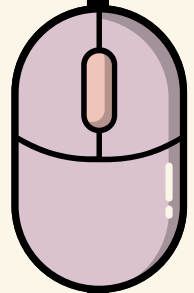
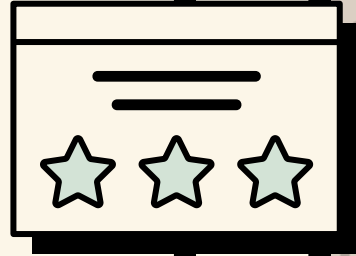
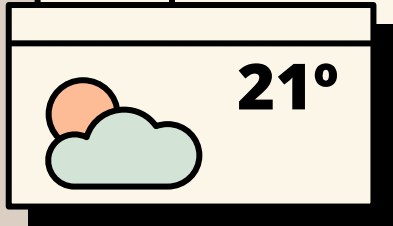
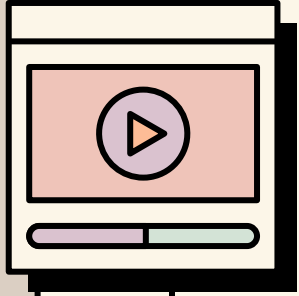


# Acad Day CS Sharing



Insights from current students on academics and student life

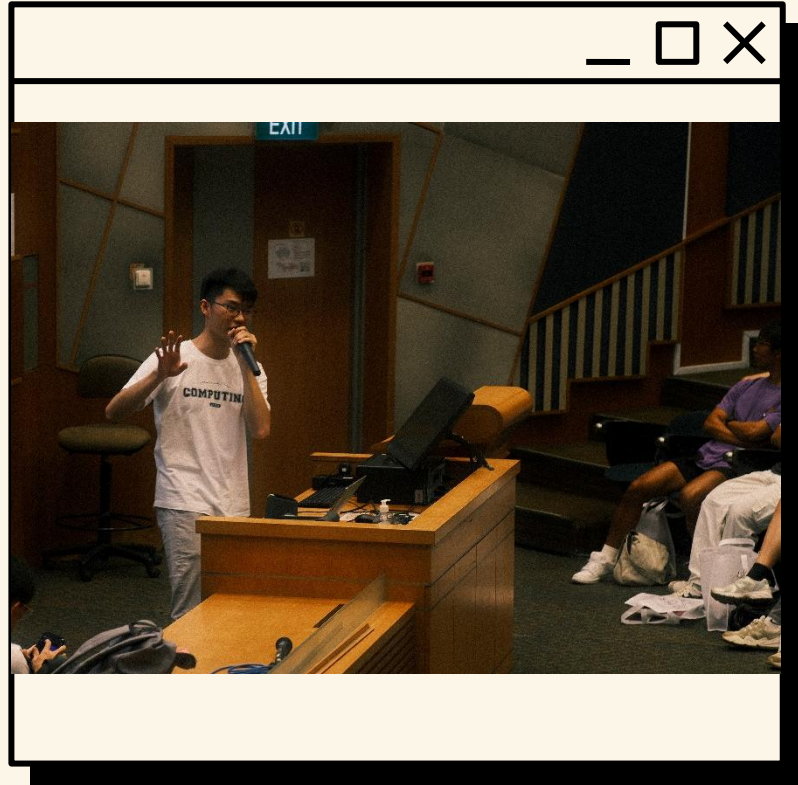
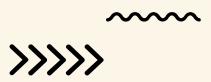




# Introduction

- Joshua
- Year 1 -> 2 Computer Science
- FOP Director of NUS Students' Computing Club
- Favorite food in Terrace is the mixed rice

.....





