CPEN 322
Software Construction | |

Ali Mesbah

Today's Objectives

- Getting to know each other
- What is this course about?
- How to successfully pass this course

Instructor

Lecturer: Ali Mesbah

Email: amesbah@ece.ubc.ca

Office: KAISER 4044

Office hours: by appointment

Bio

- Professor @ UBC (2021-now)
- Visiting Scientist @ Google (2017-2018)
- Associate Professor @ UBC (2016-2021)
- Assistant Professor @ UBC (2011-2016)
- Visiting Researcher @ Fujitsu Labs (2010)
- PhD in CS @ TU Delft (2005-2009)
- Software Engineer (2001-2005)

Software AnaLysis and Testing (**SALT**)

http://salt.ece.ubc.ca

Research Group:

- 3 PhD, 3 MSc
- Undergrad Interns

Research:

- Software engineering
- AI4SE: Generative AI
- Software testing and analysis
- Web and mobile engineering

Teaching:

- Software construction
- Software verification
- AI4SE

Partners:

- Collaboration with ~10 different companies


amazon


SAP


intel


hootsuite


OptimusQA
Your Trusted QA Partner


coverity


HUAWEI


Microsoft


FUJITSU


Google

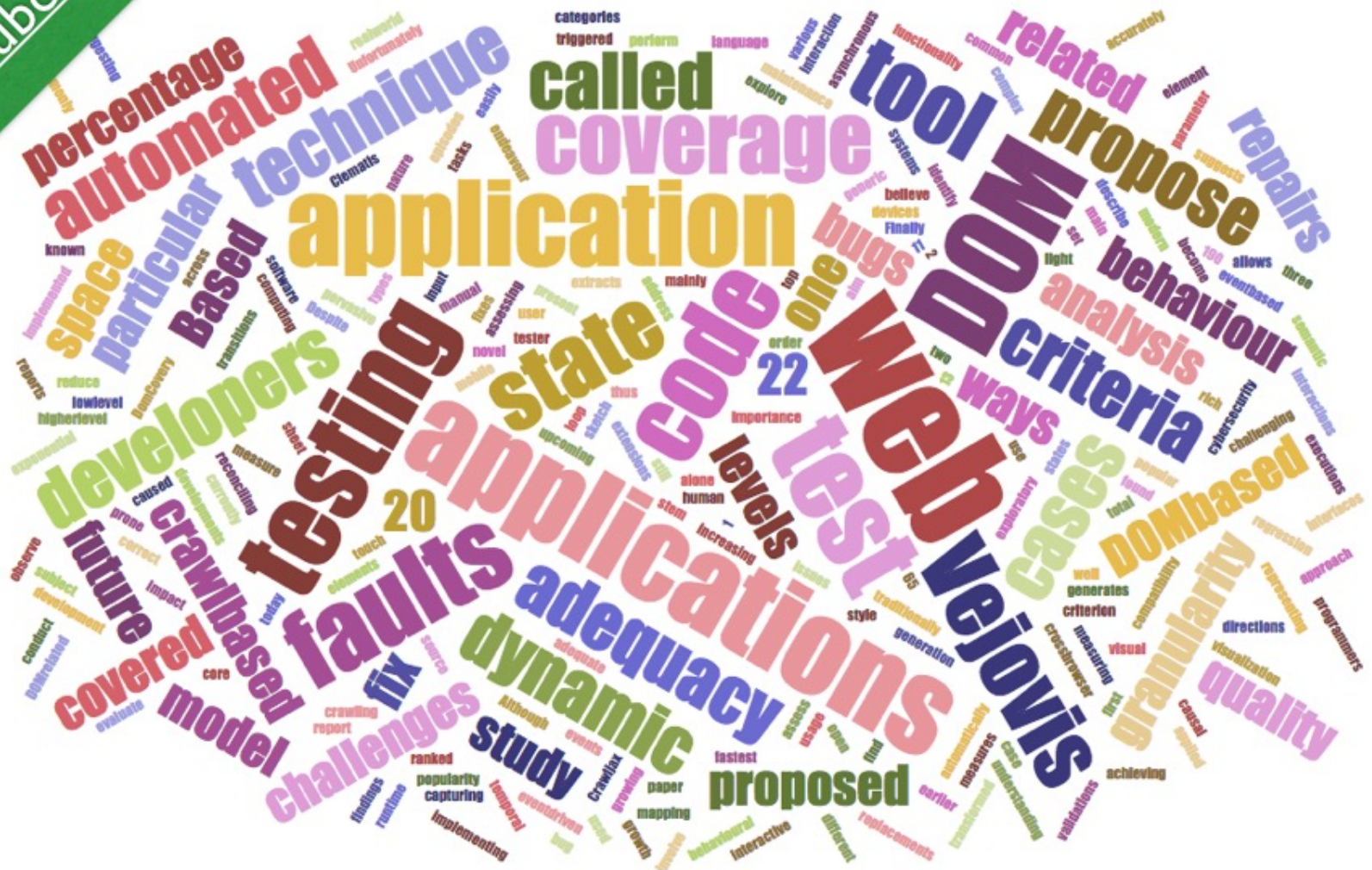

IBM


quickmobile


LOCKHEED MARTIN


FusionPipe
SOFTWARE SOLUTIONS
Fueling the Mobile Internet Revolution

salt.ece.ubc.ca



"automated techniques for supporting software dependability and evolution

Google

- Sabbatical (every 6 years)
- Visiting Scientist @ Google (8 months)
  - Mountain View office

What I liked about my visit @ Google

What I liked about my visit @ Google

- The free food and snacks (of course)!

What I liked about my visit @ Google

- The colorful offices!

What I liked about my visit @ Google

- The amazing software dev. Infrastructure!

Google Developers

My research @ Google

- Learning to fix compilation errors **automatically**
  - Dataset: Failed -> Fixed code changes
  - AI: Neural Machine Translation

- Large-scale Refactoring for Java Type Migration
  - 300 million lines of Java code

Resources

- UBC Canvas
