Syllabus

M1 Overview of AI and ML

- M1-a Introduction to Logic
- M1-b Local Search Techniques, Genetic Algorithms

M2 ML and Data

- Data and generalization, Overfitting, Underfitting, Bias-variance tradeoff, Feature Engineering, Data Preprocessing

M3: ML Concepts

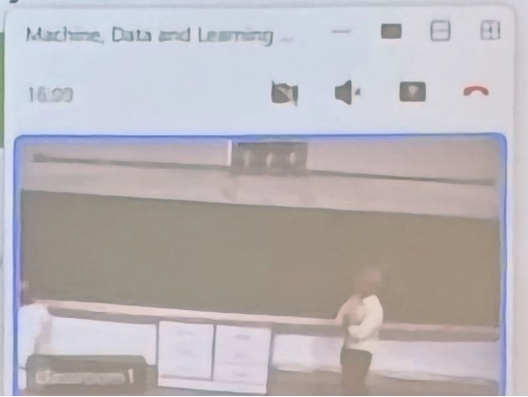
- Classification, Decision Tree Learning, Construct decision trees from examples, Notion of information gain
- Data Mining, Clustering

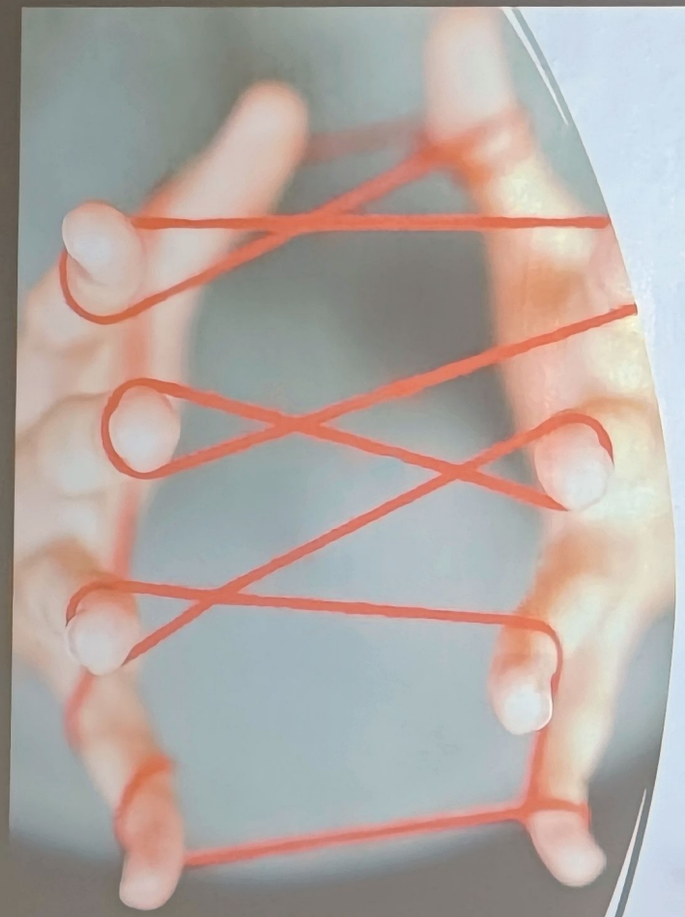
M4 Mathematical Foundations

- Basics of Probability and Utility Theory , Decision Theory

M5: Additional Topics

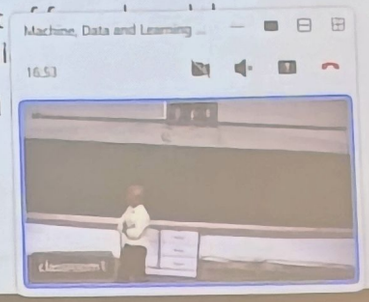
- Reinforcement Learning, Multi-Armed Bandits
- Markov Decision Process, Modeling observation errors, networks





Course Outcomes

- CO-1. Understand basic ML concepts such as Underfitting, Overfitting and Bias-Variance tradeoff
- CO-2. Gain hands-on experience of applying these concepts to example problems
- CO-3. Understand local search techniques with focus on Genetic algorithms
- CO-4. Understand the basics of Probability and Utility theory
- CO-5. Usage of these concepts in the context [illegible] such as Decision theoretic models and Bayesi
- CO-6. Understand Decision tree learning and Information Gain



Class Etiquettes

- Use of mobiles in the class STRICTLY PROHIBITED
- 10:05 is class time. You MUST be settled before that
- No photos of slides/board
  - If you want to take notes, must bring pen/pencil and notebook
  - Use of digital pads/laptops for taking notes not allowed

Artificial Intelligence (AI)

- Artificial Intelligence (AI) is the science and engineering of making machines (including computers, robots and others) intelligent and enabling them to take actions and decisions that humans typically do

- *"Artificial intelligence is one of the most profound undertakings in science, and one that will affect every aspect of human life."*
  - Stanford President John Hennessy

