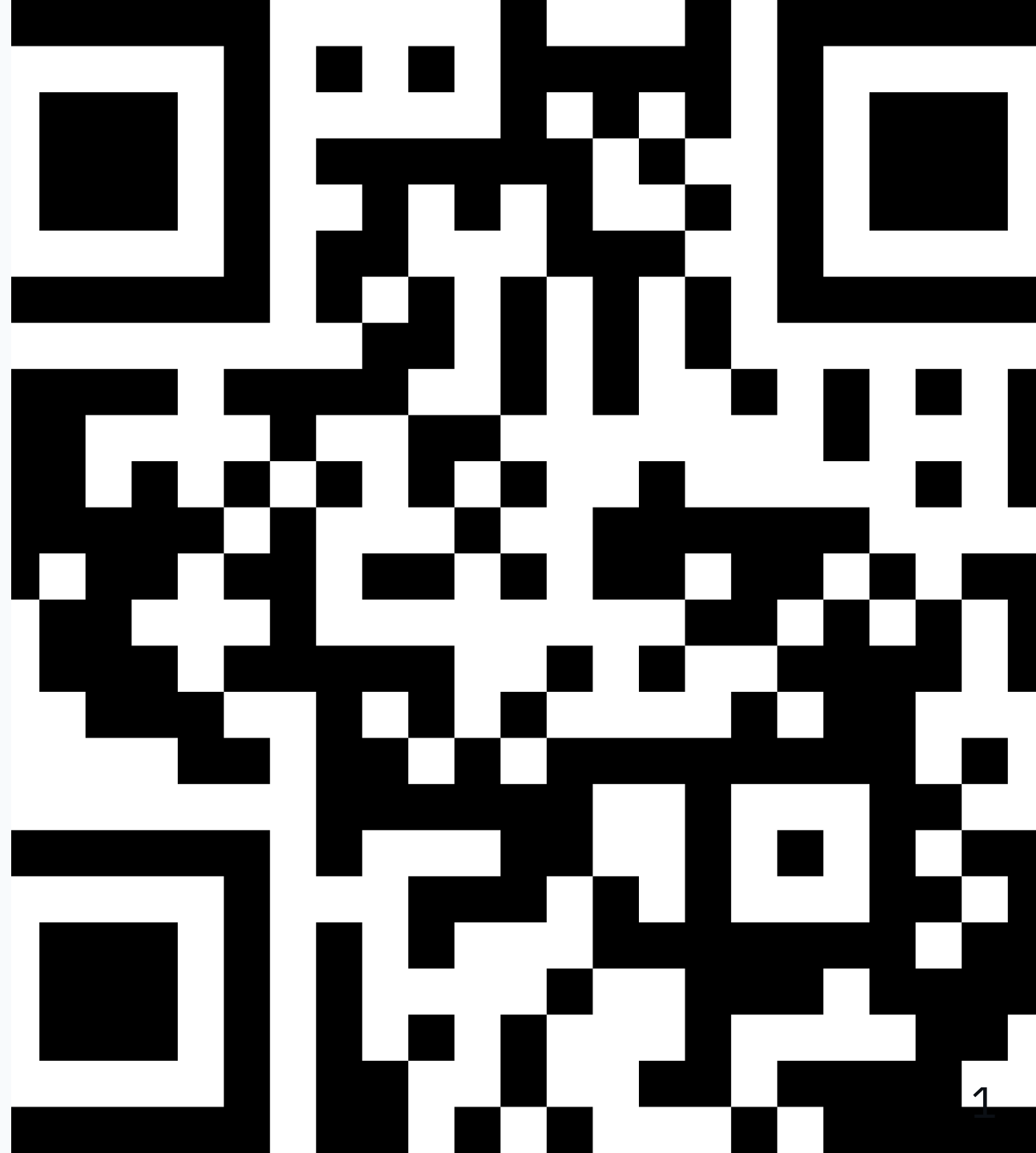# Join Discord Server

Introduction to Computer Science

BCS1110

Dr. Ashish Sai

📅 Week 1 Lecture 1

💻 bcs1110.ashish.nl

📍 EPD150 MSM Conference Hall

Welcome to BCS-110

Plan for Today

- About Us
- What is Computer Scinece?
- Why I love CS?
- Computational Thinking
- Course Logistics

About Us

(Humans of BCS1110)