  - Syllabus
  - Announcements
  - Lecture notes
- Piazza: Q&A
- GitHub (for assignments)
  - https://github.com
  - Create an account on GitHub if you don't have one!
  - Submit your account (see Canvas for announcements).

Course TAs

- Nashid Noor
- Kumseok Jung
- Mobina Shahbandeh
- Parsa Alian

Questions about assignments should be asked on **Piazza** only

- For private matters, write a private msg to “Instructors” on Piazza.
- No questions will be answered **48 hours** before a deadline
- **No Email whatsoever - we'll ignore all email messages**
  - **except accessibility-related**

What’s this course about ?

- **Core principles behind constructing** modern ubiquitous applications
- Abstractions and design principles
- Application of technologies such as CSS, HTML, JavaScript, REST APIs, to the above
- Keywords: Saas, Generative AI

What's it not about?

- Learning of specific technologies
  - These will get outdated by the time you finish
  - Fast changing field, so new technologies tomm.
  - Can learn any technology if you understand the principles and concepts behind modern software construction
- Frameworks or libraries (e.g., React)
  - These are built on the principles and concepts
  - Too many to cover in a reasonable time

Why take this course?

- You will understand the **technologies** and **principles** for building **Software as a Service**
- Understand the foundations of leveraging **Generative AI** in modern software construction
- This enables you to design and construct intelligent ubiquitous applications

Why not take this course ?

- You just want to write a lot of code
  - Online tutorials will teach you how to do this
  - While you'll do a series of programming assignments, their focus is to teach you the principles
- You want to learn AI
  - This is not a course on AI / ML
- You want an **easy** elective course
  - You couldn't have gotten it more wrong!