Machine

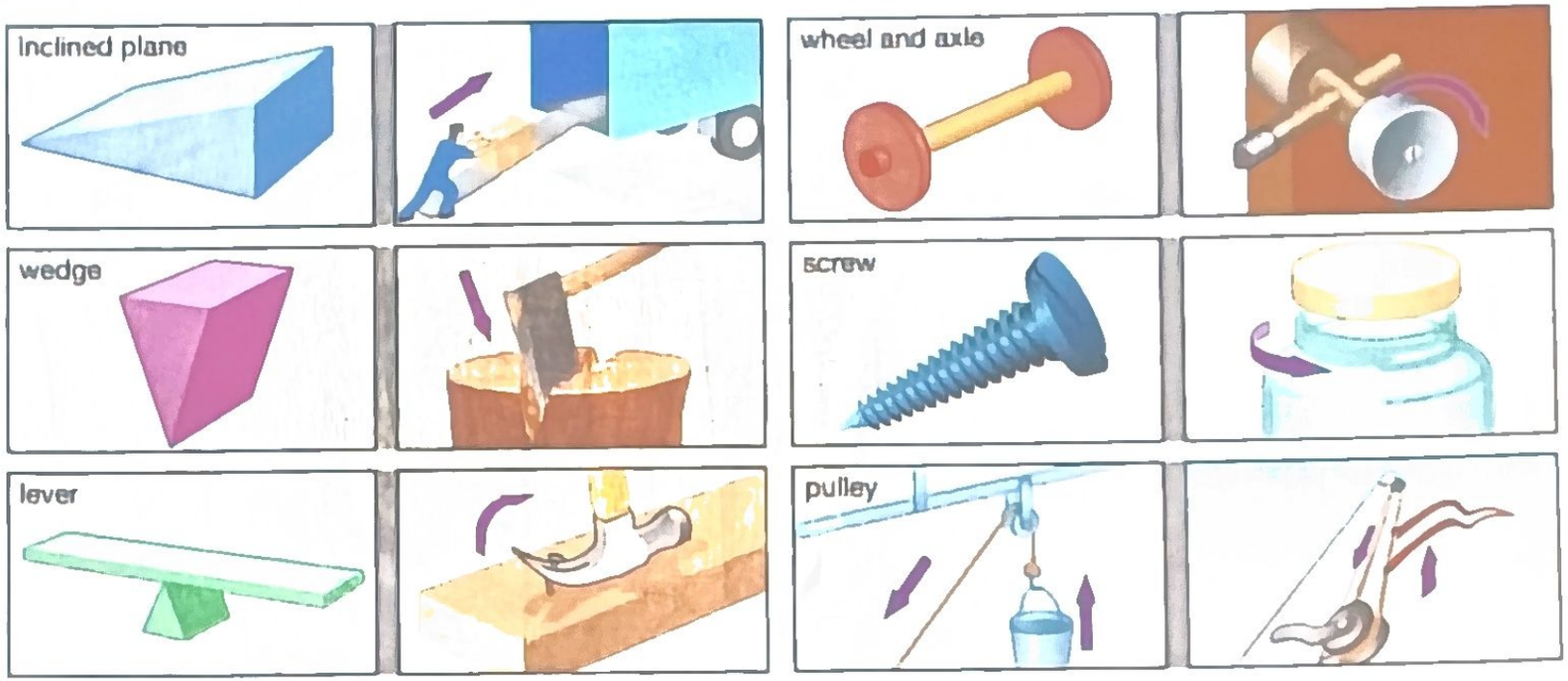


Image credits: Britannica

- (wiki) A machine is a physical system that uses power to apply forces and control movement to perform an action

- Make our lives easier

Future of AI

- Pieter Abbeel (UCB) : robots will keep us safer, especially from disasters
- Stuart Russell (UCB): very smart computers could solve all our problems, including climate change
- Matthew Taylor (WSU) : friendly robots could give the elderly live better lives
- Murray Shanahan (Imperial College): AI will improve medical care

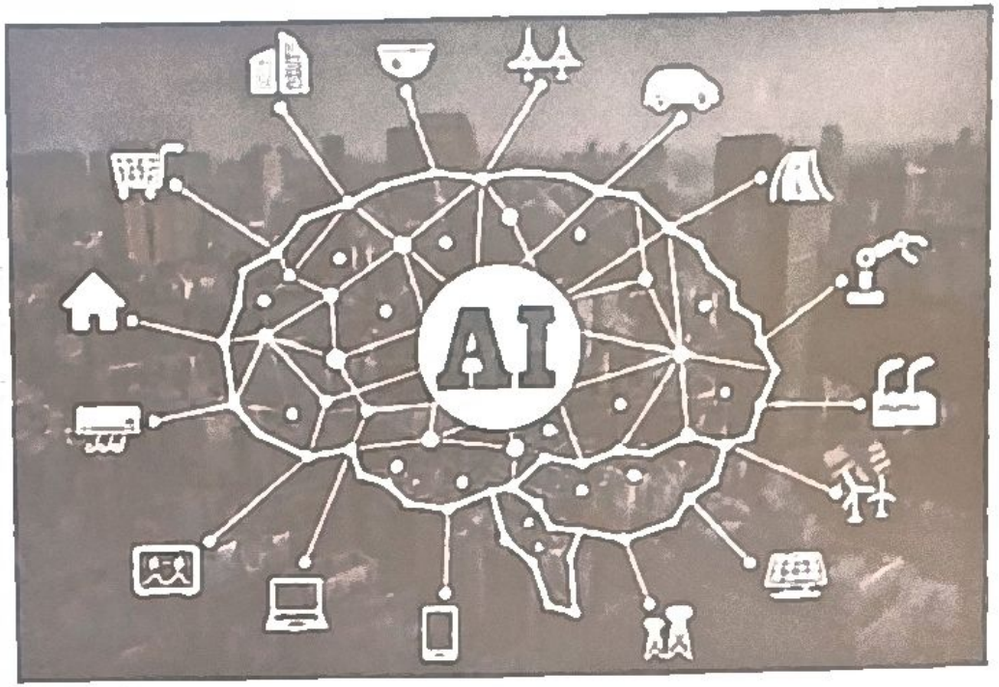
- University of Oxford:
  - Machines will be better than us at translating languages by 2024 and writing school essays by 2026
  - Within ten years computers will be better at driving a truck than us and by 2031 they will be better at selling goods
  - AI will write a bestselling book by 2049 and conduct surgery by 2053

Is AI a Threat?

