

Variables, Data Types & Arithmetic Operators



- Produced Ms. Mairead Meagher,
 - by: Ms. Siobhán Roche.

Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

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.
- Begin with either:
 - a **letter (preferable)**,
 - the dollar sign "\$", or
 - the underscore character "_".
- Can contain letters, digits, dollar signs, or underscore characters.
- Can be any length you choose.
- Must not be a **keyword or reserved word** e.g. int, while, etc.
- Cannot contain white spaces.

Variable names should be carefully chosen

- Use full words instead of cryptic abbreviations e.g.
 - variables named **speed** and **gear** are much more intuitive than abbreviated versions, such as **s** and **g**.
- If the name consists of:
 - only one word,
 - spell that word in all lowercase letters e.g. **ratio**.
 - more than one word,
 - capitalise the first letter of each subsequent word e.g. **gearRatio** and **currentGear**.
 - This is called **camelCase**

Kinds of variables so far

- Fields are one sort of variable.
 - They store values through the life of an object.
 - They are accessible throughout the class.
- Parameters are another sort of variable:
 - They receive values from outside the method.
 - They help a method complete its task.
 - Each call to the method receives a fresh set of values.
 - Parameter values are short lived.

Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

Java Keywords

- Words with a special meaning in the language, e.g:
 - `public`
 - `class`
 - `private`
 - `int`
- Also known as *reserved words*.
- Special words that the Java language keeps for itself.

Keywords – list

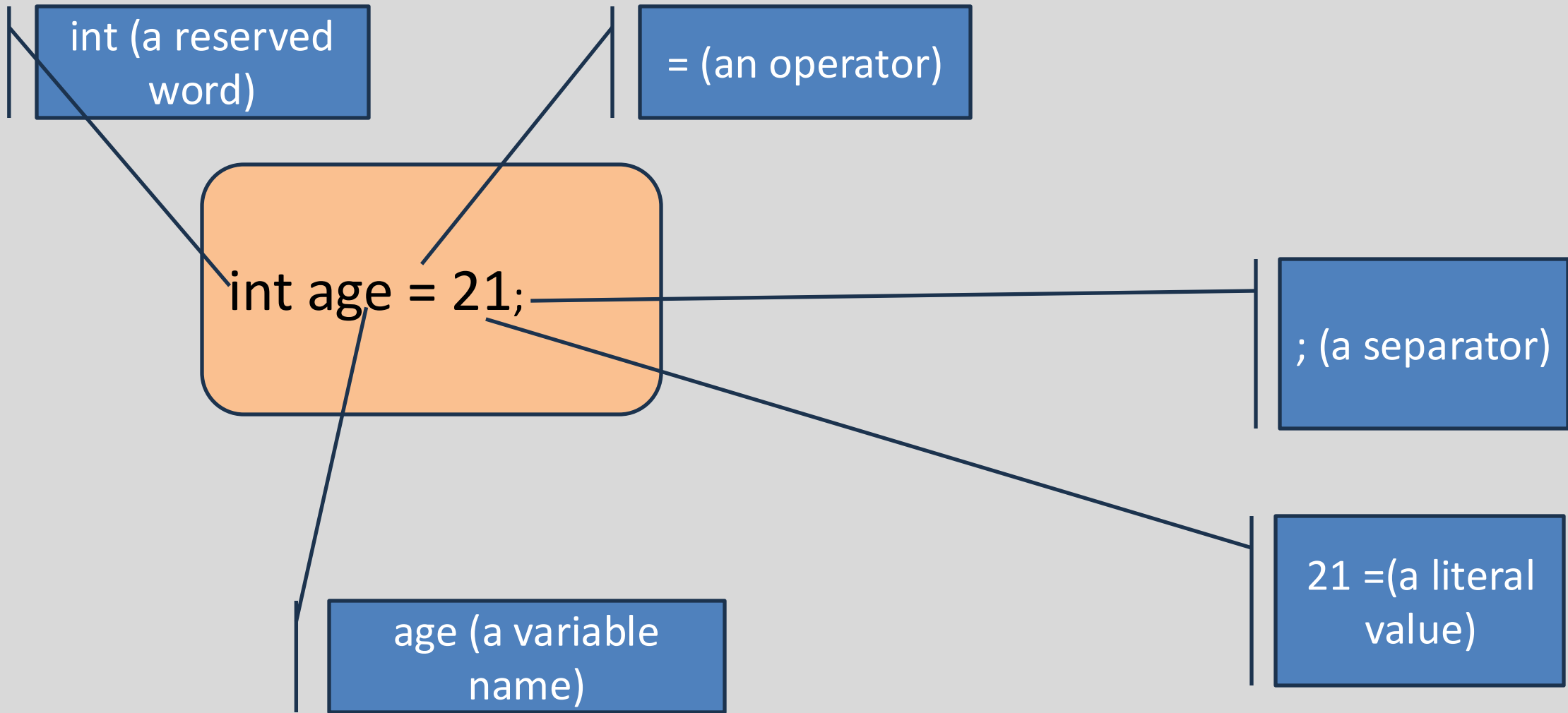
Java Keywords - for the moment, only the blues are important							
_	catch	double	float	int	private	super	TRUE
abstract	char	else	for	interface	protected	switch	try
assert	class	enum	goto	long	public	synchronized	void
boolean	const	extends	if	native	return	this	volatile
break	continue	FALSE	implements	new	short	throw	while
byte	default	final	import	null	static	throws	
case	do	finally	instanceof	package	strictfp	transient	