Pre-requisites

- EECE 210 or equivalent (e.g., CPSC 210)
  - Principles of software development
  - Knowledge of invariants, specifications etc.
  - Experience using at least one programming language (e.g., Java, Python)
- Maturity to tackle large software development tasks
- No AI or Web programming experience is needed
  - However, you should be able to pick things up quickly
  - Invest considerable time **outside** of class in learning

Grading

- Class participation: 5%
  - In-class and Piazza
- Assignments: 45%
  - Six lab assignments
- Midterm (28 Feb): 20%
- Final exam: 30%

Lecture Style

- This will vary!
- Sometimes the lectures will be very typical, where I talk on a certain topic
- Mostly **mini-lectures** plus **in-class exercises** -- here's where participation comes in!
- Some will be **async** (watch a video/lecture)
- Presentations by you on your final product (assignment 6)

Reference resources (recommended)

- **Engineering Software as a Service**, by Armando Fox & David Patterson.
- **Building Intelligent Systems: A Guide to Machine Learning Engineering**, by Goeff Hulten.

- Web technologies:
  - **Getting Started with the Web** https://developer.mozilla.org/en-US/docs/Learn/
  - **Eloquent JavaScript**, by Marijn Haverbeke.

Assignments

- First 5 Assignments: build a complete web application from scratch
  - Assignments build cumulatively on each other. Missing even one means you'll lose big!
- Assignment 6 (mini project)
  - Propose an idea: new AI-based feature
  - Assignment 6: build the AI-based feature
- Attendance is **compulsory** at lab sessions when the assignments are due (optional in other sessions)
- To be done in teams of two (Choose a partner in the **same lab session** as you, see Canvas)

Participation

- We will use Piazza for all course-related communication
  - Do NOT email teaching staff unless its an emergency
  - Use private posts for specific situations
  - Use appropriate tags for your posts
- You will receive class participation points for asking good questions and answering questions
- **Class participation:** Showing up is a necessary but NOT a sufficient condition for participation

“Our civilization runs on software”

“Software is eating the world”

Discuss with your neighbors and write down

- What does this sentence mean?
- Can you provide some examples?

“Software is eating the world”

Marc Andreessen (Netscape)

- Digital docs (PDF) ate libraries
- E-mail ate mail services
- Netflix ate video stores
- Expedia ate travel agents
- Amazon is eating retail stores
- Uber is eating taxis
- AirBnB is eating hotels
- …

Software as a Service (SaaS)

Examples: Google Search, Amazon, Netflix, AirBnB, Expedia, ChatGPT, …

What is SaaS?

Software as a Service (SaaS)

- A software distribution model
- Build a software application using standards
- And make it accessible through the Internet
  - Via a web browser

Pros/cons of SaaS

- What are the pros (and cons)?

Benefits of SaaS

- **Ubiquitous**
  - accessible from anywhere
- **Platform independent**
  - Standard-based
  - Runs on any device, os, etc
- **Continuous Delivery**
  - Deploy to the server and every user gets the new version instantly
- **Scalability**: millions of users
- **Monetization**: subscription-based payments