- Industrialization: Machines replaced lot of human jobs
- Computers replaced lot of book-keeping/Accountants
- Ray Zinn on Forbes in May 2017
  - economics, like life, finds a way
  - Jobs will change, the economy will expand
  - AI will help create jobs
  - technical applications require a range of workers, from field techs who repair sensors to data scientists who model from massive data sets

Proliferation of AI

- Autonomous Driving, Robotics, Llama
- Computer Vision
- In general AI assisted decision making
  - Crowdsourcing, Crowdfunding
  - Online Advertisements
  - Classification, Jurisdiction, Recruitments
  - Resource Allocation, Course Allocation, Ride Shares, Rent Shares



Ubiquitous AI systems

Image Credits: Manisha Padala Thesis

Healthcare

Diagnostic, personalized medicines

Finance

Fraud detection, algorithmic trading

Daily life

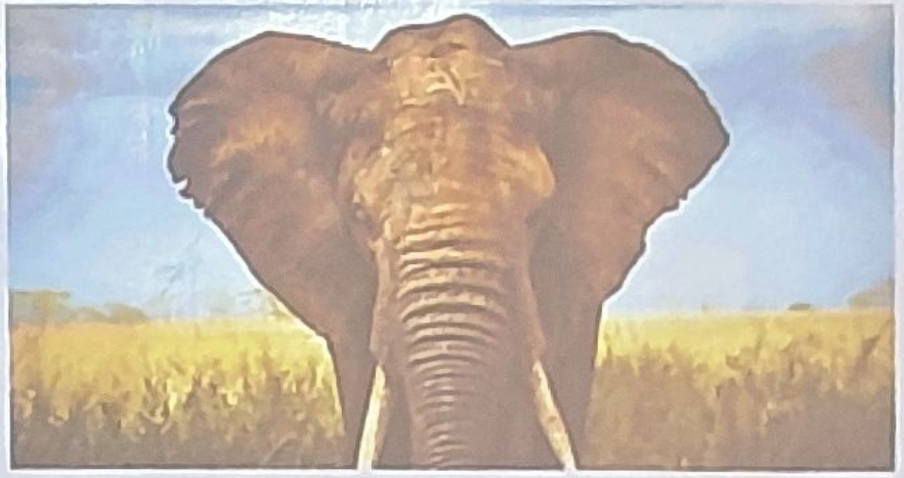
Online Shopping, Digital Assistants, Recommender systems (personalized recommendations) and the list can go on...

AI for Social Good (1)

- The global population: 9.7-10B by 2050
  - Increased agricultural production in order to meet food demands
- Computer Vision
- Data analytics for farmers
  - (weather patterns, soil type, water resource availability, nature of consumption patterns)
- Efficient supply chain management

AI for Social Good (2)

- AI for Public and Wild life Safety
- Limited Resources
- Use Game Theory, ML and AI
  - Automated Event Discovery
  - In Place in USA[1]:
    - PROTECT
    - ARMOUR
    - PAWS



Images: Teamcore, Harvard

What is 'Intelligence'

- the ability to understand, learn and think

- (wiki) the capacity for abstraction, logic, understanding, self-awareness, learning, emotional knowledge, reasoning, planning, creativity, critical thinking, and problem-solving

Intelligence in Humans

Image credits: https://doodlesof.com

Intelligence in Humans

Image credits: https://doodlenot.com

- We think, do learn from observations, act/make deicsions
- Can machines think?
- Can we make machine learn from observations?
- Can we make machines take decisions?

Two dimensions to AI

- Human vs Rational
  - Rationality: doing always right thing
- Act vs Think

Acting Humanly Turing Test

- An human interrogator is not able to distinguish between computer and human after posing some written questions

- Computer need following capabilities
  - Natural language processing
  - Knowledge representation
  - Automated reasoning
  - Machine Learning

Total Turing Test

Involves interaction with objects, people,
Moving objects around etc.,

Additionally need Computer Vision and Robotics

- Think Humanly
  - Cognitive modelling
- Act Rationally
  - Agent based modelling
  - Agent should operate autonomously, perceive the environment, adapt to change
- Think Rationally
  - Right thinking
  - Logic-based approach
  - Need to involve probability to model uncertain things

What is Machine Learning?

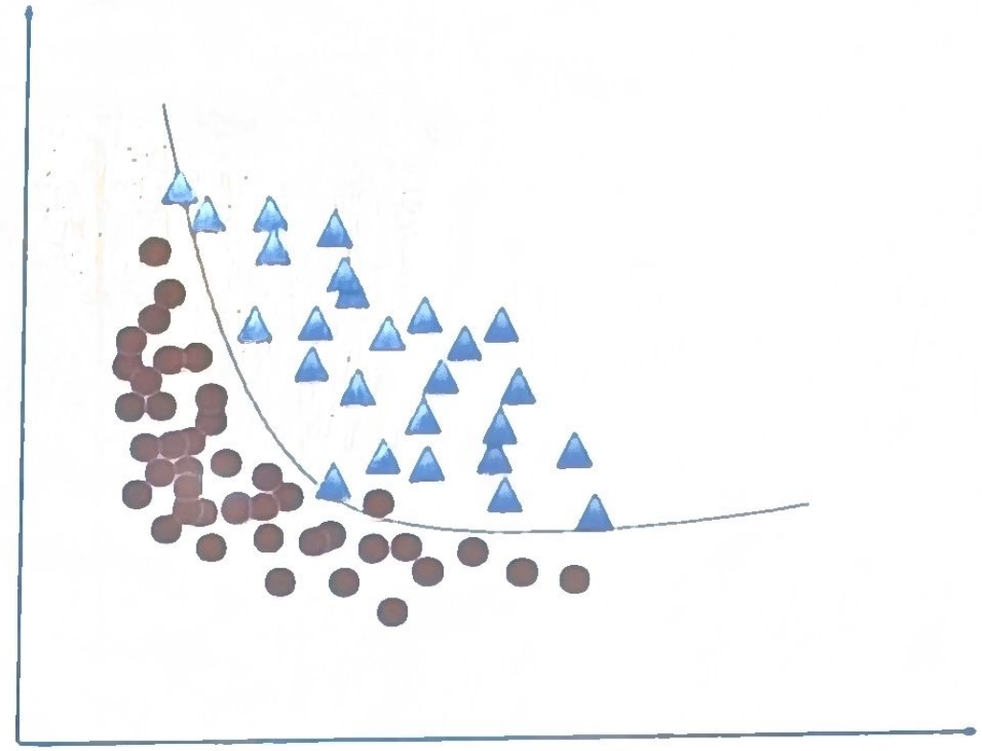
- Machine learning: construction of algorithms that can learn from and make predictions on data

