

# Programming Fundamentals

An Introduction to the module

---

Produced by: Mr. Colm Dunphy  
Dr. Siobhán Drohan



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

# Agenda

---

- *Lecturers / Tutors*
- *Module Structure & Delivery*
- *Technologies*
- *Module Assessment (Assignments)*
- *Troubleshooting Labs*
- *Ethos*

# Introducing your lecturers / tutors

---

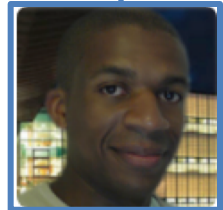
## Colm Dunphy

- Profile: [https://www.wit.ie/about\\_wit/contact\\_us/staff\\_directory/colm\\_dunphy](https://www.wit.ie/about_wit/contact_us/staff_directory/colm_dunphy)
- Email: [cdunphy@wit.ie](mailto:cdunphy@wit.ie)



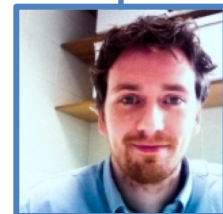
## Patrick Felicia

- Profile: [https://www.wit.ie/about\\_wit/contact\\_us/staff\\_directory/patrick\\_felicia](https://www.wit.ie/about_wit/contact_us/staff_directory/patrick_felicia)
- Email: [pfelicia@wit.ie](mailto:pfelicia@wit.ie)



## Jonathan Brazil

- Profile: [https://www.wit.ie/about\\_wit/contact\\_us/staff\\_directory/colm\\_dunphy](https://www.wit.ie/about_wit/contact_us/staff_directory/colm_dunphy)
- Email: [jbrazil@wit.ie](mailto:jbrazil@wit.ie)



# Module Structure

---

12 weeks of delivery

2 \* Lectures / Talks  
(webinar/video)

Mon  
12:15 –  
2:00

Wed  
12:15 –  
2:00

2 \* Labs

Mon  
Tues

Wed  
Thurs

Lab Support  
on Slack

within 24 hours M-F

\* Double Module (10 credits)



# Learning Technologies

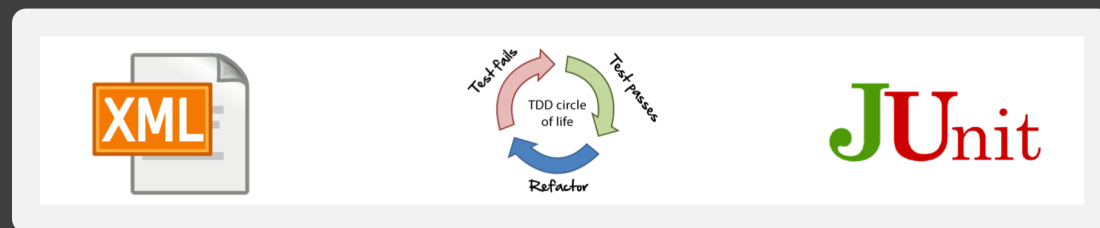




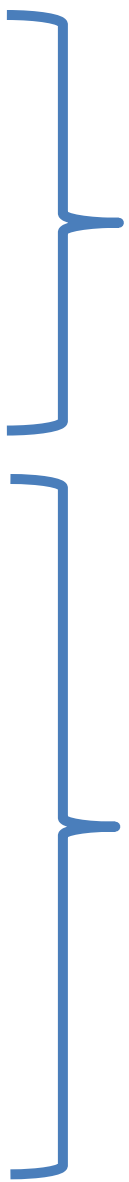
# Programming Technologies



# Programming Technologies



Semester 1		S	M	T	W
January	Week	6	7	8	9
	1	13	14	15	16
	2	20	21	22	23
	3	27	28	29	30
February	4	3	4	5	6
	5	10	11	12	13
	<i>reading-week</i>	17	18	19	20
	6	24	25	26	27
March	7	3	4	5	6
	8	10	11	12	13
	<i>reading-week</i>	17	18	19	20
	9	24	25	26	27
April	10	31	1	2	3
	11	7	8	9	10
	<i>Easter-break</i>	14	15	16	17
	<i>Easter-break</i>	21	22	23	24
	12	28	29	30	1
May	<i>reading-week</i>	5	6	7	8
	<i>reading-week</i>	12	13	14	15