Keywords (technical description)

A token is small
meaningful piece
of code

- A **reserved word** (or keyword) in Java is a **token** that has a predefined meaning in the language's grammar and cannot be redefined or used as an identifier.
- They are part of the Java specification.
- They are case-sensitive (e.g., Class is not the same as class).
- Using a reserved word as an identifier results in a compile-time error.

Example of tokens – let's **tokenise** a stmt



Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

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.

Data Types

- Java uses two kinds of data types:
 - **Primitive** types
 - **Object** types
- We are only looking at **Primitive** types now; we will cover Object types later in the module.

Java's Primitive Data Types

- Java programming language supports eight primitive data types.
- A primitive type is predefined by the language and is named by a reserved keyword.
- A primitive type is highlighted red when it is typed into the BlueJ editor e.g.

int numberOfItems;

boolean isAlive

float sizeOfSquare;

Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

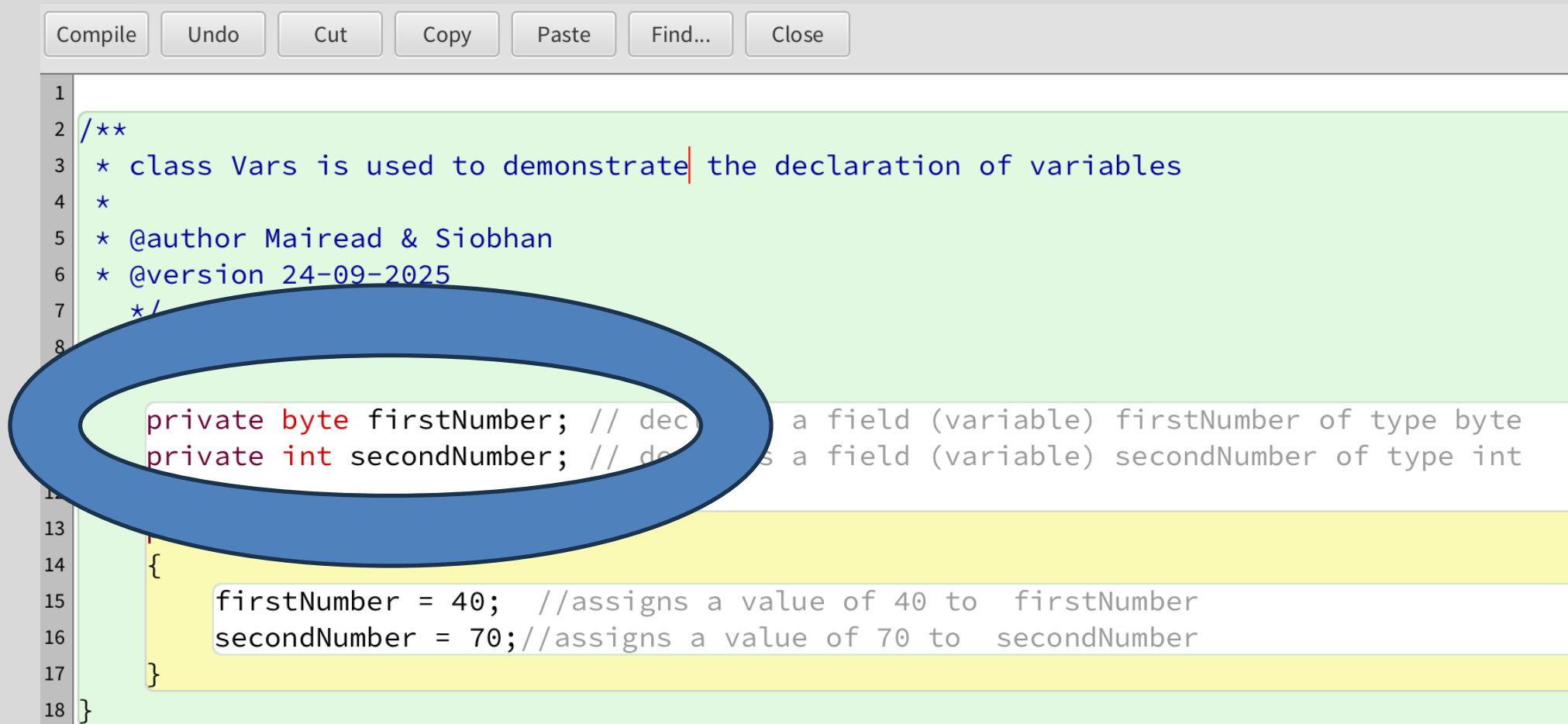


Java's Primitive Data Types (whole numbers)

Type	Byte-size	Minimum value (inclusive)	Maximum value (inclusive)	Typical Use
byte	8-bit	-128	127	Useful in applications where memory savings apply.
short	16-bit	-32,768	32,767	
int	32-bit	-2,147,483,648	2,147,483,647	Default choice.