What is Machine Learning?

- Machine learning: construction of algorithms that can learn from and make predictions on data

- data == features

- Supervised: Classification, regression
- Unsupervised: Clustering, anomaly detection



What is Machine Learning?

- Machine learning: construction of algorithms that can learn from and make predictions on data

- data == features

- Supervised: Classification, regression
- Unsupervised: Clustering, anomaly detection

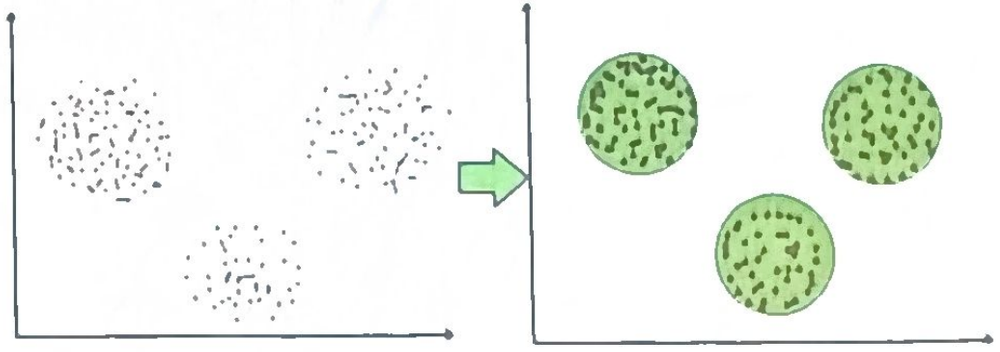
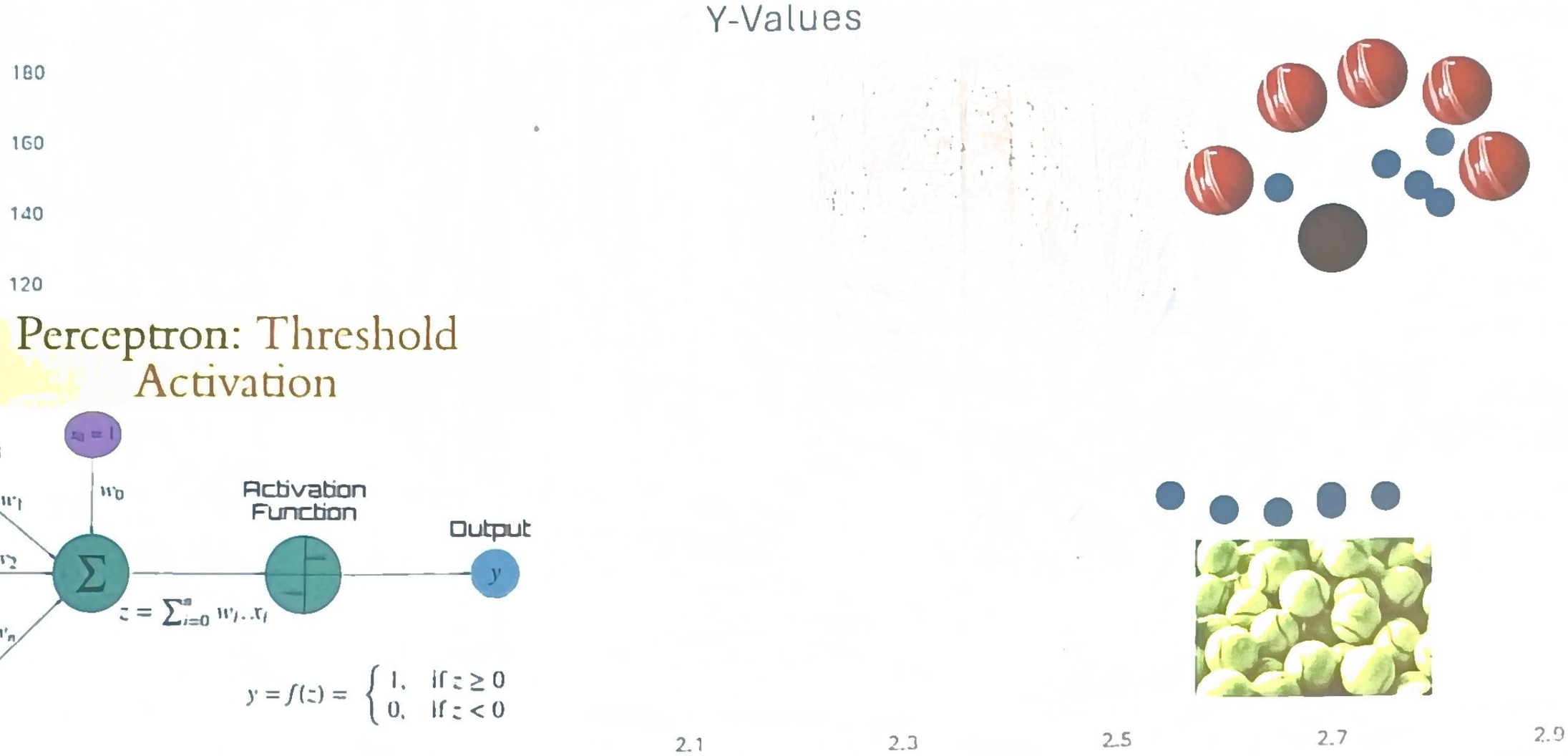