Part 1/5

Dr. Ashish Sai

Assistant Professor
Department of Advanced Computing Sciences

📍 PHS1 C4.005

✉ ashish.sai@maastrichtuniversity.nl

💻 ashish.nl

Work Experince

- Expert Group Member - Crypto Sustainability, **World Econonmic Forum**
- Research Scholar - **University of California, Berkeley**
- Lecturer - **University of Amsterdam**
- Teaching Fellow - **Trinity College Dublin**

Teaching Assistants

Awaab Hussein

Derrick Timmermans

Sorin Betisor

Denis Gafarov

Tomasz Mizera

Igor Swierlikowski

Pedro Vaz Serrao Santos

Irina Iarlykanova

Junior Lecturer

Shi Qiu

What is Computer Science

Part 2/5

Computer Science: An Evolving Discipline 🖥️

What is Computer Science?

- Difficult to define🤷:
  - Evolving nature of the field
  - Broad scope covering diverse disciplines (e.g., mathematics, engineering, linguistics)
  - Deep interweaving of theory and applications 🔗

Computer Science: A Science Viewpoint🔬

- **Definition**: Study of algorithms, computation, and information processing 🧮

- **Emphasis**: *Understanding theoretical foundations and problem-solving* 💡

- **Scope**: Investigates algorithmic complexity, computability, and mathematical nature of computing systems

Computer Science: An Engineering Viewpoint 🛠️

- **Definition**: Focus on designing and developing computer systems and applications

- **Emphasis**: *Practical implementation, optimization, and building technologies* 🔧

- **Scope**: Includes hardware design, software development, networking, and user interface design 🌐

The Holistic View of Computer Science 🌐

- Embraces diverse methodologies from multiple disciplines

- Combines **mathematical rigor**, **scientific inquiry**, and **engineering methodologies** to innovate

- As computer scientists, understanding this interplay helps us in creating cutting-edge technologies with real-world impact 💡

Computer Science is <u>not only</u> Programming

- Programming is an essential part, but computer science opens doors to a wide array of exciting fields and opportunities beyond programming

Themes in Computer Science 📝

Broadly speaking, there are three disciplines in CS^ :

1. **Hardware**: Focus on physical computer components and systems

2. **Software**: Diverse applications, systems, and development tools

3. **Theory**: Study of algorithms, computability, and cryptography

  ^ *Please note that this list is not exhaustive*

Computer Systems: Hardware 🔧

- **Hardware**: Physical components of a computer and its supporting devices
- **Subfields**: Computer Architecture, Circuit Design
- **Career Opportunities**: Hardware Engineer, Computer Architect
- **ASML (Circuit Design, Semiconductor Manufacturing)**

ASML PHILIPS

Software: Applications, Systems, and Development 💻📊🛠️

Applications Software

- Programs that perform various tasks for users
- **Subfields:** Web Development, Mobile App Development
- **Careers:** Web Developer, Mobile App Developer

Systems Software

- Programs that directly control computer hardware
- **Subfields:** Operating Systems, Device Drivers
- **Careers:** Systems Administrator, Device Driver Developer

Development Software

- Programs used to create other software applications
- **Subfields:** Integrated Development Environments (IDEs), Version Control
- **Careers:** Software Engineer, IDE Developer

Theory: Algorithms, Computability, and Cryptography 🔬🔐🔍

Algorithms

- Study of step-by-step procedures for problem-solving
- **Subfields:** Algorithm Analysis, Data Structures
- **Careers:** Algorithm Developer, Data Scientist

Computability

- Investigates the power and limitations of computation
- **Subfields:** Computational Complexity Theory, Formal Languages
- **Careers:** Theoretical Computer Scientist, Researcher

Cryptography

- Ensures secure communication and data protection
- **Subfields:** Encryption, Cryptanalysis
- **Careers:** Cryptographer, Security Analyst

Applications of CS 📱

Computer Vision	**Natural Language Processing**
Virtual Reality	**Robotics**

Social Aspects of Computer Science 🌐 🤝

- Addressing the societal impact of automation, privacy, and ethical considerations

- Ensuring technology benefits society while avoiding harmful effects

Required reading: *80 Million Tiny Images dataset by MIT* (on Canvas)

The Beauty of Computer Science

Part 3/5

I think CS is an extraordinary field ✨💡

- Combines **logic** and **creativity**, **structure** and **chaos**, **standardisation** and **non-standardisation**