long	64-bit	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	Used when you need a data type with a range of values larger than that provided by int.

We will mostly use int for whole numbers

Declaring variables of a specific type



```
1  
2 /**  
3  * class Vars is used to demonstrate the declaration of variables  
4  *  
5  * @author Mairead & Siobhan  
6  * @version 24-09-2025  
7  */  
8  
9 private byte firstNumber; // declares a field (variable) firstNumber of type byte  
10 private int secondNumber; // declares a field (variable) secondNumber of type int  
11  
12 {  
13  
14     firstNumber = 40; //assigns a value of 40 to firstNumber  
15     secondNumber = 70; //assigns a value of 70 to secondNumber  
16  
17 }  
18 }
```

Declaring variables of a specific type

Compile Undo Cut Copy Paste Find... Close

```
1
2 /**
3  * class Vars2 is used to demonstrate the declaration of variables
4  *
5  * @author Mairead & Siobhan
6  * @version 24-09-2025
7  */
8 public class Vars2
9 {
10
11     private int thirdNumber = 80;    //you can declare a variable and assign
12                                     // a variable on the one line
13
14     private int x,y,z;                //multiple variables of the same type
15                                     // can be declared on one line
16
17     // code omitted
18 }
```

We can assign a value at the same time as declaration. This is called **defining the variable**

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

private int x,y,z; //multiple variables of the same type
// can be declared on one line