Cons of SaaS

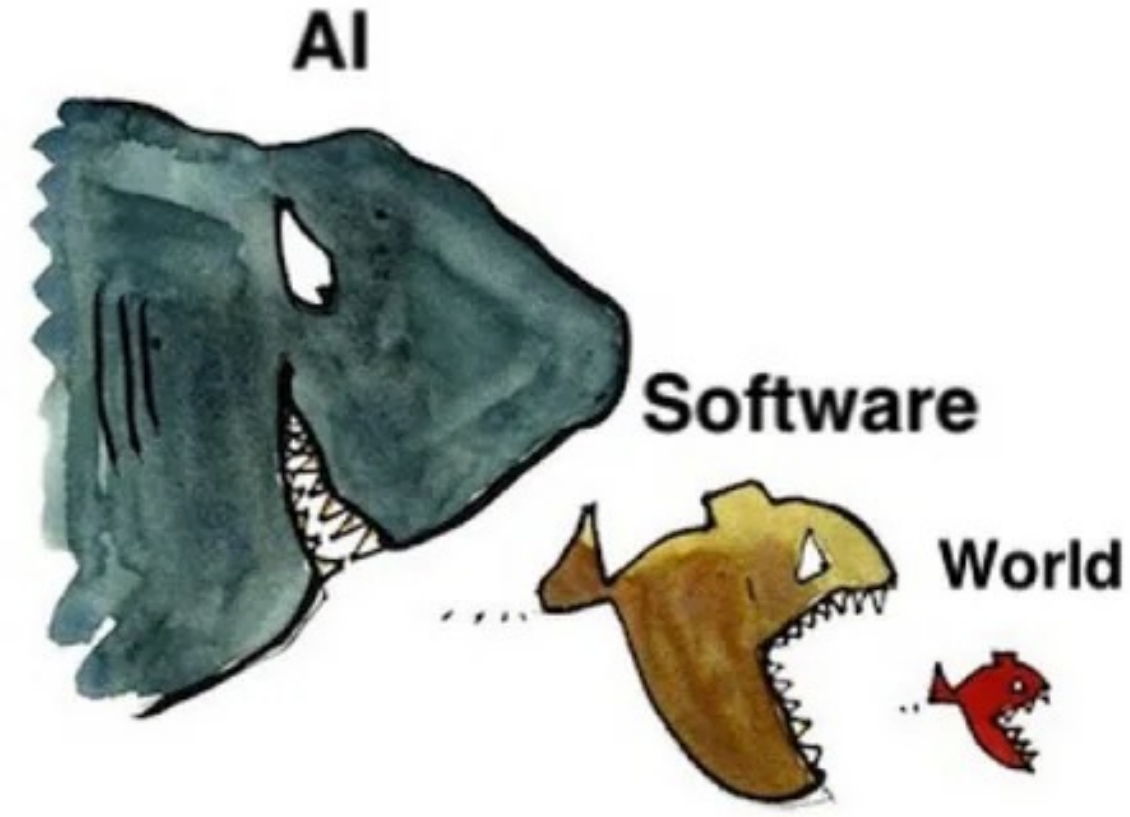
- Complex architecture!
  - Compared to a desktop application
- Heterogeneous: mixture of many software technologies, standards, and languages
- Challenging to analyze and test

Topics Covered

- **Understand Architectural Patterns**: client-server, peer-to-peer
- **Web Page Development**: HTML, CSS, JS, …
- **Asynchronous Programming**
- **Stateless vs stateful programming**: REST API
- **Software as a Service (SaaS) and Continuous Delivery**: Monolithic vs serverless
- **Security, Concurrency, Scalability, and Performance**
- **Testing, Accessibility, and Internationalization**
- and **Generative AI**

“Software is eating the world”