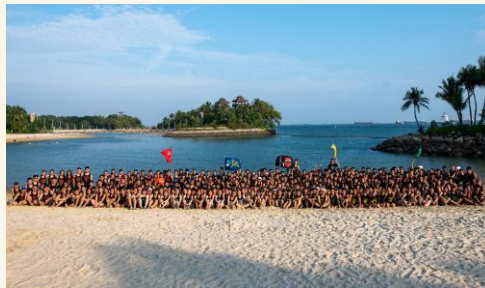
# 01 Before Uni Starts

# SOC Orientation Camps

FSC (Ended alr)

FOW (21<sup>st</sup> – 24<sup>th</sup>  
July)

FFC (4<sup>th</sup> – 6<sup>th</sup> Aug)





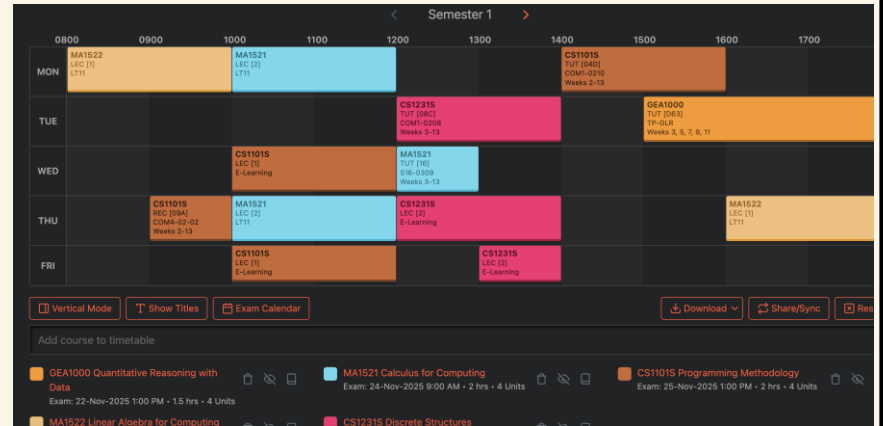
# 02 Academic Experience

# CourseReg & Course Bidding

Right before sem starts  
Consists of 2 parts, course bidding  
and tutorial bidding

Use NUSMods to plan your  
timetable

For sem 1 don't need to worry, most  
of your courses (up to 3 courses) will  
be preallocated





# Planning Courses

You don't have to go so overboard, but make sure to make a planner so that you can expect what courses to take

SOC curriculum is quite varied depending on what focus area you want to take, and you are free to choose all your courses

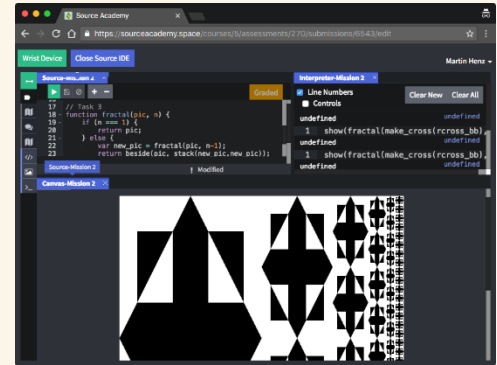
[Link to this sheet if u want](#)