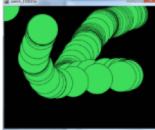



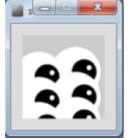


Week	Lecture
1	Intro to Processing Animate your Drawings
2	If statement and Events Iteration (for and while)
3	Methods More on Methods
4	Strings and Intro to Classes Classes and Encapsulation
5	Swing (JOptionPane) and Arrays Arrays and Classes
6	Pong Intro Pong Solutions




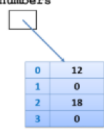
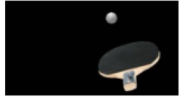


Week	Lecture
7	IntelliJ, JVM and I/O
8	Grouping Objects (ArrayLists)
9	Menu and CRUD
10	Persistence (XML) and Exceptions
11	Inheritance and Polymorphism
12	Collections (Map and Set)





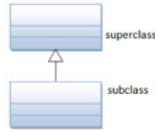

## Assignment 1 (P1)

<p>1a: Introduction to Processing</p>  <p>processing · java · drawing shapes · colour · grayscale · RGB · syntax errors · commenting code</p>	<p>1b: Animating your Drawings</p>  <p>animating simple drawings · variables · system variables · primitive data types · operators</p>	<p>2a: Selection and Events</p>  <p>animated drawings · conditional statements · relational operators · logical operators · variables · mouse events · key events</p>	<p>2b: Iteration (Loops)</p>  <p>iteration · for loops · while loops · variable scope · compound assignment statements · print · println</p>	<p>3a: Methods (Part 1)</p>  <p>method signature · return types · parameters · mouse event methods · bespoke methods</p>	<p>3b: Methods (Part 2)</p>  <p>more sophisticated methods · return types · parameters · recursion · Strings · API · String methods · JOptionPane</p>
--	--	--	---	---	--

## Assignment 2 (P2)

<p>4a: Classes (Part 1)</p>  <p>String methods · Objects · Classes · behaviour · attributes · Spot class</p>	<p>4b: Classes (Part 2)</p>  <p>classes · methods · behaviour · overloading · validation · this · encapsulation · access modifiers · accessors · mutators</p>	<p>5a: Swing and Arrays</p>  <p>GUI · Swing · JOptionPane · dialog boxes · primitive arrays · array syntax</p>	<p>5b: Arrays and Classes</p>  <p>primitive arrays · array syntax · arrays and loops · length · classes</p>	<p>6: Game of Pong</p>  <p>Game of Pong · Ball class · Paddle class · Player class · Tournaments · Statistics · Collision detection</p>
---	--	---	--	--

## Assignment 3 (P3)

<p>7: IntelliJ and Basic I/O</p>  <p>IntelliJ · Java Virtual Machine (JVM) · main method · Scanner · OO recap · Array recap</p>	<p>8: Grouping Objects</p>  <p>primitive arrays · classes · algorithms for collections · ArrayLists</p>	<p>9: Menu Driven Apps and Persistence</p>  <p>Switch · Loops · Menus · persistence · CRUD · debugging</p>	<p>10: Persistence (XML &amp; Exceptions)</p>  <p>Streaming · XML · Exception Handling · Validate User Input · Static · JavaDoc · ShopV5.0 · DVD3.0</p>	<p>11: Inheritance and Polymorphism</p>  <p>Inheritance · <i>is-a</i> relationship · Polymorphism (many shapes!) · Overriding</p>	<p>12: Collections (Map and Set)</p>  <p>Collections · Map · Set · Tech Support App</p>
--	--	---	--	--	--

# ASSESSMENT

Semester 1	
January	Week
	1
	2
	3
February	4
	5
	<i>reading-week</i>
	6
March	7
	8
	<i>reading-week</i>
	9
April	10
	11
	<i>Easter-break</i>
	<i>Easter-break</i>
May	12
	<i>reading-week</i>
	<i>reading-week</i>