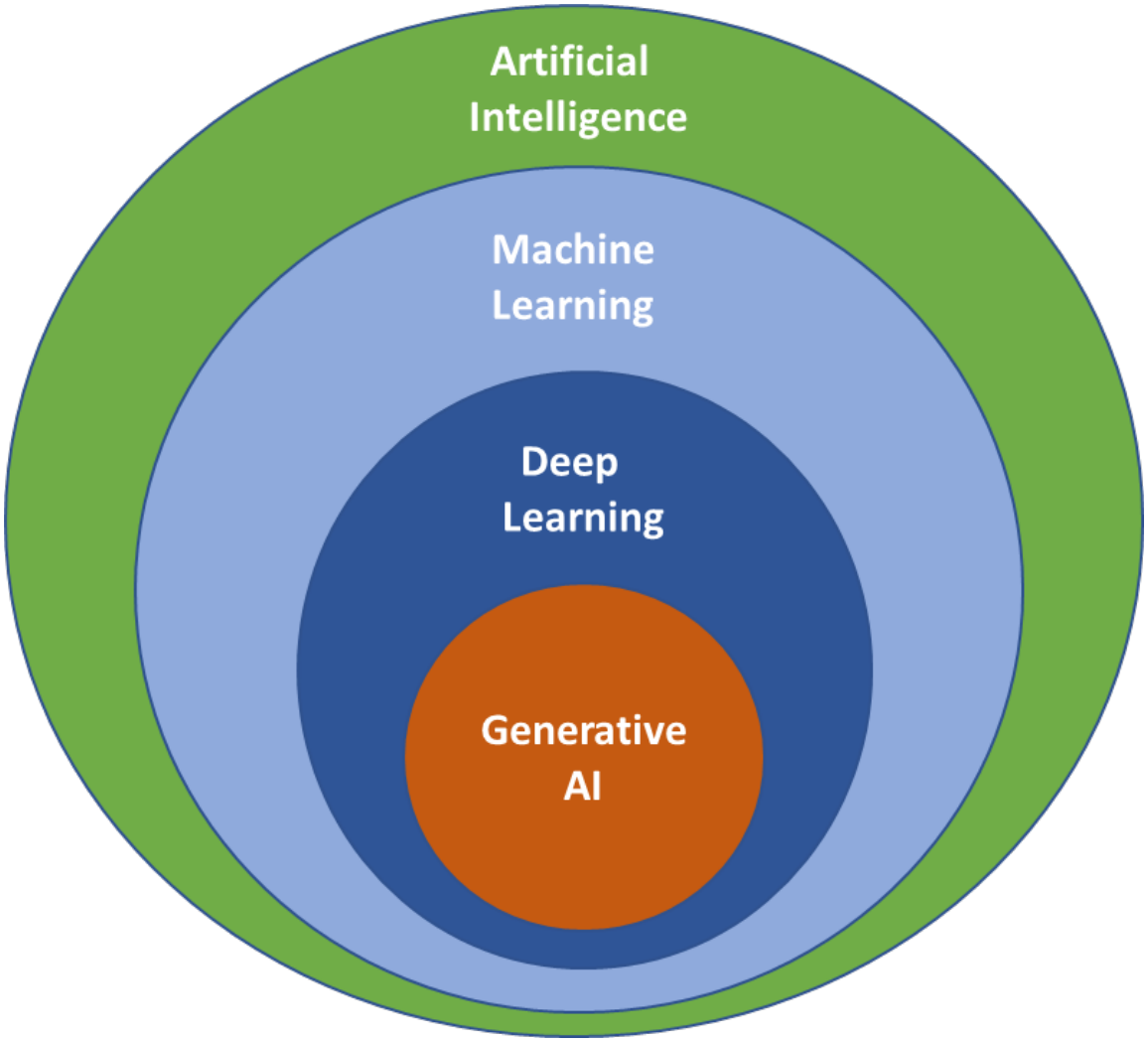
“AI is eating software”



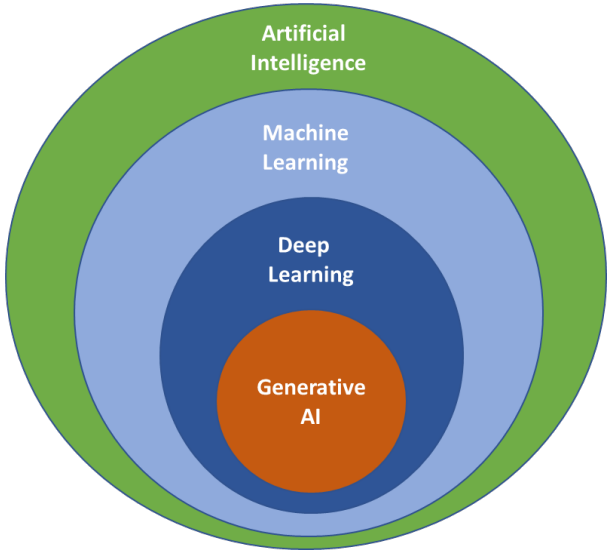
Generative AI

- What is Generative AI?