Done	Mod	Link	Uni	CS	Math	Yes	Units
<input checked="" type="checkbox"/>	MA1521 Ca	<a href="#">Calculus for Computing</a>	4	Mathematics and Sciences	1000	Y1S1	46 160
<input checked="" type="checkbox"/>	CS1231S D	<a href="#">Discrete Structures</a>	4	Computer Science Foundation	1000	Y1S1	
<input checked="" type="checkbox"/>	CS1101S Pi	<a href="#">Programming Methodology</a>	4	Digital Literacy		Y1S1	Y1S1 22
<input checked="" type="checkbox"/>	MA1522 Lin	<a href="#">Linear Algebra for Computing</a>	4	Mathematics and Sciences	1000	Y1S1	Y1S2 28
<input checked="" type="checkbox"/>	GEA1000 Q	<a href="#">Quantitative Reasoning with Data</a>	4	Data Literacy		Y1S1	Y2S1 26
<input checked="" type="checkbox"/>	CFG1002 C	<a href="#">Career Catalyst</a>	2	Unrestricted Electives		Y1S1	Y2S2 26
<input checked="" type="checkbox"/>	CS2030S P	<a href="#">Programming Methodology II</a>	4	Computer Science Foundation		Y1S2	Y3S1 12
<input checked="" type="checkbox"/>	CS2040S D	<a href="#">Data Structures and Algorithms</a>	4	Computer Science Foundation		Y1S2	Y3S2 4
<input checked="" type="checkbox"/>	IS1108 Digi	<a href="#">Digital Ethics and Data Privacy</a>	4	Computer Ethics		Y1S2	Y4S1 10
<input checked="" type="checkbox"/>	EG1311 De	<a href="#">Design and Make</a>	4	Interdisciplinary Education		Y1S2	Y4S2 6
<input checked="" type="checkbox"/>	VHC1001 S	<a href="#">Sports for Life</a>	4	Unrestricted Electives		Y1S2	
<input checked="" type="checkbox"/>	MA2104 Mu	<a href="#">Multivariable Calculus</a>	4	Unrestricted Electives	2000	Y1S2	134
<input type="checkbox"/>	GEN2050X	<a href="#">Teach SG</a>	4	Communities and Engagement		Y2	
<input type="checkbox"/>	ES2660 Co	<a href="#">Communicating in the Information Age</a>	4	Critique and Expression		Y2S1	
<input type="checkbox"/>	CS2100 Co	<a href="#">Computer Organisation</a>	4	Computer Science Foundation		Y2S1	
<input type="checkbox"/>	MA2101 Lin	<a href="#">Linear Algebra II</a>	4	Unrestricted Electives	2000	Y2S1	
<input type="checkbox"/>	CDE2501 L	<a href="#">Liveable Cities</a>	4	Singapore Studies		Y2S1	
<input type="checkbox"/>		#VALUE!	4	Interdisciplinary Education		Y2S1	
<input type="checkbox"/>	CS2109S In	<a href="#">Introduction to AI and Machine Learning</a>	4	Computer Science Foundation		Y2S1	
<input type="checkbox"/>	MA2108S M	<a href="#">Mathematical Analysis I</a>	4	Unrestricted Electives	2000	Y2S2	
<input type="checkbox"/>	CS2101 Eff	<a href="#">Effective Communication for Computing Profess</a>	4	Computer Science Foundation		Y2S2	
<input type="checkbox"/>	CS2103T S	<a href="#">Software Engineering</a>	4	Computer Science Foundation		Y2S2	
<input type="checkbox"/>	ST2334 Pro	<a href="#">Probability and Statistics</a>	4	Mathematics and Sciences	2000	Y2S2	
<input type="checkbox"/>		#VALUE!	4	Cross-Disciplinary Education		Y2S2	
<input type="checkbox"/>		#VALUE!	4	Cultures and Connections		Y2S2	
<input type="checkbox"/>	CS2106 Intr	<a href="#">Introduction to Operating Systems</a>	4	Computer Science Foundation		Y3S1	
<input type="checkbox"/>	CS3230 De	<a href="#">Design and Analysis of Algorithms</a>	4	Computer Science Foundation		Y3S1	
<input type="checkbox"/>	CP4101 B.C	<a href="#">B.Comp. Dissertation</a>	12	Computer Science Breadth and Depth		Y4	
<input type="checkbox"/>		#VALUE!	4	Computer Science Breadth and Depth			
<input type="checkbox"/>		#VALUE!	4	Computer Science Breadth and Depth			
<input type="checkbox"/>		#VALUE!	4	Computer Science Breadth and Depth			



# CS1101S – Programming Methodology

- CS Kids intro to CS, taught in the prof's language that is definitely not just javascript (might change)
- Will teach you programming from scratch
- Bans a lot of programming features (variables are not even allowed until many weeks in)
- Tutorial sizes of max 8 students



# MA1521 - Calculus

- A bit more than introduction to multivariable calculus
- Even with H2 Math knowledge this is not an easy course