// code omitted

Declaring variables – some errors - 1

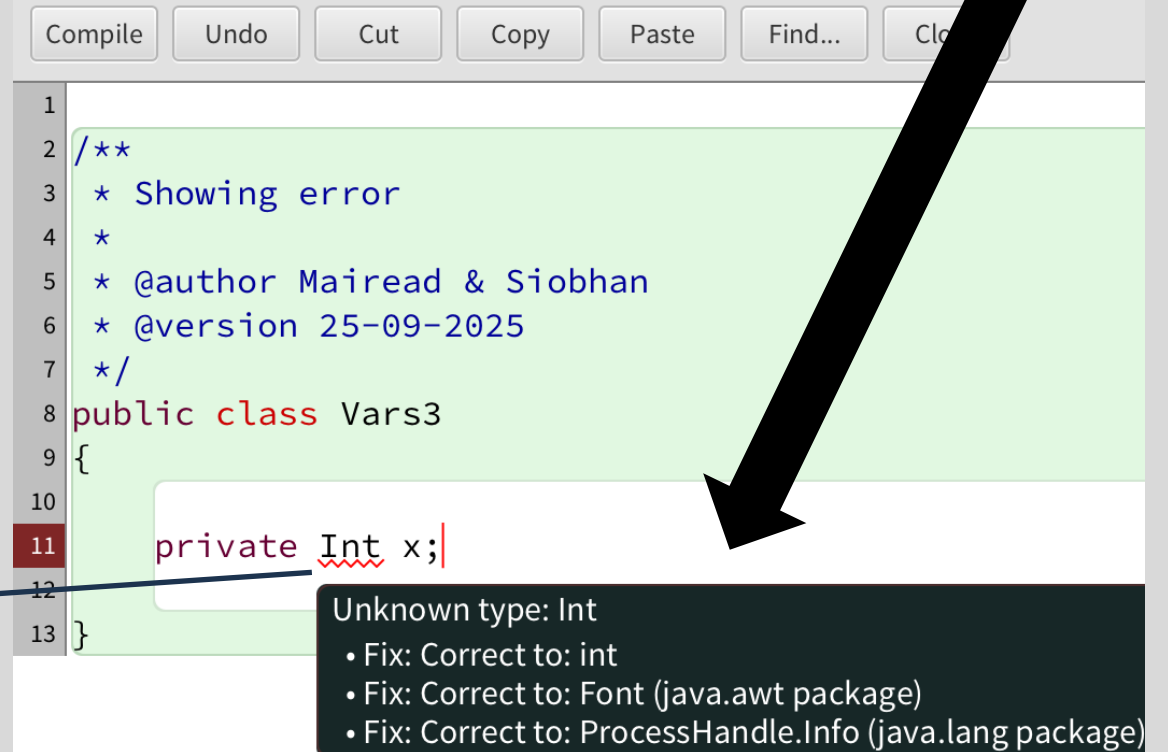
- Data types are case sensitive

– **Int** is not valid

– **int** is valid

Red underline indicates that there is a syntax error.

