

# CS620c Structured Programming

## Lesson 5

Joe Duffin

Email: [Joseph.Duffin@nuim.ie](mailto:Joseph.Duffin@nuim.ie)

---

# Be mindful of the following issues....

- Common mistakes....
    - Semi-colons – *at the end of every statement*
    - Brackets – *close all open brackets*
    - Capital letters – *String, System*
  - JRE- How to run a program using this.
  - The special meaning of \ in Java
-

# Recap

- So far on the module we've covered:
  - Writing algorithms
  - Creating programs with Notepad++ using the SDK
  - Writing a program to print to the screen
  - Last lecture we moved on to variables...
- Today, you'll learn how to declare and initialise variables, and how to solve simple mathematical expressions.

# Data Types (Revisited)

- Java has eight primitive data types:

Type	Contains	Size (in bits)
byte	whole number	8
short	whole number	16
<b>int</b>	whole number	32
long	whole number	64
<b>float</b>	decimal point number	32
<b>double</b>	decimal point number	64
<b>char</b>	a unicode character	16
<b>boolean</b>	true or false	1

# Variables – declaration and instantiation

## ■ Example approaches:

```
int num1; //declare a variable called num1
```

```
num1 = 5; //initialise to store value of 5
```

```
int num1 = 5; //declare and initialise together
```

```
int num1 = 5, num2 = 6; //more than 1 variable at a time
```

```
boolean isFalse = false;
```

```
float num3 = 8.9f;
```

```
double num4 = 5.6;
```

```
char letter = 'z';
```

# Let's do an example together...

- Write a Java program which:
  - ❑ declares 4 variables, `num1`, `num2`, `result1` and `result2` of type `int`.
  - ❑ initialises `num1` to 5 and `num2` to 7, `result1` and `result2` to default values of 0.
  - ❑ adds `num1` to `num2` and stores the result in `result1`.
  - ❑ multiplies `num2` by 5 and stores the result in `result2`.
  - ❑ prints out the value stored in `result1` and `result2`
- **All your programs from now on should start with an algorithm**

---

## Points to note...

- How else could we declare and initialise our variables?
  - How about if I wanted to print the values of the numbers I added and the result to the screen on the same line?
  - How about the question on the next slide.. Its very similar but we overwrite the previous value of our variables.... See Variables.java
-

# Overwriting variable values...

- Write a Java program called Variables.java which
  - ❑ declares 3 variables, `num1`, `num2`, `num3` of type `int`.
  - ❑ initialises `num1` to 5 and `num2` to 7.
  - ❑ adds `num1` to `num2` and stores the result in `num1`.
  - ❑ multiplies `num2` by 5 and stores the result in `num3`.
  - ❑ prints out `num1`, `num2` and `num3`



# Some points on identifiers

## ■ Identifiers (names)

- ❑ *Identifiers* are the names of variables, methods and classes
- ❑ can contain numbers, letters, the underscore symbol and the dollar symbol
- ❑ must begin with a letter, or with \_ or \$
- ❑ can contain uppercase and lowercase letters
- ❑ case sensitive
- ❑ must not be a **keyword**
- ❑ must be unique within its scope
- ❑ cannot contain whitespace
- ❑ if the identifier is composed of two or more words capitalise the second and subsequent words or separate words using an underscore, e.g. **(this is a coding convention not enforced by the compiler)**

```
int totalSales = 0;  
int total_sales=0;
```

# Keywords (The full set this time)

abstract	boolean	break	byte	case
catch	char	class	const	continue
default	do	double	else	extends
final	finally	float	for	goto
if	implements	import	instanceof	int
interface	long	native	new	package
private	protected	public	return	short
static	strictfp	super	switch	synchronized
this	throw	throws	transient	try
void	volatile	while		

# Valid variable names

- The following are **legal** variable names:
  - ❑ MyVariable
  - ❑ myVariable
  - ❑ MYVARIABLE
  - ❑ myvariable
  - ❑ x
  - ❑ i
  - ❑ \_myvariable
  - ❑ \$myvariable
  - ❑ This\_is\_an\_insanely\_long\_variable\_name\_that\_just\_keeps\_going\_and\_going\_and\_going\_and\_well\_you\_get\_the\_idea\_I\_cant\_imagine\_why\_you\_would\_need\_such\_a\_long\_variable\_name\_but\_if\_you\_do\_you\_can\_have\_it

# Invalid variable names

- The following are not legal variable names (why not?):
  - ❑ My Variable
  - ❑ 9pins
  - ❑ a+c
  - ❑ testing1-2-3
  - ❑ O'Reilly

# Try this one in pairs...

- Write a Java program called Variables.java which
  - ❑ declares 3 variables, `num1`, `num2`, `num3` of type float.
  - ❑ initialises `num1` to 5.7 and `num2` to 14.3.
  - ❑ divides `num2` by `num1` and stores the result in `num1`.
  - ❑ multiplies `num2` by 4 and stores the result in `num3`.
  - ❑ prints out `num1`, `num2`, and `num3`

---

# Summary

- This Lesson, you learnt:
    - How to declare and initialise variables
    - How to perform simple mathematical calculations
    - Valid identifiers in Java
    - Keywords in Java
-