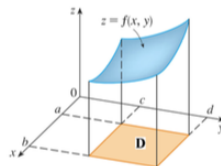
## Functions of 3 variables

$f(x, y, z)$	$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$
Gradient $\nabla f$	$\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \rangle$ steepest ascent normal to the surface $f(x, y, z) = C$
Directional Derivative	$D_{\mathbf{u}}f = \langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \rangle \cdot \mathbf{u}$
Tangent Plane to $f(x, y, z) = C$	$\nabla f(a, b, c) \cdot \langle x - a, y - b, z - c \rangle = 0$

## Double integrals

Double integral as the (algebraic) volume of the solid region under the graph of  $z = f(x, y)$  over  $D$ .

$$\text{Volume} = \iint_D f(x, y) dA.$$





## **GEA1000 - Quantitative Reasoning with Data**

- Basically statistics, excel, and probability
- A good portion of NUS has to take this course, including business, engineering, and social science students, so you are competing against them.
- I would say SOC students have an advantage in this mod due to being somewhat inclined in this field

# CS1231S – Discrete Math

→ Covers topics from Predicate logic to set and graph theory

→ Good foundation for future CS

Courses like CS2040S

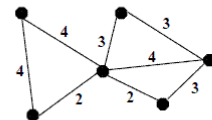


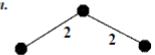

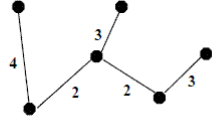
D2. Draw a truth table to show that the following statement is a tautology:

$$((a \rightarrow x) \wedge (b \rightarrow y)) \rightarrow ((a \wedge b) \rightarrow (x \wedge y)).$$

D3. Use the laws given in **Theorem 2.1.1 (Epp)** and the **implication law** to prove that the following are tautologies.

- (a)  $(p \wedge q) \rightarrow p$
- (b)  $((p \vee q) \wedge \sim p) \rightarrow q$
- (c)  $((p \rightarrow q) \wedge p) \rightarrow q$
- (d)  $(\sim p \rightarrow (q \wedge \sim q)) \rightarrow p$

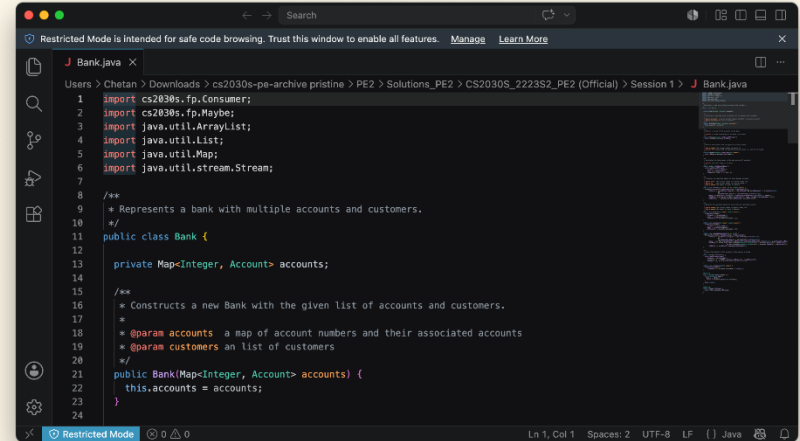
## Prim's Algorithm

<p>1 Given a network.....</p> 	<p>2 Choose a vertex</p> 	<p>3 Choose the shortest edge from this vertex.</p> 
<p>4 Choose the nearest vertex not yet in the solution.</p> 	<p>5 Choose the next nearest vertex not yet in the solution, when there is a choice choose either.</p> 	<p>6 Repeat until you have a minimal spanning tree.</p> 