Image credits: https://www.geeksforgeeks.org/

Cricket ball vs tennis ball

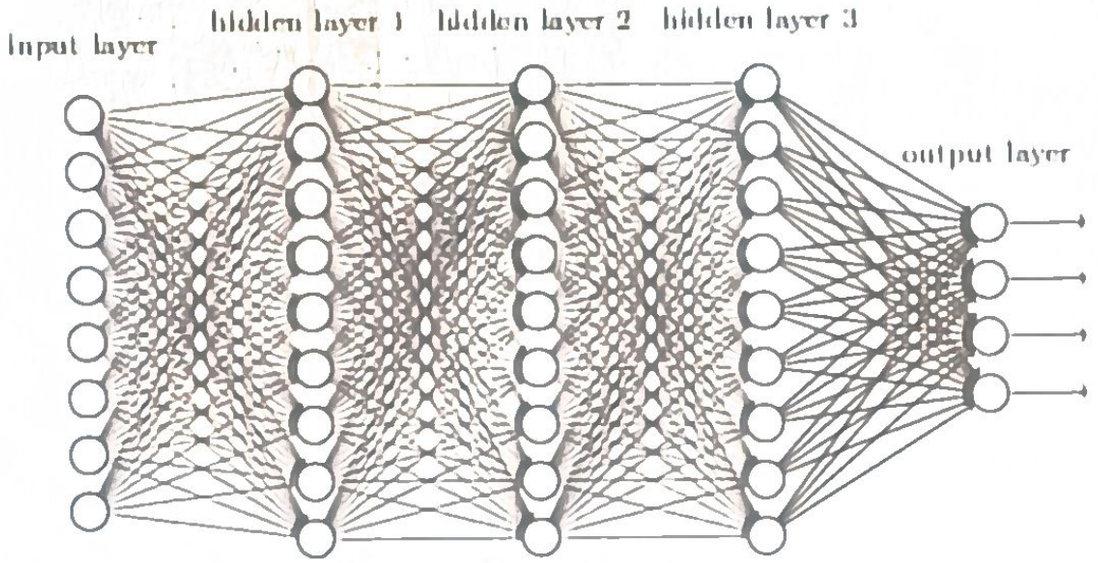


Perceptron: Threshold Activation

Image:

Neural Networks and Deep Learning

+Replicate neurons in human brains (Neural Network)



Complexity of neural networks

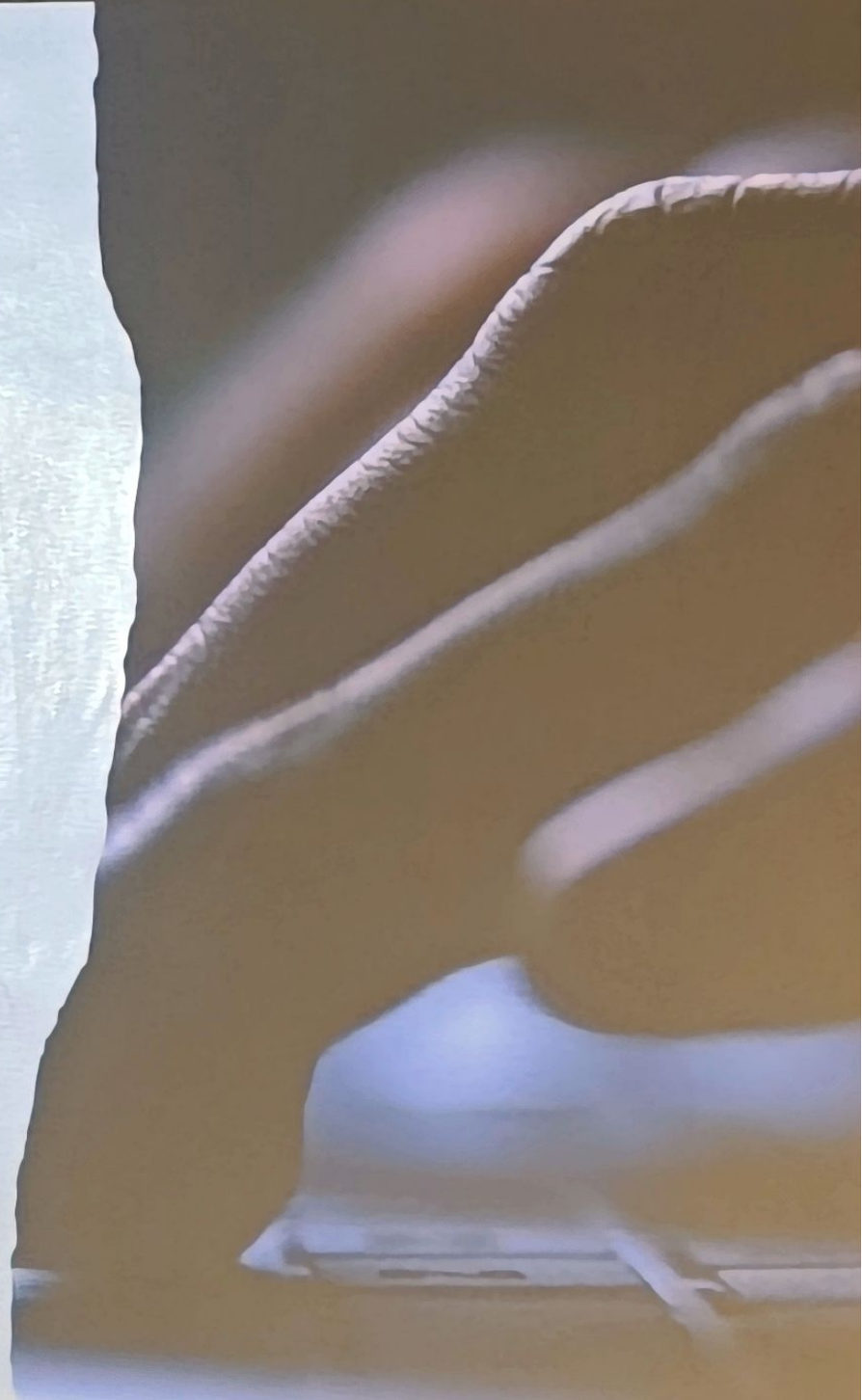
Image credits: Turing-NLG: A 17-billion-parameter language model by Microsoft - Microsoft Research