P1 – 15%

P2 - 30%

P3 – 55%

% reflects difficulty and time required

# Assignments

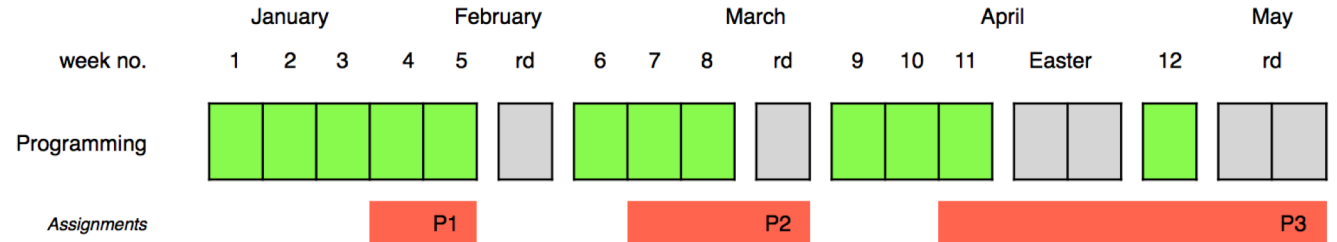
## Programming



 Assignment 1

 Assignment 2

 Assignment 3



Online  
Interviews / Demos





# Assignments

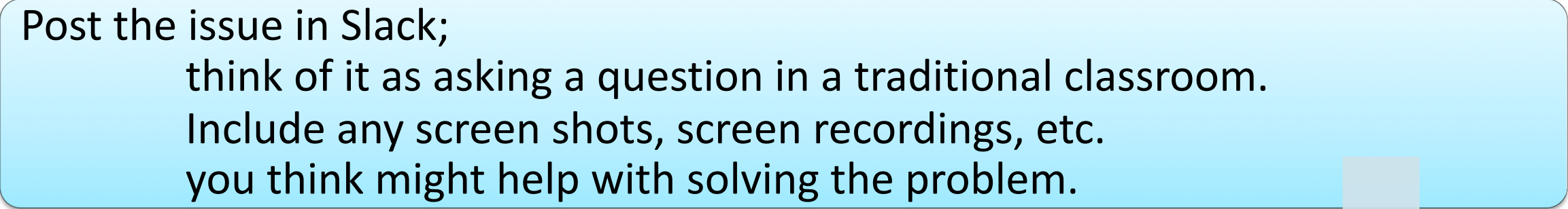
---

- 100% Continuous Assessment (CA).
- All Individual assignments
  - (no team-based ones).
- Submit via Moodle assignment dropboxes.
- Hard deadlines;  
extensions only permitted if [mitigating circumstances](#) apply.
- Interviews


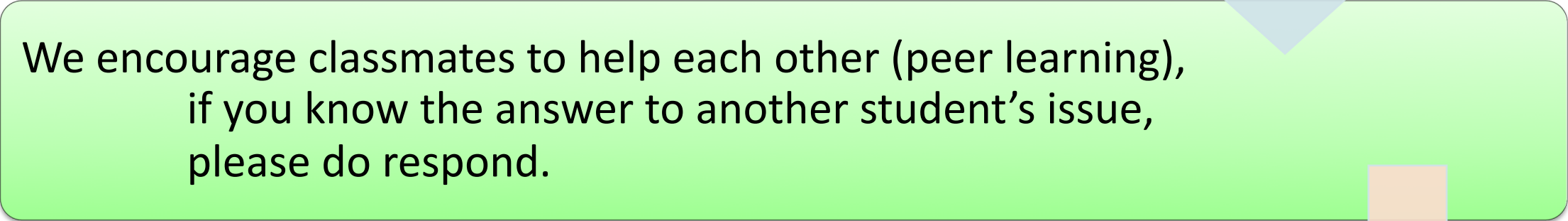
# *Troubleshooting labs*

## *...during the lab sessions*


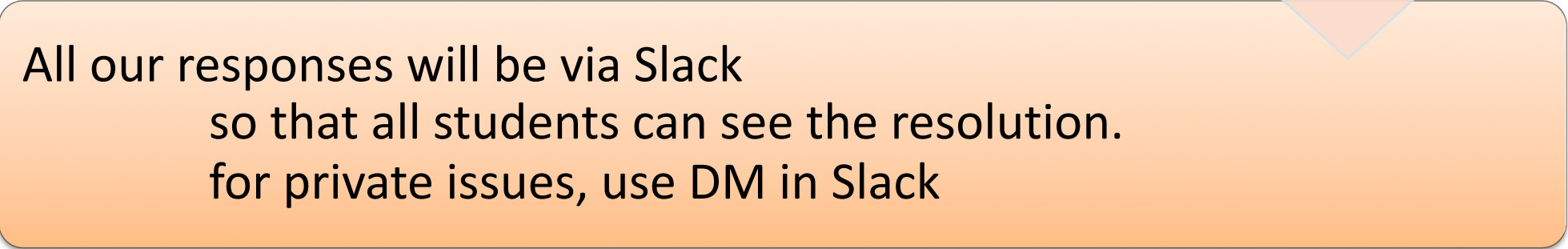
Post the issue in Slack;  
think of it as asking a question in a traditional classroom.  
Include any screen shots, screen recordings, etc.  
you think might help with solving the problem.