- Builds something from nothing and solves previously unsolved problems

- The potential of Computer Science seems limitless, constrained only by our own **creativity**
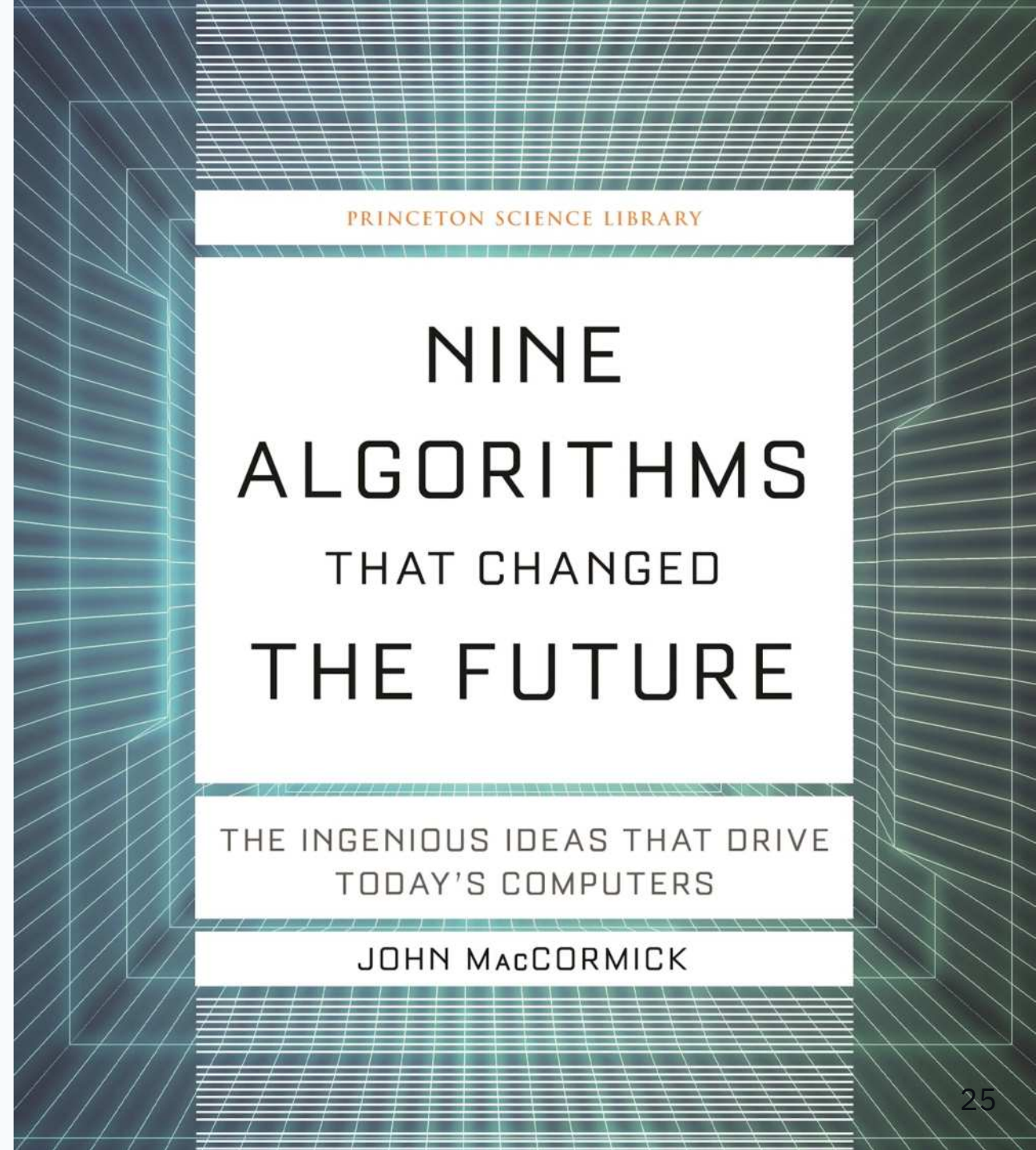
Intersection of Logic and Creativity

Part 3.1

Applying Logical Thinking to Create Innovative and Creative Solutions 🧩✨

- Finding **logical patterns** and **principles** to drive **innovation**

- Utilizing **principled engineering techniques** to design **efficient** and **effective** solutions

Examples ⭐

- Search engine indexing
- PageRank
- Public-key cryptography
- Forward error correction
- Pattern recognition
- Data compression
- Database
- Digital signature
- Computability

PRINCETON SCIENCE LIBRARY

NINE ALGORITHMS THAT CHANGED THE FUTURE

THE INGENIOUS IDEAS THAT DRIVE TODAY'S COMPUTERS

JOHN MacCORMICK

Structure and Chasos

Part 3.2

CS requires organizing complex systems and ~~data structures~~ while handling unpredictable events and edge cases.