In Summary, Where Does AI Stand Today?

- AI has made a significant progress in the last three decades in the areas of computer vision, natural language processing, speech recognition

- AI is touching the lives of billions of people through new era applications such as Cortana, Siri, Copilot, Gemini, Perplexity which run even on smart phones

- ...we are far from making machines that match human capabilities



- Fainress/Ethical Issues

- Who takes responsibility of AI fails
  - The person who wrote the code? Or the organization who sold the AI software? The person who bough the software?

  - Uber Car Accident in 2018


You're sharing your screen
Stop sharing
Image credits: copilot

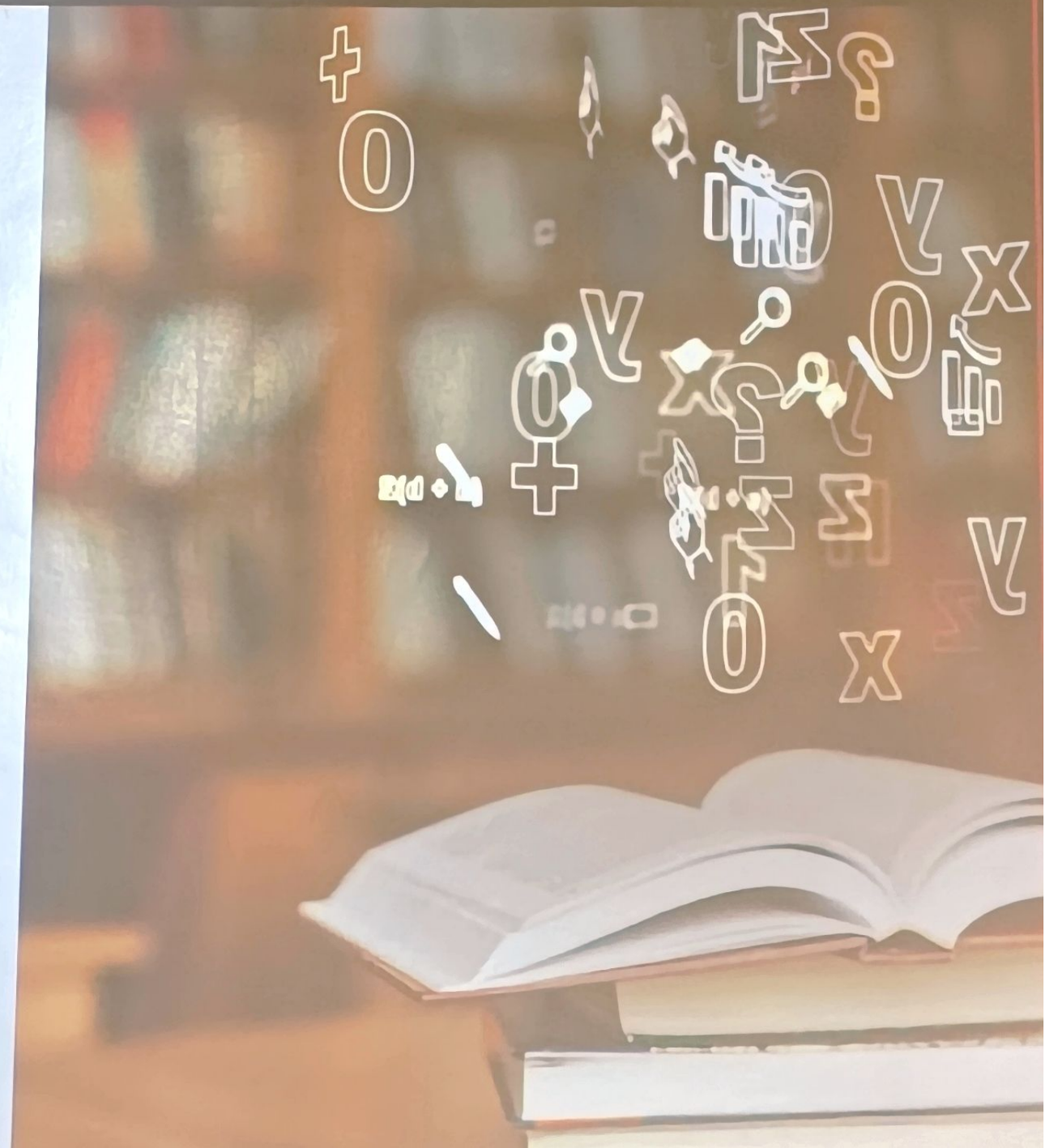
Intelligence

- Intelligence
  - the ability to understand, learn and think
- Intelligence of humans is achieved—not by purely reflex mechanisms but by processes of reasoning that operate on internal representations of knowledge
- Artificial Intelligence (AI) is about building machines and systems that can reason, learn, and solve problems, similar to how humans do. In many AI applications, logical reasoning plays a crucial role, allowing systems to make decisions based on given conditions and facts