Generative AI



AI Topics



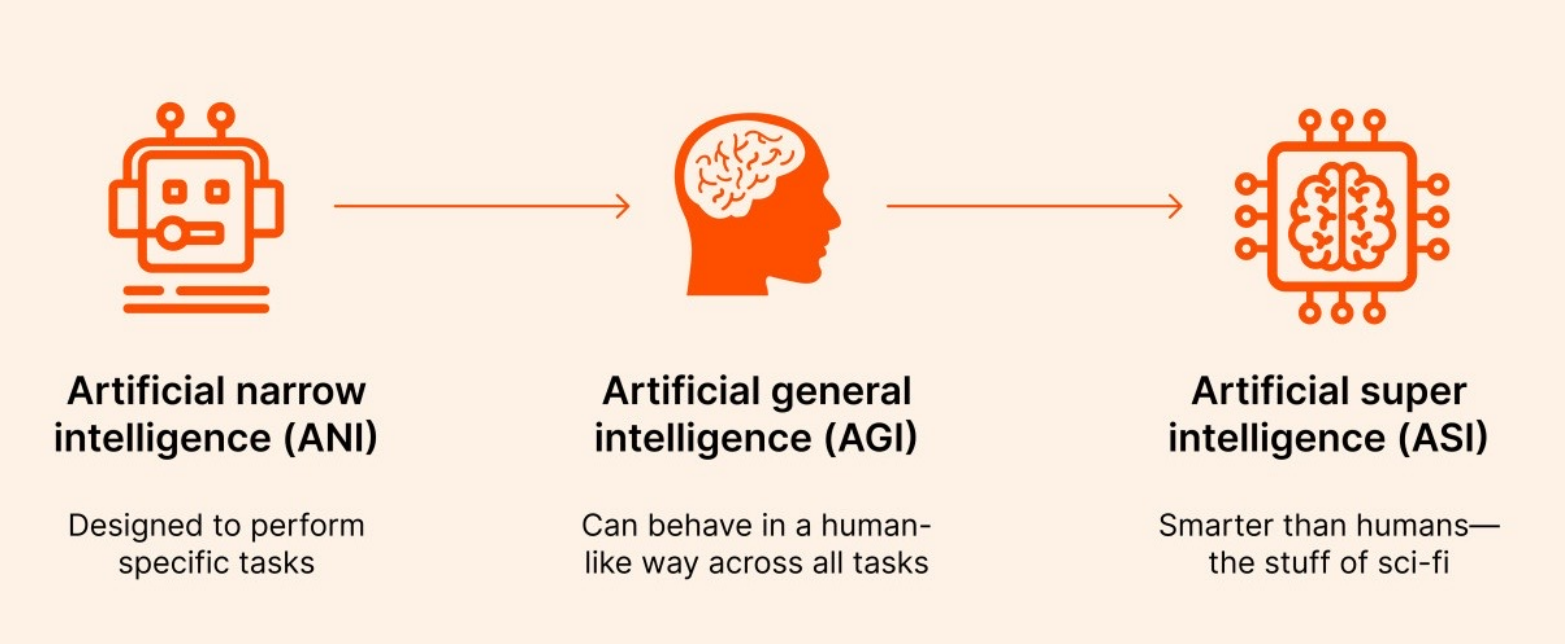
- Understand the potential and limitations of applying AI in software construction.

- Become familiar with AI concepts
  - such as Generative AI, Deep Neural Networks, Foundation Models, Large Language Models, and Prompt Engineering,
  - and their significance in AI-driven software engineering.