How a communications failure crippled the Dutch rail network

Report by The RailTech.com Link

Standardisation and Non-Standardisation

Part 3.3

Standardised and Non-Standardised Concepts 🖥️🔍

Computer Science involves **standardised (concrete)** and **non-standardised (abstract)** concepts

Standardised Concepts

- Specific, well-defined elements

- Precise and consistent, enabling interoperability

Non-Standardised Concepts

- Generalized ideas and theoretical principles

- Allow flexibility and innovation

The human was supposed to have clicked DRILL - PACOM (CDW) - STATE ONLY that morning but accidentally clicked PACOM (CDW) - STATE ONLY instead, thereby sending an actual alert

1. State EOC

1. TEST Message

DRILL-PACOM (DEMO) STATE ONLY

False Alarm BMD (CEM) - STATE ONLY

Monthly Test (RMT) - STATE ONLY

PACOM (CDW) - STATE ONLY

Input

```html
<!DOCTYPE html>
<html>
<head>
  <title>Yoda Concreteness Example</title>
  <style>
    .my-yoda {
      color: green;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1 class="my-yoda">
    "Strong in the ways of the Force, HTML and CSS must be. <br>
    Syntax and semantics, correctly you must follow. <br>
    Concrete rules, they are. Applied they must be, <br>
    to style and structure your web pages."
  </h1>
</body>
</html>
```

Output

"Strong in the ways of the Force, HTML and CSS must be.

Syntax and semantics, correctly you must follow.

Concrete rules, they are. Applied they must be,

to style and structure your web pages."

Input

```html
<!DOCTYPE html>
<html>
<head>
  <title>Yoda Concreteness Example</title>
  <style>
    .my-yoda {
      colour: green;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1 class="my-yoda">
    "Misguided, the web page becomes when
    syntax and semantics are ignored. <br>
    Deviating from the concreteness
    requirement leads to chaos, it does. <br>
    Styling and structure, lost they are,
    rendering confusion and frustration, they will."
  </h1>
</body>
</html>
```

Output

"Misguided, the web page becomes when syntax and semantics are ignored.

Deviating from the concreteness requirement leads to chaos, it does.

Styling and structure, lost they are,

rendering confusion and frustration, they will."

34

Building Something New

Part 3.4

Creating Something New in CS 🌟

- Even when utilizing existing knowledge and resources, computer scientists are constantly creating something new

- They learn and explore new concepts, techniques, and technologies to build innovative solutions