If you right-click on red underline, BlueJ will help you fix the error



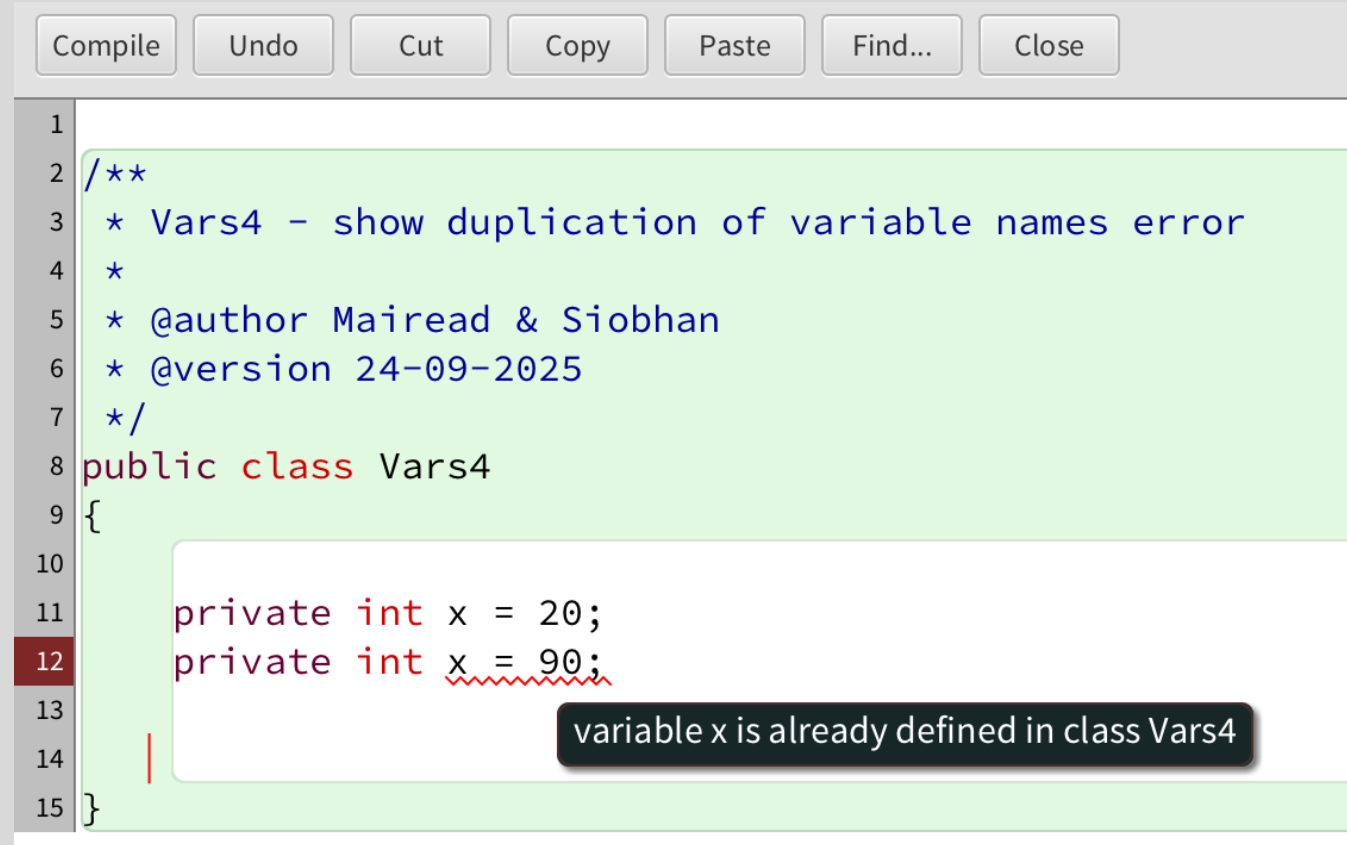
```
1
2 /**
3  * Showing error
4  *
5  * @author Mairead & Siobhan
6  * @version 25-09-2025
7  */
8 public class Vars3
9 {
10
11     private Int x;
12
13 }
```

Unknown type: Int

- Fix: Correct to: int
- Fix: Correct to: Font (java.awt package)
- Fix: Correct to: ProcessHandle.Info (java.lang package)

Declaring variables – some errors - 2

- You can only have one declaration of any variable in a context.



The screenshot shows an IDE window with a toolbar at the top containing buttons for Compile, Undo, Cut, Copy, Paste, Find..., and Close. The code editor displays a Java class named Vars4. The code is as follows:

```
1
2 /**
3  * Vars4 - show duplication of variable names error
4  *
5  * @author Mairead & Siobhan
6  * @version 24-09-2025
7  */
8 public class Vars4
9 {
10
11     private int x = 20;
12     private int x = 90;
13
14
15 }
```

Line 12 is highlighted in red, and the variable `x` in the second declaration is underlined with a red wavy line. A tooltip message is displayed next to the error, stating: "variable x is already defined in class Vars4".

Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

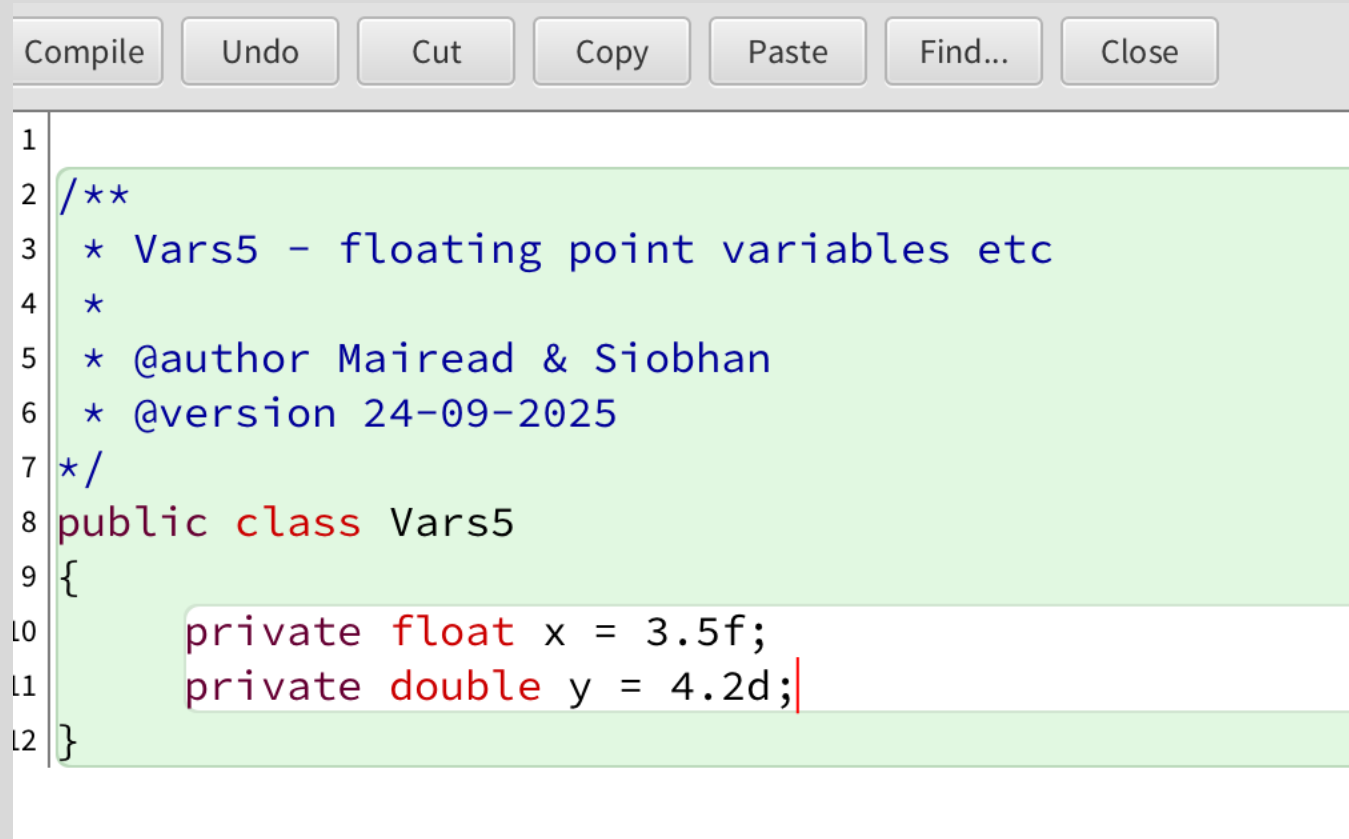


Java's Primitive Data Types (decimal numbers)

Type	Byte-size	Minimum value (inclusive)	Maximum value (inclusive)	Typical Use
float	32-bit	<i>Beyond the scope of this lecture .</i> <i>There is also a loss of precision in this data-type that we will cover in later lectures.</i>		Useful in applications where memory savings apply. Default choice when using Processing .
double	64-bit			Default choice when programming Java apps .