We encourage classmates to help each other (peer learning),  
if you know the answer to another student's issue,  
please do respond.




All our responses will be via Slack  
so that all students can see the resolution.  
for private issues, use DM in Slack

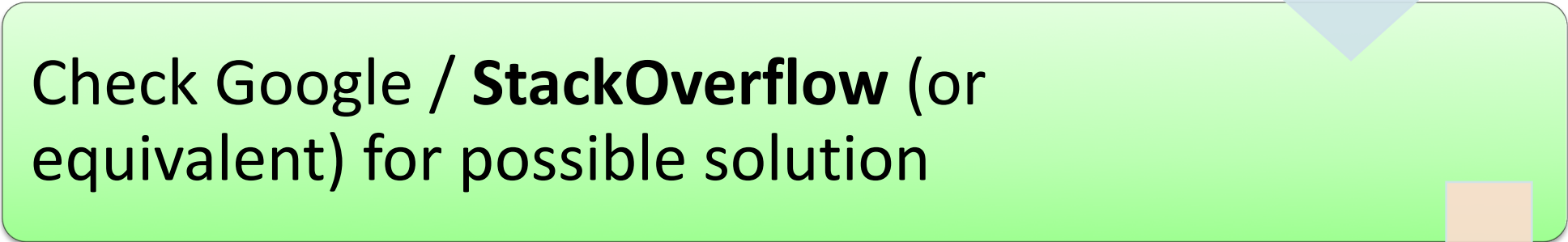


*Troubleshooting labs*  
*...**outside** of the lab sessions*

Search **#Slack** Chatroom



Check Google / **StackOverflow** (or equivalent) for possible solution



Post the issue on **#Slack** programming channel



# Ethos

---

- **Self-directed learning** outside of lectures / labs.
- Inquisitive and motivated.
- Help your **peers (use #Slack!)**.
- **Engagement** and staying current with the module.
- All work submitted must be **your own work**.
  - all code/approaches given in the module by us can be re-used / re-purposed in your assignments.

# Introduction to Processing



# What is Processing?

---



“Processing is a programming language,  
development environment,  
and online community.”

Source: <https://processing.org/>

Examples:

<http://www.openprocessing.org/browse/>

# Processing...

---



...can be used to develop static or interactive online material and data **visualisations**.

...is often used by visual **artists**.

...produces **visual** and **interactive** representations of programming code.

# What is Processing?

---



- Different programming languages can be used with Processing  
e.g. :

- Java: we will use this language.



Java

- JavaScript
- Python
- CoffeeScript
- Etc.