Predictive policing- Data science in politiewerk

https://kombijde.politie.nl/vakgebieden/ict/predictiv

Software is Magic ✨

- Software is often regarded as the closest thing to actual magic

- It transforms simple instructions into limitless possibilities

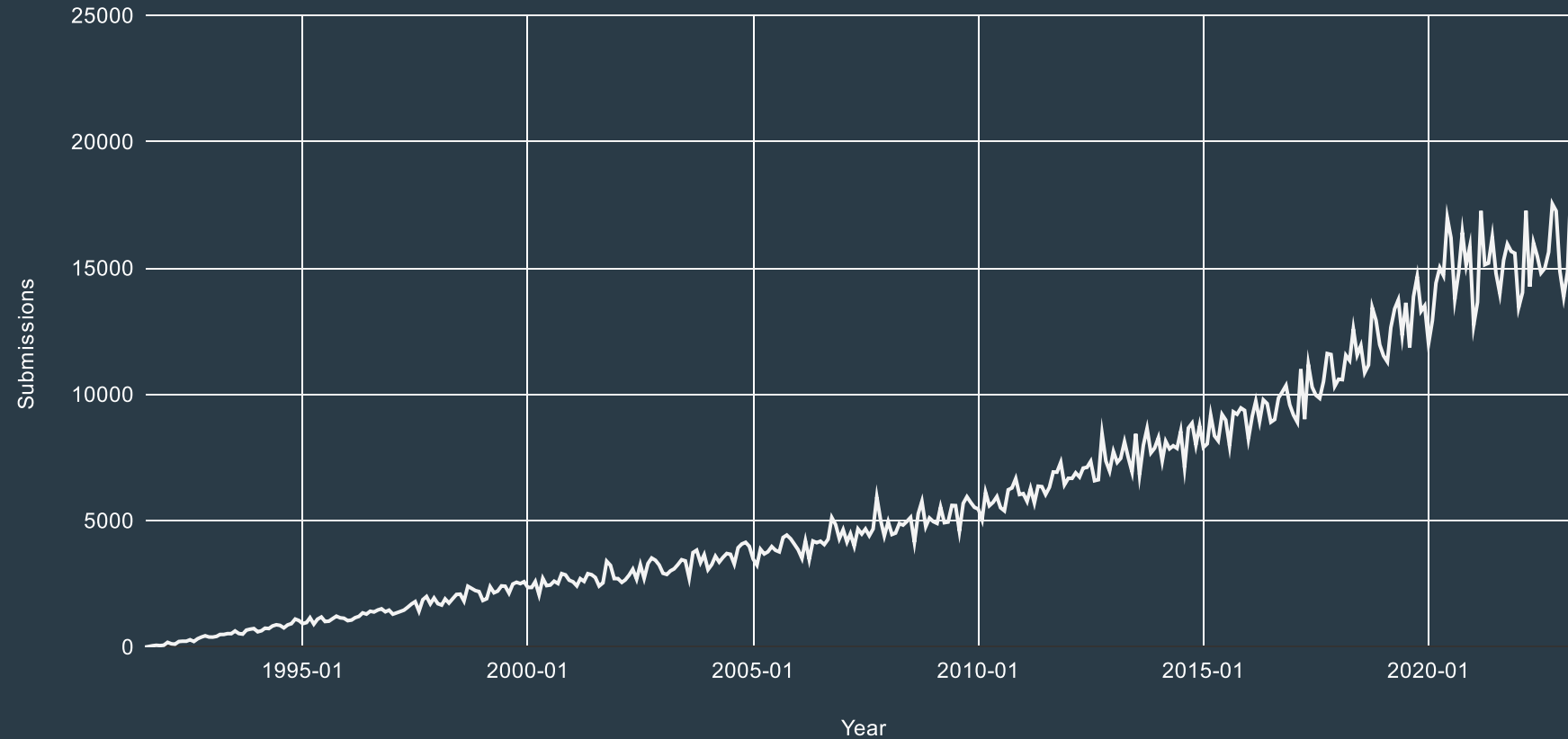- Enables machines to perform complex tasks at an unprecedented scale

ICD

"An implantable device that monitors and treats life-threatening heart rhythm abnormalities through electrical shock therapy"

The Early Stages

Part 3.5

Submissions to arXiv

We are continuously exploring and learning, like banging sticks together, not yet composing symphonies

Short Break

Do not leave your seat (5 min)

What will you learn from BCS1110?

The process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes.

Computational Thinking

Part 4/5

Problem Solving

Introduction to Problem-Solving 🧩

- Problem-solving is the process of transforming an **undesirable state** into a **desirable one**.

- Real-world problems are often **complex** and require a **systematic approach**.

- Following a **guide or process** can help in tackling complex tasks effectively.

Pólya's Systematic Approach 📝

- George Pólya's problem-solving approach:
  (Don't give up)

  1. **Understand the problem**

  2. **Devise a plan**

  3. **Execute the plan**

  4. **Review and extend the solution**

- Pólya's method is inspired by the traditions of
  mathematical and **natural sciences**

Computational Thinking

Part 4/5

Decomposition and Abstraction

Divide-and-Conquer Strategy 🪓🛡️

- **Divide-and-Conquer**: a strategy of breaking down a large, complex problem into smaller, more manageable parts

- Widely used in various domains:

  - Military 🪖

  - Politics 🏛️

  - Computer Science 💻

Other Effective Problem-Solving Strategies 🧠💡

- **Critical thinking** → Questioning ideas and justifying decisions

- **Solving a concrete instance** → Simplifying problems with specific examples

- **Finding related problems** → Examining solutions to analogous problems

- **Working backward** → Starting from the goal and deducing steps backward
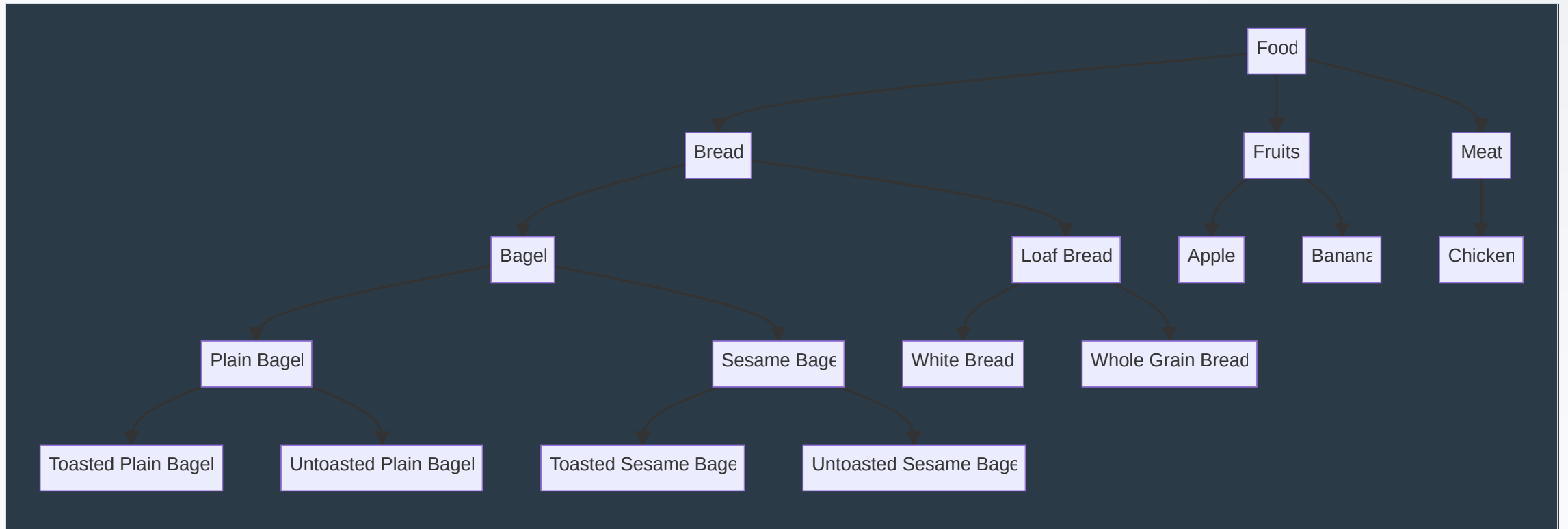
Abstraction