Declaring/defining a floating point field

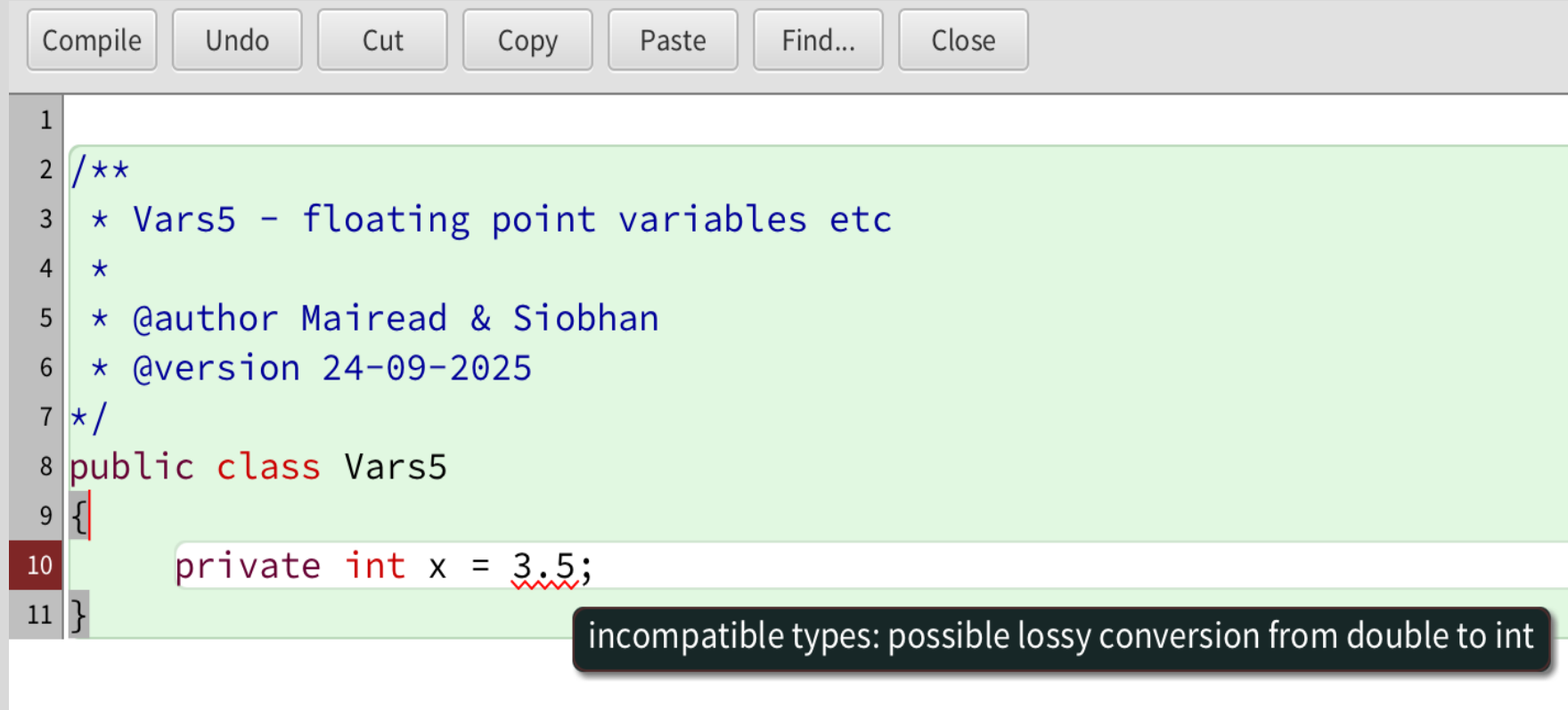
- Use float or double for non-integer numbers.
- When assigning numbers, use the f or d suffix with float or double.



```
1  
2 /**  
3  * Vars5 - floating point variables etc  
4  *  
5  * @author Mairead & Siobhan  
6  * @version 24-09-2025  
7  */  
8 public class Vars5  
9 {  
10     private float x = 3.5f;  
11     private double y = 4.2d;  
12 }
```


Declaring variables – some errors - 3

- Cannot assign a non-integer value to an integer variable.