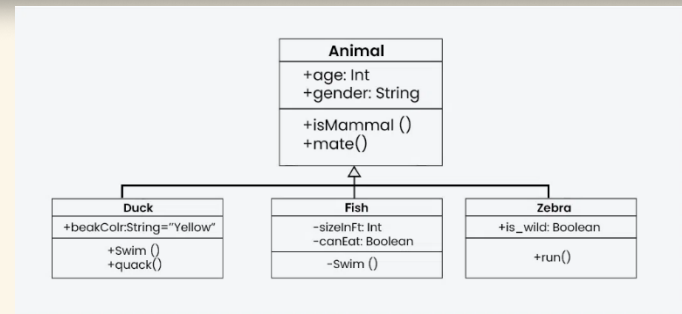
	P	Q	R	(P → Q) ↔ (~ R → Q)
1	T	T	T	T T T F T T T
2	T	T	F	T T T T F T T
3	T	F	T	T F F F T T F
4	T	F	F	T F F T F F F
5	F	T	T	F T T F T T T
6	F	T	F	F T T T F T T
7	F	F	T	F T F F T T F
8	F	F	F	F T F T F F F

# CS2030S – Programming Methodology II

- One of the toughest courses I have taken so far
- But also one of the most interesting
- Covers important coding concepts like abstraction barrier, Object Oriented Programming and code maintainability in the Java programming language



```
1 import cs2030s.fp.Consumer;
2 import cs2030s.fp.Maybe;
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.Map;
6 import java.util.stream.Stream;
7
8 /**
9  * Represents a bank with multiple accounts and customers.
10 */
11 public class Bank {
12
13     private Map<Integer, Account> accounts;
14
15     /**
16      * Constructs a new Bank with the given list of accounts and customers.
17      *
18      * @param accounts a map of account numbers and their associated accounts
19      * @param customers an list of customers
20      */
21     public Bank(Map<Integer, Account> accounts) {
22         this.accounts = accounts;
23     }
24 }
```



Class Diagram example



# CS2040S – Data Structures and Algorithms

- One of the most important courses in NUS
- Basically what interviewers ask in job interviews
- Covers common DSAs like stacks, Queues, AVL Trees, Hash Tables, SSSPs, DP and so on

10 Data Structures Used in Daily Life ByteByteGo

Data Structure	Illustration	Use Cases
List		Twitter feeds
Array		Math operations Large data sets
Stack		Undo/Redo of word editor
Queue		Printer jobs User actions in game
Heap		Task scheduling
Tree		HTML document AI decision
Suffix Tree		Search string in document
Graph		Friendship tracking Path finding
R-tree		Nearest neighbour
Hash Table		Caching systems



# Results out, now what?

If you are happy with your results, congrats!

If not, don't worry!



# S/U system

- For courses without any pre-requisites, you can nullify their grade (make them pass/fail) and have it not affect your GPA!
- You have 32 units of S/U over your candidature, with 20 units expiring after the end of Sem 4 (Y2S2).



# 03

## Other Info / Summer and Winter Activities





# Winter School!

I didn't go, but a lot of my friends did

Take 1 mod in Korea / Japan / any country during winter break

Clear it without it affecting your grades

Usually people take MA1521 or MA1522 there

Need to pay out of your own pocket

# Some Tips

- try not to overload in Y1S1 (max. 23 units unless double major/degree, etc.)
- some people find that Y1S2 is a good time to overload! But don't overload just because everyone is overloading
- take courses that can be SU-ed in Y1 and Y2
- helps to pick courses with friends since some require you to do group projects or can also help each other cross-check answers or explain tricky concepts

# Some Tips

- don't need to start learning coding pre-university! But important to keep up with coding courses – don't let your doubts accumulate as it may be more difficult to keep up with everything towards the end...
- learn to google! Not just for academics, but to look for resources online (e.g. course reviews, course requirements, NOC, SEP, internships)
- check your nus emails and telegram groups, there are regular updates on workshops, company visits, fun activities, hackathons and more!! (@nuscomputingclub on telegram)

# Orbital

- only available during y1 summer
- design your own project, can be a web, mobile, game, tele bot etc.
- 2-people team, can choose partner
- orbital proposals start mid y1s2 so can start thinking of your project idea and partner early!
- almost everyone does (including us) because you learn a lot! (and very good on portfolio)



# NOC

- **Don't think, just TRY if you WANT.**
- keen about startups & entrepreneurship you will never be prepared for it, just shoot your shot.
- 3 months, 6 months and 1 year
- academically and financially neutral
- get to intern in a startup and learn a lot