- Leads to knowledge-based agents approach for AI

Knowledge Base

- Knowledge Base – Set of sentences represented in knowledge representation language

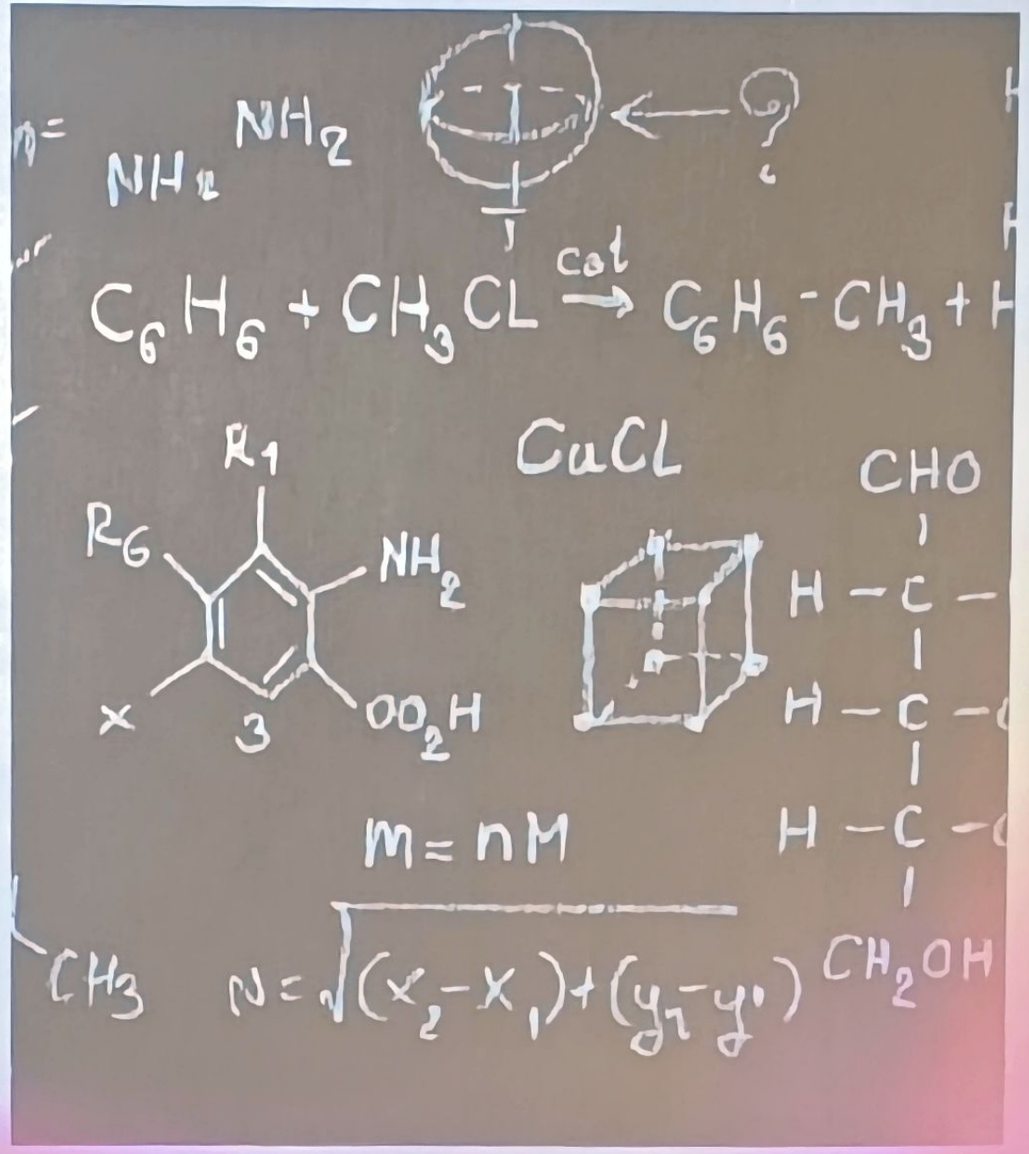
- Knowledge representation language– Expressing knowledge explicitly in a computer-tractable way
  - (logic)



Logic

- Oxford dictionary
  - a way of thinking or explaining something
  - sensible reasons for doing something
  - the science of thinking about or explaining the reason for something using formal methods

- Logic is the basis of all mathematical reasoning and all automated reasoning



BOOKS/RESOURCES

- Software Engineering – A Practitioner's Approach, 9th Edition, Roger Pressman
- Student Resources: http://highered.mcgraw-hill.com/sites/0073375977/student_view0/
  - has Chapter Overviews and Practice Quizzes from 7th edition of the book
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, 3rd Edition, Craig Larma
- Object-Oriented Software Engineering: Practical Software Development using UML and Java, Second Edition
- Head-First Object Oriented Analysis and Design, 1st Edition, Brett McLaughlin, Gary Pollice, and David West

