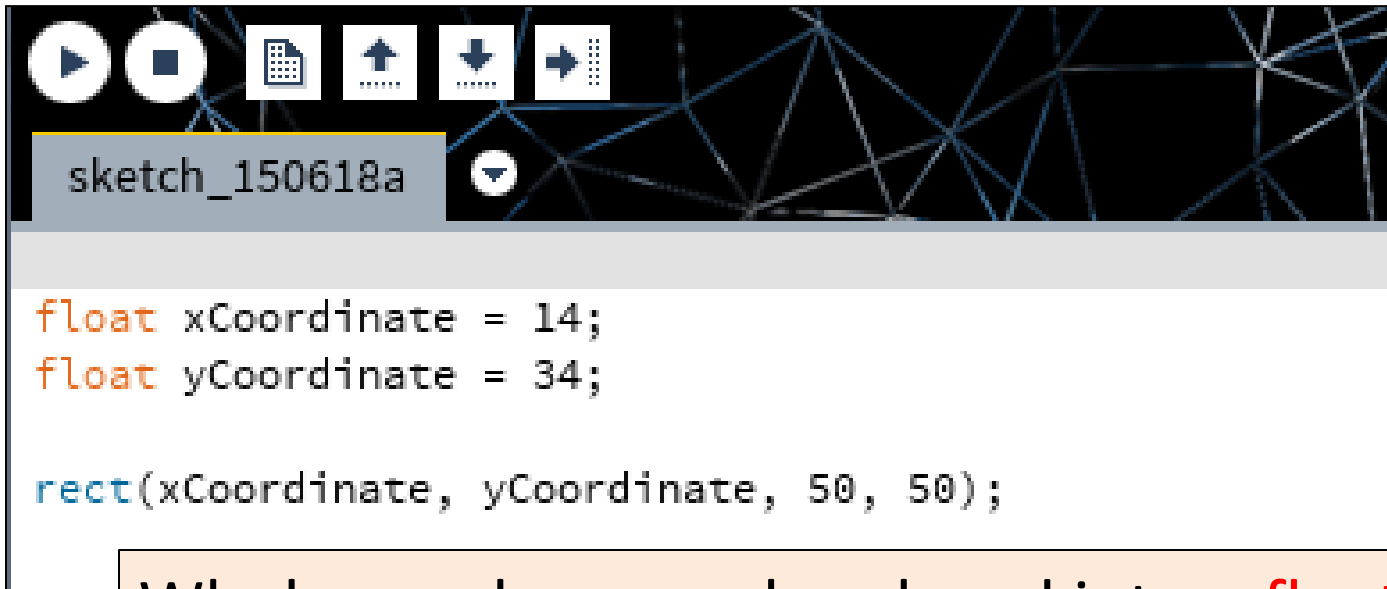
We can pass the defined variables as values to functions.



Java's Primitive Data Types: float Example 4.3



Java's Primitive Data Types: float Example 4.4



```
float xCoordinate = 14;
float yCoordinate = 34;

rect(xCoordinate, yCoordinate, 50, 50);
```

Whole numbers can be placed into a **float** variable.

Q: Why?

A: There is no loss of precision. We are not losing any data.