The screenshot shows a code editor window with a toolbar at the top containing buttons for Compile, Undo, Cut, Copy, Paste, Find..., and Close. The code is as follows:

```
1
2 /**
3  * Vars5 - floating point variables etc
4  *
5  * @author Mairead & Siobhan
6  * @version 24-09-2025
7  */
8 public class Vars5
9 {
10     private int x = 3.5;
11 }
```

Line 10 is highlighted, and a red squiggly line is under the value 3.5. A tooltip at the bottom right of the editor displays the error message: "incompatible types: possible lossy conversion from double to int".

Topics list

1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.



Java's Primitive Data Types (others)

Type	Byte-size	Minimum value (inclusive)	Maximum value (inclusive)	Typical Use
<code>char</code>	16-bit	'\u0000' (or 0)	'\uffff' (or 65,535)	Represents a Unicode character.
<code>boolean</code>	1-bit	n/a		Holds either true or false and is typically used as a flag.

- We will go into more detail on these two data types in later lectures.

http://en.wikipedia.org/wiki/List_of_Unicode_characters

Topics list

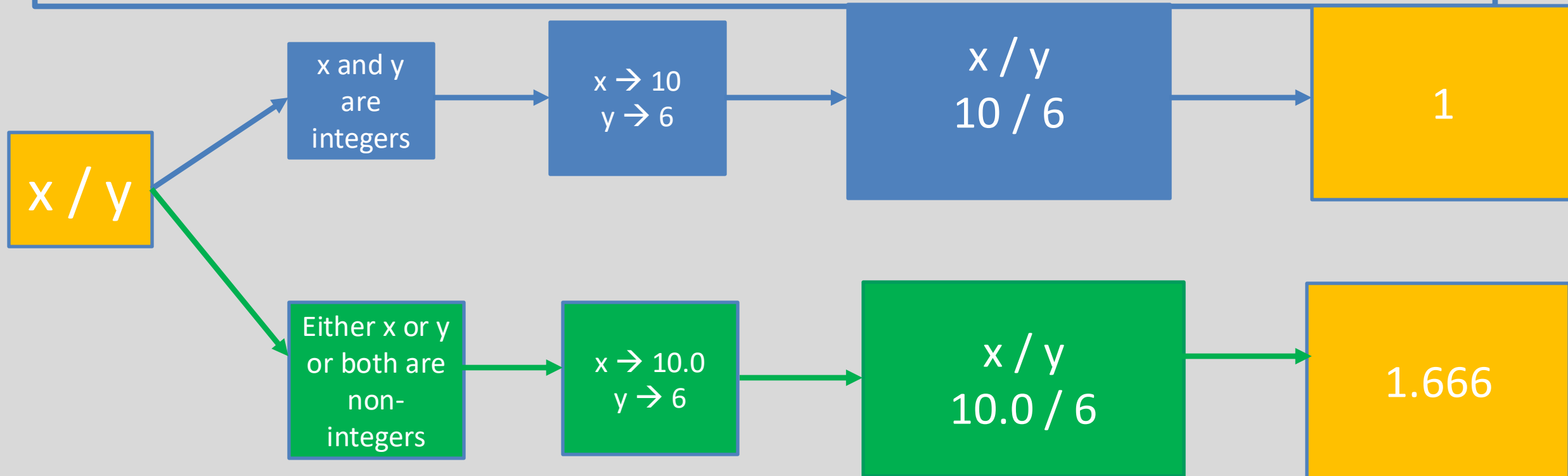
1. Variables.
2. Java Keywords, tokens
3. Data Types.
4. Java's Primitive Data Types
 1. Whole numbers.
 2. Decimal numbers.
 3. Others.
5. Arithmetic operators.

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
%	Remainder	16 % 5 → 1

Note on division '/' – integer or 'normal' division

When we use the '/' operator, how it behaves depends on the type of the two operands :



Questions?