GRADING

Distribution

- 12 % : Quiz 1 & Quiz 2
- 12 % : Mid-Sem Exam
- 10 % : Final - Comprehensive
- 16 % : Assignments (Assignment 1 – 6%; Assignment 2 – 6%; Assignment 3 – 4 %)
- 34 %: Team Project (see Project section)
- 6 % : Class Activities

Cut-off (Tentative):

>= 90.0 :	A
>=85.0 and < 90.0 :	A-
>=80.0 and < 85.0 :	B
>=75.0 and < 80.0 :	B-
>=70.0 and < 75.0 :	C

>=50.0 and < 55.0 : B
>=75.0 and < 80.0 : D-
>=70.0 and < 75.0 : C
>=65.0 and < 70.0 : C-
>=60.0 and < 65.0 : D
< 60.0 : F

The cut-off may be revised upward or downward if the instructor deems it necessary.

QUIZ/EXAMS

Quizzes/Mid-sem exam will be based on topics covered up to that point. A comprehensive final exam shall be conducted during the finals week. Note that exam topics may cover class activities, assignments, project and text book reading - whether discussed in class or not.

ASSIGNMENTS

Assignments shall be given to supplement the learning. Tutorials shall be delivered (as needed) to help with the assignments.

CLASS ACTIVITIES/PARTICIPATION

Impromptu in-class activities will be given throughout the semester. Some of these activities might be individual and some others might be team based. Not all activities will be graded. Specific activities that might be graded shall be randomly decided. So, ensure you submit all activities.

Overall about 0 activities will be graded. Please note there will be no make up activities. So, if you miss an activity you will get a 'U' (unless there is valid medical exception from academic office). It can't be compensated with another that you may have attended.

COURSE PROJECT

Team work

The focal point of the course is a 3 month long project executed by a team of 5 students. The goal of the project is to introduce and practice the fundamental software development life-cycle activities of planning, tracking, designing, implementing and delivering an actual software product. Teams will be formed in the first week and will remain together throughout the semester. The instructor and TAs will choose the teams.

For most of the students this might be the first time you are developing a team project. Working in teams will be challenging in various ways due to the short amount of time teams are able to meet in person. Teams have to decide on their meeting schedules and stick to it till the end of the semester. At times, the instructor may allocate time during the class session for project teams to meet/work and clarify questions (if any).

Participating in the team project is perhaps the most valuable experience you will take away from this course. Far more projects fail as a result of poor collaboration and communication within project teams than due to technical shortcomings. Becoming an effective team member is a critical career skill and the one of the goals of this course is to gain some experience with effective (and non-effective) teaming practices. The expectation is that all students will make an honest effort to work with their team members to the best of their ability. Should non-constructive conflict arise, your instructor will expect the team to first make an effort to resolve issues internally. Beyond that your instructor will intervene as needed to help in resolving team issues. In extreme cases, if a particular student does not cooperate with, he/she may be "FIRED" from the team. In such cases the student will NOT have the opportunity to contribute to the project any more and will end up getting failing grade for the project.

COURSE PROJECT

Team work

The focal point of the course is a 3 month long project executed by a team of 5 students. The goal of the project is to introduce and practice the fundamental software development life-cycle activities of planning, tracking, designing, implementing and delivering an actual software product. Teams will be formed in the first week and will remain together throughout the semester. The Instructor and TAs will choose the teams.

For most of the students this might be the first time you are developing a team project. Working in teams will be challenging in various ways due to the short amount of time teams are able to meet in person. Teams have to decide on their meeting schedules and stick to it till the end of the semester. At times, the instructor may allocate time during the class session for project teams to meet/work and clarify questions (if any).

Participating in the team project is perhaps the most valuable experience you will take away from this course. Far more projects fail as a result of poor collaboration and communication within project teams than due to technical shortcomings. Becoming an effective team member is a critical career skill and the one of the goals of this course is to gain some experience with effective (and non-effective) teaming practices. The expectation is that all students will make an honest effort to work with their team members to the best of their ability. Should non-constructive conflict arise, your instructor will expect the team to first make an effort to resolve issues internally. Beyond that your instructor will intercede as needed to help in resolving team issues. In extreme cases, if a particular student does not cooperate with, he/she may be "FIRED" from the team. In such cases the student will NOT have the opportunity to contribute to the project any more and will end up getting failing grade for the project.

Team work does not imply 80/20 rule. Everybody is expected to contribute equally. The instructor will seek explicit **individual evaluations** from team members at the end of every release (or as needed). The student project grade may be **increased/decreased by up to 50%** depending on the individual evaluations and TA/faculty/client evaluations. So make sure you actively contribute towards the success of this project.

Project Artifacts

Each team should have version control repository set up by week 2 or week 3. All project artifacts including Project Plan, Requirements Specification, Test Plan, Design document, etc. should be submitted through the repository. More details on the set up of the version control repository and other artifact submissions will be announced soon.

Project Releases

This project will be implemented using an iterative incremental process. Each team is expected to produce minimally two project releases – R1 during Week 11 or 12 (tentative) and R2 during Week 15 or 16 (tentative). The initial project plan should clearly specify the functionality being implemented in the two releases. The release plan should be approved by the mentor (project sponsor) and TA. A release includes the current state of all project artifacts created till that point including executable source code. Each release may be accompanied by an in-class team presentation and product demonstration.

More details shall be announced in class as and when necessary.

Project Grading

Your TA and Instructor will grade your project using the following point distribution. The course schedule states the expected project deliverables. Project grades are assigned on