Abstraction is a way to simplify complex systems by focusing on the high-level overview rather than the nitty-gritty details. It allows us to understand and solve problems more efficiently by removing unnecessary information
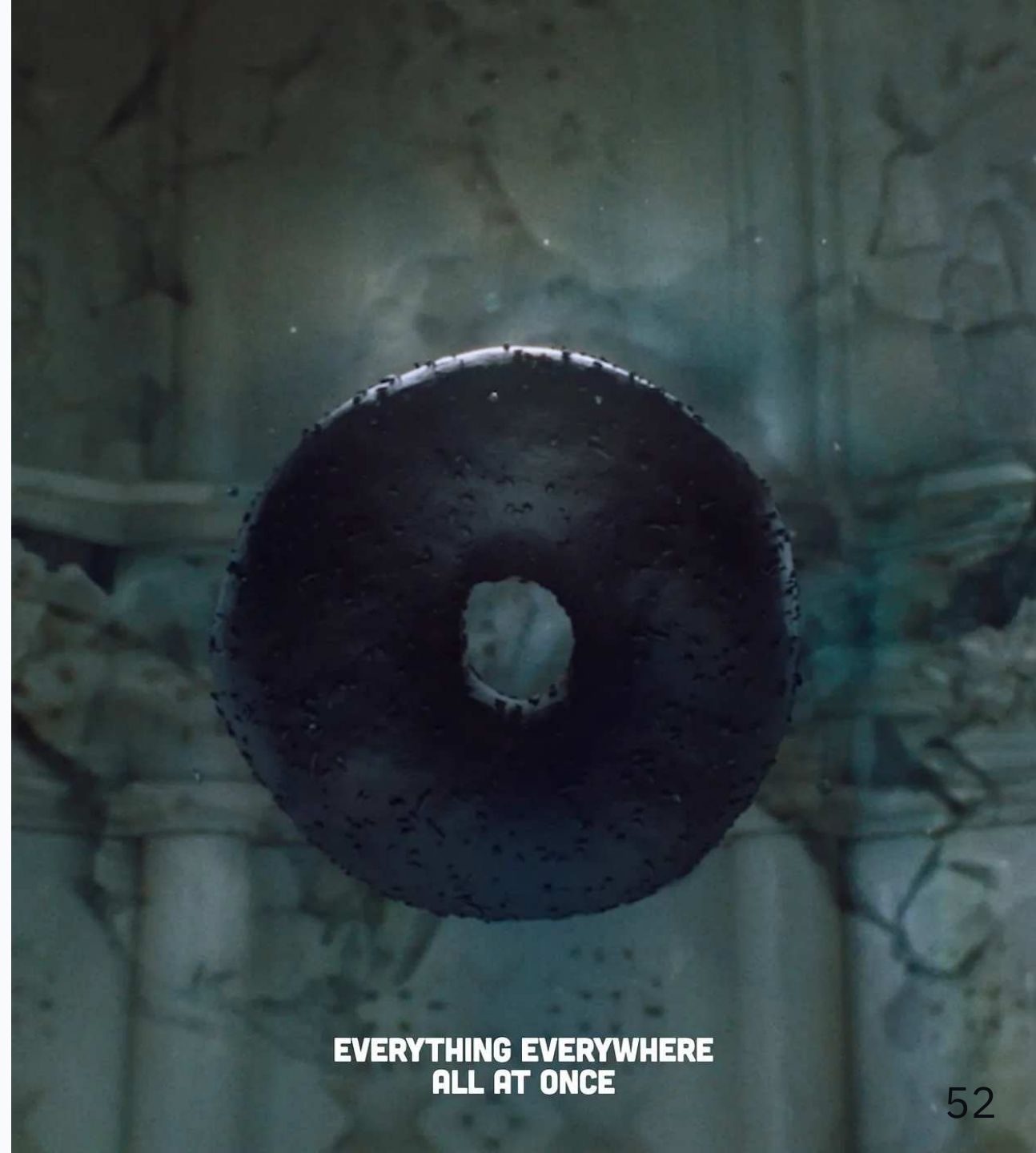
EVERYTHING EVERYWHERE
ALL AT ONCE

Abstraction

Course Overview

Part 5/5

Course Philosophy 🎓

- Most introductory courses focus on programming proficiency but often overlook computational thinking

- We start with a complete system and explore various aspects of computer science.

- So less focus on programming and more focus on computational thinking

Essential Concepts

- Algorithms
- Computing hardware
- Models of computation
- Computer networks
- Cyber Security

Course Schedule 📅

Topic	Lectures	Lab
Week 1: Introduction	2 Lectures	1 Lab
Week 2: Algorithm and OS	2 Lectures	1 Lab
Week 3: Theory of Computation	2 Lectures	1 Lab
Week 4: Computer Networks	2 Lectures	1 Lab
Week 5: Cyber Security	2 Lectures	1 Lab
Week 6: Exam Prep	2 Q&A Sessions	No Lab

Grading 📊

Assignment	Points	Percent
Project	25	25%
Final Exam	75	75%
Total	100	–

Grade	Range
10	95–100%
9	85–94%
8	75–84%
7	65–74%
6	55–64%
F	<55

To pass the course, you need to get more than **60%** in total

Budget ⏳

Budget	Number	Time Each (hrs)	Total (hrs)
Class time			
Lectures	10	2	20
Labs	5	2	10
Revision class	1	2	2
Q&A session	1	2	2
Total in class time			34
Remaining budget			78
Outside class time			
Practice exams	1	3	3
Going through lecture slides/notes	10	5	50
Revision			26
Total outside class			78
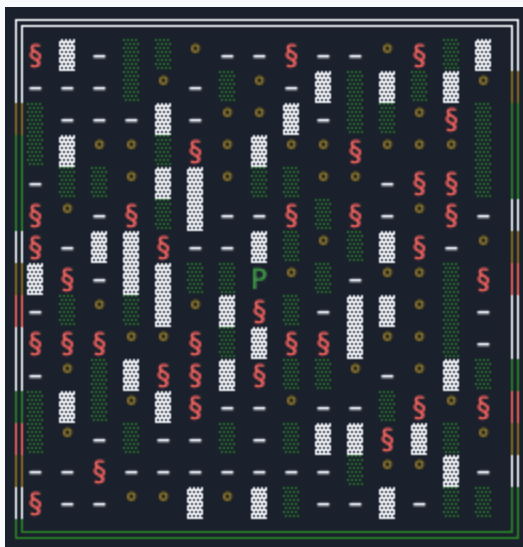
JAVA CRAFT

BCS1110

Not at all inspired by minecraft

What do we expect from you?