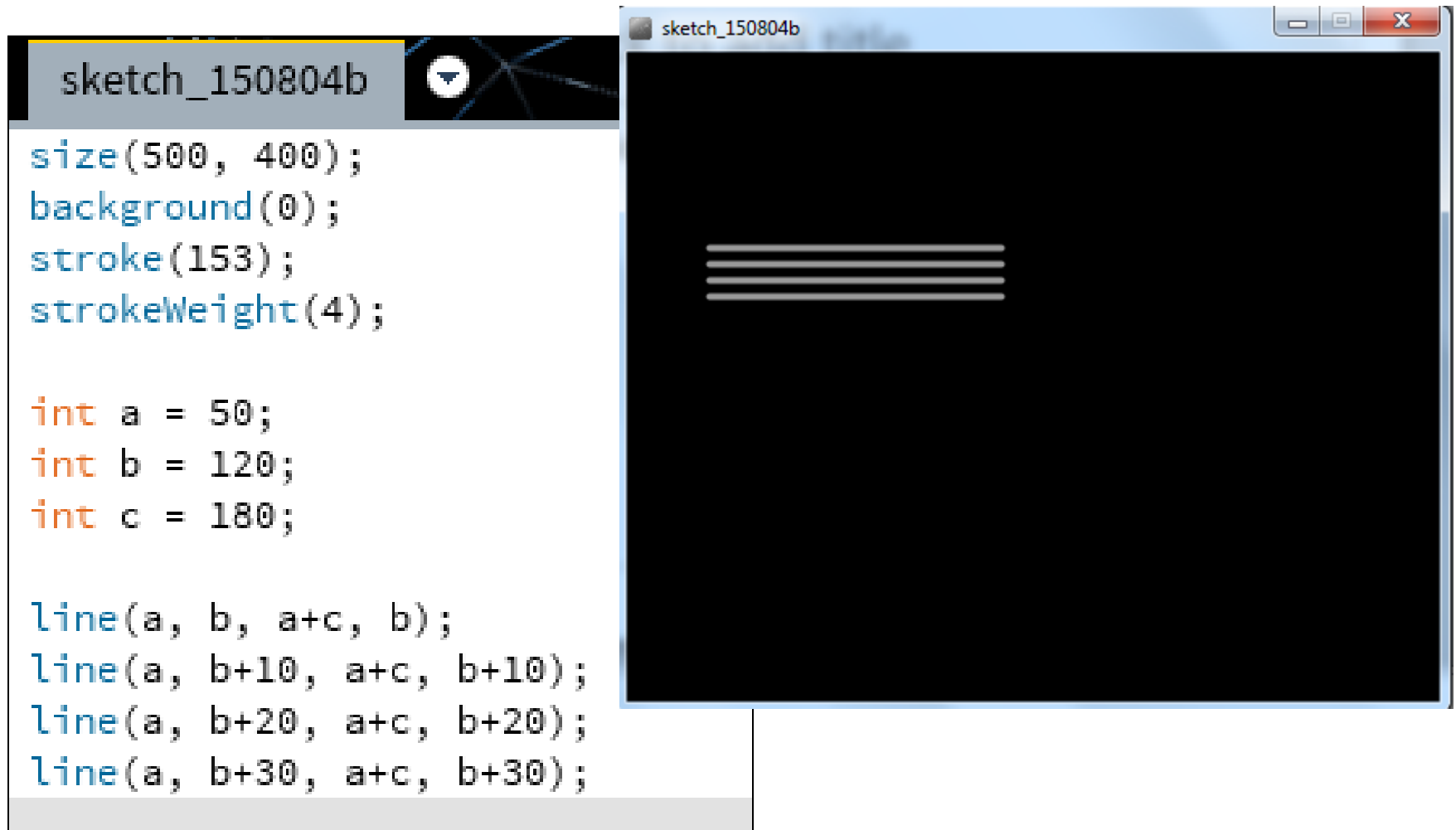
Topics list

- Variables.
- Java's Primitive Data Types.
- Arithmetic operators and Order of Evaluation.

Arithmetic Operators

Arithmetic Operator	Explanation	Example(s)
+	Addition	$6 + 2$ amountOwed + 10
-	Subtraction	$6 - 2$ amountOwed - 10
*	Multiplication	$6 * 2$ amountOwed * 10
/	Division	$6 / 2$ amountOwed / 10