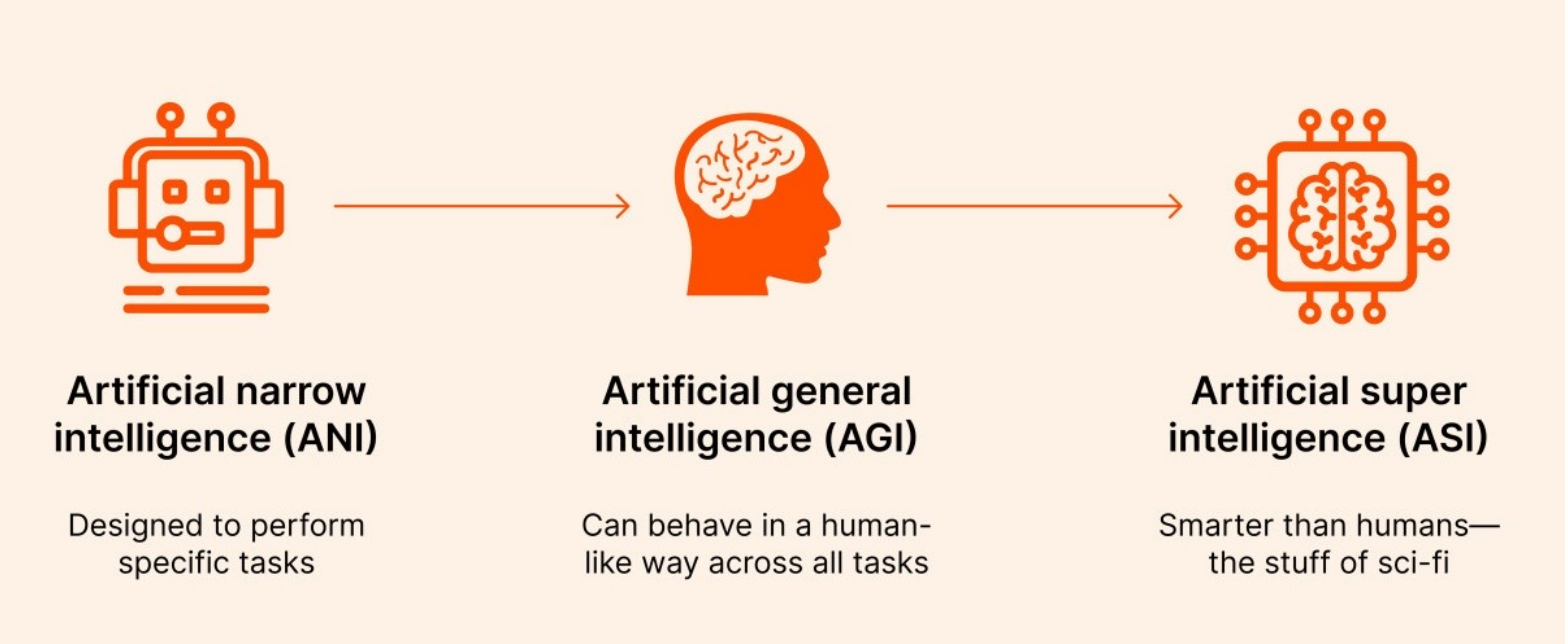
- Integrate basic AI functionalities into software applications (assignment 6).

AGI?

AGI


Artificial narrow intelligence (ANI)
Designed to perform specific tasks
Artificial general intelligence (AGI)
Can behave in a human-like way across all tasks
Artificial super intelligence (ASI)
Smarter than humans—the stuff of sci-fi

Will we ever achieve AGI?



Use of AI in CPEN 322

1. **Authorized Use**: Students may use AI tools such as CoPilot or ChatGPT, for code generation, debugging, or optimization as part of their assignments given they meet the ***conditions*** listed here.
2. **Understanding and Originality**: While AI can assist in generating code or providing solutions, students are expected to understand and be able to explain any code or content submitted. Reliance on AI should not replace the student's own learning, problem-solving, and coding skills.
3. **Attribution**: Any use of AI-generated code, text, images, or insights must be fully and clearly attributed, specifying what portions of the work were assisted by AI; the disclosure must be clear in their reports, code, presentations, etc. For code, explicit code comments are needed indicating the nature of the AI's contribution.
4. **Ethical Considerations**: Students must use AI tools responsibly, considering the ethical implications and avoiding any form of misuse that could lead to dishonesty, plagiarism, or other forms of academic misconduct.

To do..

- Submit your GitHub username via Canvas
- Find a partner to do the assignments with
  - Must be in the same lab session as you–no exceptions
  - Ask them whether they're going to drop the course ☺
  - Both you and your partner get the same marks
  - If you break up with your partner at any point, you need to do the work alone for future assignments (both of you'll get the common code)

Final thoughts

- Do you want to take this course?
  - Involves a significant amount of work and time
  - Easier electives are available
  - If you're dropping, please do so ASAP as waitlist is full
- If you're staying, welcome on board!
  - Feel free to give us suggestions for improvement
  - Tell us what you liked and what you didn't like

Building a SaaS

- You are hired as a software engineer
- The goal is to construct a SaaS application that recommends movies to users based on their profiles and hobbies
- In pairs, devise a plan in which you outline how you would go about achieving the goal.