Part 5/5

Programming Expectations 👨🏽‍💻

- You follow **BCS1120** and learn **Java** ☕ (or you already know how to work with Java)

- You will need to write some code but most importantly you need to **understand** and **modify** existing code in Java
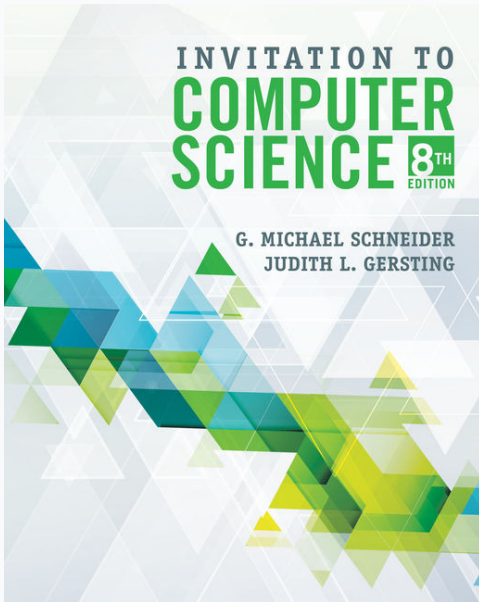
Attendance and participation 🎯

- You are expected to come to the lectures each week 🏢 ^

- You also have to attend your 🥼 labs on Friday

^ I strongly recommend that you attend all the lectures and labs

Course Material 📚

You do not need to buy the book, I will provide you all the information you need within the lecture notes

Other materials

- I will occasionally also use two other text (see course page)
- (No need to buy these either)

Java and VSCode

- We piggyback on BCS1120's setup
- Use VSCode for the project

Important pep talk! 💬

- I promise **you can (and will) succeed** in this class
- I'm *fully committed* to making sure that you **learn everything** you were hoping to learn from this class!

Support

Support from me 🤝

- I will make whatever **accommodations** I can to help you learn and understand the class material and finish the project

- If you tell me you're having trouble, I will **not judge you** or think less of you.
  I hope you'll extend me the same *grace*

- You are always welcome to **talk to me** about things that you're going through.
  If I can't help you, I usually know somebody who can

If you need extra help, or if you need more time with something, or if you feel like you're behind or not understanding everything, do not suffer in silence! Talk to me! I will work with you. I promise.

Student hours ⏳

- Student hours are **set times dedicated to all of you** (most professors call these *"office hours"*; I don't)

- This means that I will be in my office 🏢 **PHS1 C4.005 Friday** from *12:00 to 12:30* waiting for you to come by and talk to me with whatever questions you have

Course Policies

Simple: Be Kind, Be Nice and Be Considerate

Class Policies 📄

🚫 Class Policies

- We do not tolerate **discrimination** and/or **violence** of any sort
- We live in a world with a long history of **racism** and need to actively combat that in both our **actions** and **language**, so please be mindful

🎓 Academic Honesty

- Violation of UM's **Policy on Academic Honesty** will result in a **Fail** in the course
- Possible **disciplinary action**
-   ○ So seriously, just **don't cheat or plagiarize!**

Special Needs

- Please talk to me **this week**

Course Communication 💬

- **Course Website:** 🖥️
  bcs1110.ashish.nl & UM Canvas

- **Discord Server**

- **Email**

E-mail and Discord are the **best ways** to get in contact with me.
I will try to respond to all course-related messages within **24 hours** (really).
But life can be busy and chaotic (including for me!) — so if I don't respond right away, don't worry!

Join Discord Server 🔗

- The link is only valid for **7 days** → request a new link via **e-mail** after that

- Read the **Discord policy** on the course webpage!

Course Webpage 🌐

- Everything you need is already on **Canvas** (no need to use the website if you don't want to)
- I post my notes on the webpage that contain **additional readings** (optional)
- You need a username (**BCS1110**) and 🔐 password (**DACS**) to use the webpage

👉 bcs1110.ashish.nl

Remember 🙌: I am here to support you if you need it.

👋 Let's have a great semester!