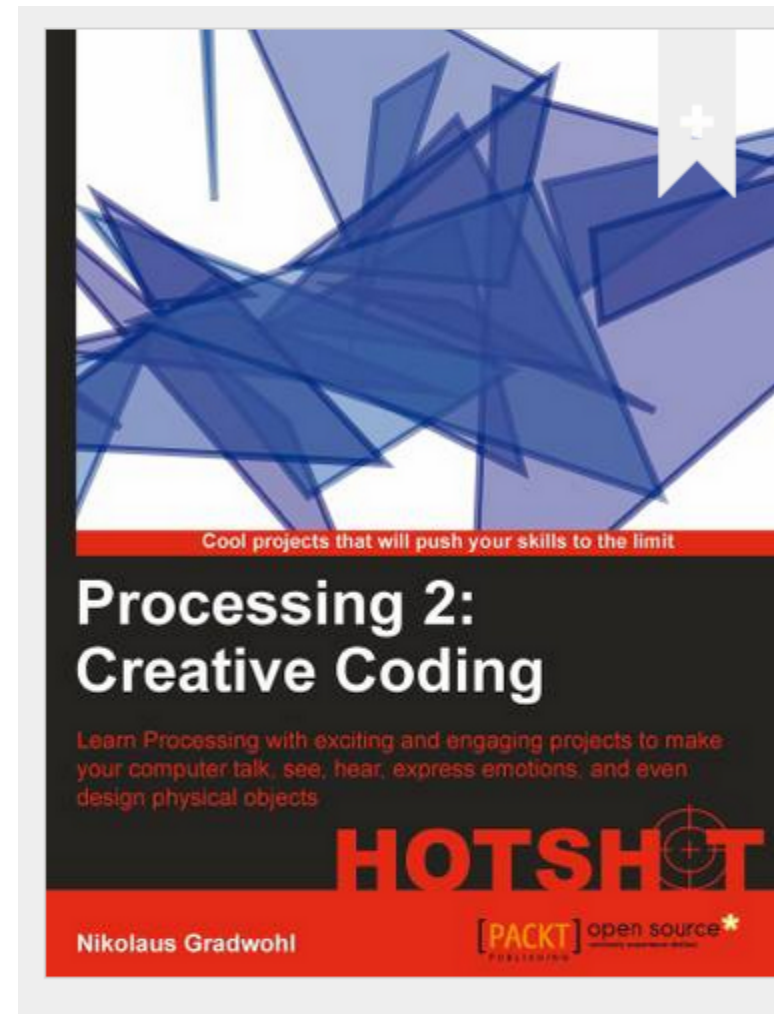
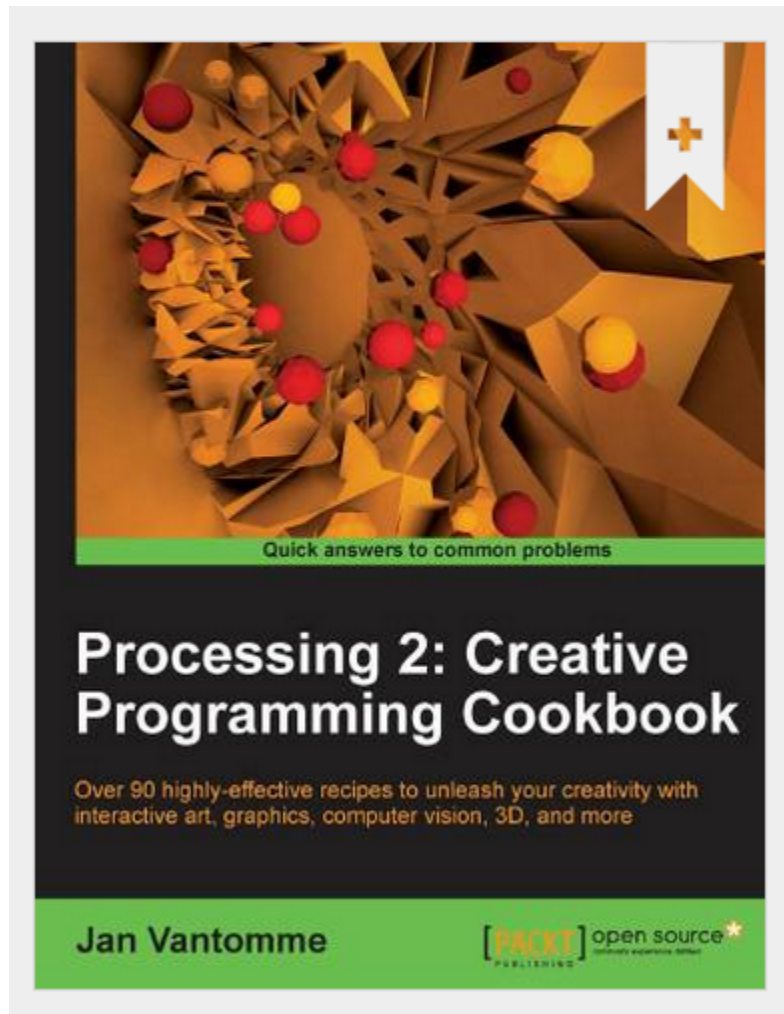
# Why are we using Processing?

---

*Processing is increasingly used  
to teach computer  
programming fundamentals  
(<https://processing.org/overview/>)*

# Some eBooks in WIT library

---



We will start coding in Processing  
in the afternoon session



# Questions?

---