# Hackathon

- just do for fun and experience and join your friends
- life is not about winning its about doing - parama, 2024



# Internships

- don't need to get an internship in year 1!!!
- normal to be chill or do your own projects or just live life / go on hackathons with friends
- usually ppl intern in y2 / y3, but if you wanna try interning you can! but if you don't get it, its okay bros. (I didn't get any yet...)
- you need apply at least 3 months ahead for tech roles tho

# FOP

- apply during end of sem 1 (mid nov before finals) for subcomm (earlier for pd/vpd)
- apply during start of sem 2 (mid jan) for OGLs/house heads/GMs
- everything is fun!!! interview is fun, gatherings are fun, dry runs are fun!!!



# CVWO (Computing for Voluntary Welfare Organisations)

- very tough to get in! ( I also didn't get into this...)
  - Work as Full Stack Developer
- Prof Ben Leong oversees this project  
mini project in winter (take home  
holiday assignment for the interview)
- work with voluntary welfare organisations (VWOs) to develop software systems

# Clubs and Interest Groups



**NUS STUDENTS' COMPUTING CLUB**  
Represents undergraduates in NUS SoC



**DEVELOPER STUDENT CLUBS (DSC)**  
Community group for students to grow their knowledge



**FINTECH SOCIETY**  
Focuses on research on machine learning and blockchain



**GAMES DEVELOPMENT GROUP**  
Student interest group for designing of games



**GREYHATS**  
Spark student's interest in information security



**HACKERS**  
Student run organization committed to the spread of hacker culture & free/open source software

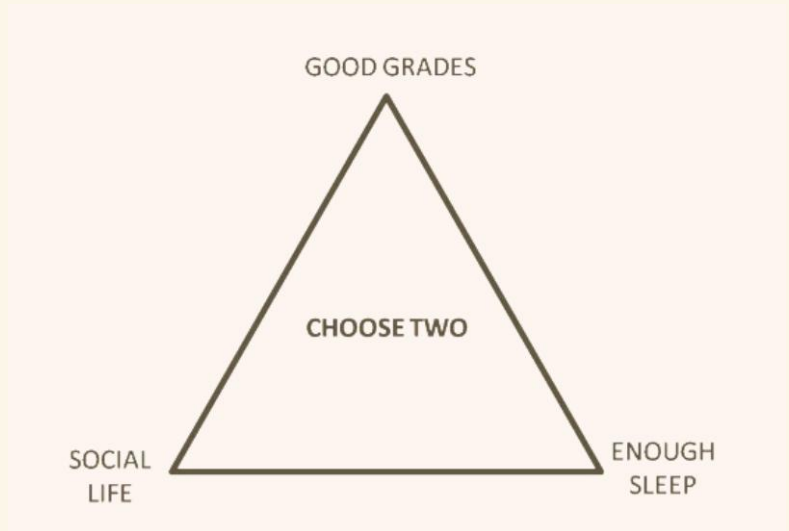


**STARTIT**  
Platform for students passionate in the field of IT and technology to expand their expertise



**WOMEN IN TECH (incubatee)**  
To empower members with networking opportunities, resources and advocacy

# What to Expect in SoC



It's very hard to maintain all 3 at once, you will need to be very talented and disciplined

Otherwise, settle for 2

# What to Expect in SoC

- if you are not doing well for studies, do some internships. can't find internships, do projects, no time for projects, enjoy your uni life
- it is easy to get lost in the rat race, remember to take sufficient breaks (your health comes first)
- speak to your seniors/TAs/Profs, we have been there before
- the competition will be extremely tough, especially in NUS SoC
- try not to be discouraged if you fall, instead keep working towards small actionable goals
- life's short paddle hard, explore your interests, learn and experience as much as you can



# 04 Student Life & Opportunities



# Computing Club – Open House 26





# Thanks!

Does anyone have any questions?

Tele: @rocksontheshore



**CREDITS:** This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#)