Arithmetic operators: Example 4.5



Arithmetic operators: Example 4.6

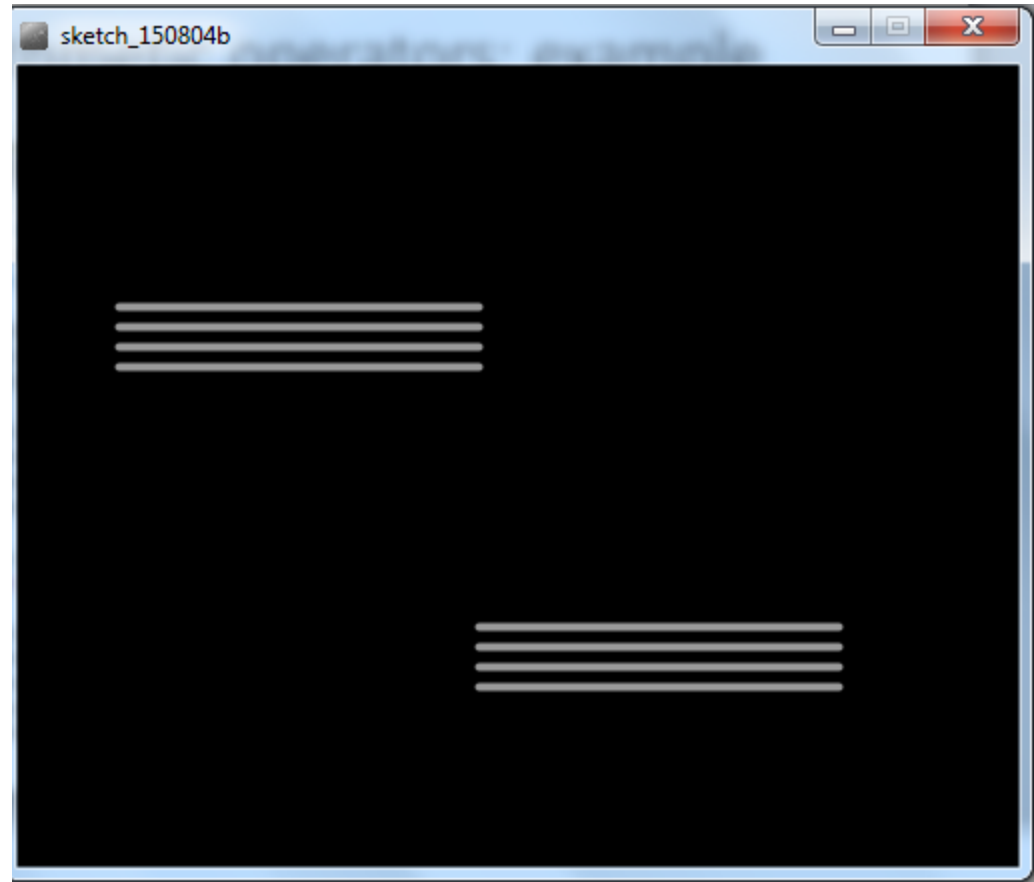
```
sketch_150804b
size(500, 400);
background(0);
stroke(153);
strokeWeight(4);

int a = 50;
int b = 120;
int c = 180;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);
line(a, b+20, a+c, b+20);
line(a, b+30, a+c, b+30);

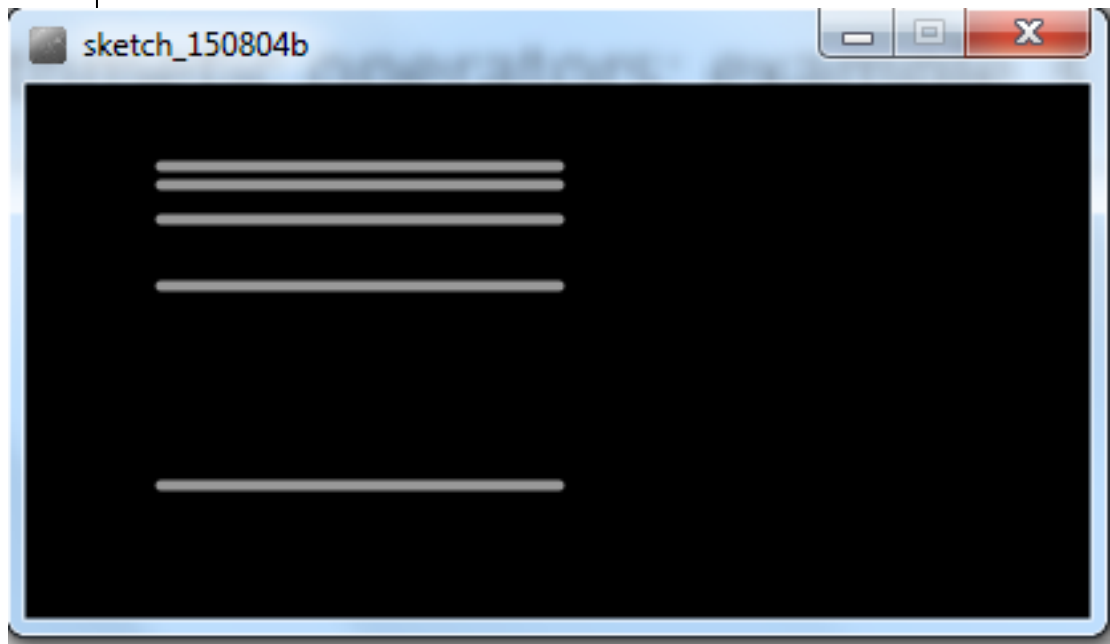
a = a + c;
b = height-b;

line(a, b, a+c, b);
line(a, b+10, a+c, b+10);
line(a, b+20, a+c, b+20);
line(a, b+30, a+c, b+30);
```



Arithmetic operators: Example 4.7

```
sketch_150804b  
  
size(400, 200);  
background(0);  
stroke(153);  
strokeWeight(4);  
  
int a = 50;  
int b = 1500;  
int c = 4;  
  
line(a, b/10, a*c, b/10);  
line(a, b/20, a*c, b/20);  
line(a, b/30, a*c, b/30);  
line(a, b/40, a*c, b/40);  
line(a, b/50, a*c, b/50);
```



Arithmetic Operators

- If you want to keep track of how many times something happens, you are keeping a **running total** e.g.
 - The number of times you drew a line on the computer screen.
 - As each line is drawn, you add one to your counter variable.

Arithmetic Operators

```
int counter = 0;
```

```
void draw()
```

```
{
```

```
  line (mouseX, mouseY, 50,50);
```

```
  counter = counter + 1;
```

```
  println (counter);
```

```
}
```


Arithmetic Operators

- These examples are straightforward uses of the arithmetic operators.
- However, we typically want to do more complex calculations involving many arithmetic operators.
- To do this, we need to understand the **Order of Evaluation**.

Order of Evaluation

- Brackets ()
- Multiplication (*)
- Division (/)
- Addition (+)
- Subtraction (-)

BoMDAS

Beware My Dear Aunt Sally

Order of Evaluation - Quiz

What are the results of these calculations?

Q1: $3+6*5-2$

Q2: $3+6*(5-2)$

Q3: $(3+6)*5-2$

Questions?





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>