Lecture 1:
History of Computer & Computer Science Research

Faculty of Computing & Informatics
Multimedia University
Cyberjaya, Malaysia

Objectives of this subject

Towards the end of this trimester, you are able to

1. name the broad field of research methods in computing
2. identify the variety of issues, concepts, methods, and techniques associated with computer science research
3. analyse and summarize research papers;
4. present your ideas through research proposal.

Assessments

Assessment Info

a. Quiz – 10%
b. Mid-Term Test – 20%
c. Assignment 1 – 30%
d. Assignment 2 – 40%

Part 1

A Brief Introduction to Computer

Computer Science

- In practice, computer science includes a variety of topics relating to **computers**, which range from the abstract analysis of **algorithms**, **formal grammars**, etc. to more concrete subjects like **programming languages**, **software**, and **computer hardware**.
  - *Everything in computing = algorithm*
- As a scientific discipline, it differs **significantly** from and is often confused with **mathematics**, **programming**, **software engineering**, and **computer engineering**, although there is some degree of overlap with these and other fields.

Early Mathematics & Computation

- Babylonians and Egyptians, $>$ 3000 years ago
  - numerical methods for generating tables of square roots, multiplication, trigonometry
  - Applications: navigation, agriculture, taxation
- Greeks, $>$ 3000 years ago
  – geometry & logic
- Chinese, $>$ 5000 years
  – agriculture & war

Early Mathematics & Computation

- Indians, $\sim$ 600 AD
  – started using placeholders and a decimal number system, similar to modern ones.
  – idea spread to middle east
- Arabs & Persians, $\sim$ 800 AD
  – algorithms

Video - Intro to Computing

Watch the following videos:

https://www.youtube.com/watch?v=k9wSW51sbuE&t=245s
https://www.youtube.com/watch?v=4VROUIAF2Do

A Famous Arab Mathematician

Abu Jafar Mohammed Ibn Mūsā Al-Khwārizmi

- In early 800s AD
- Worked at center of learning in Baghdad
- Wrote book: Hisab Al Jabr Wal-Mugabalah
  1. Described how to compute several practical problems, including linear and quadratic equations
  2. Translated into Latin, spread throughout Europe
- Solidified number system in use now: "*Arabic numerals*"
- Al Jabr gives us the word "*algebra*"
- Al-Khwārizmi gives us the word "**algorithm**"

Abacus:

- ~3000 BC by the Chinese
- developed and adapted over time by different nations/countries

John Napier's bones:

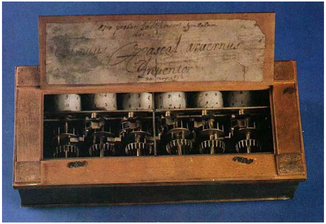
- 1617: Sticks with numbers on them
- use to do 4 basic arithmetic operations

William Oughtred's Slide Rule:

- 1622: Sticks with logarithmic scale, slide along
- much more complex calculations
- used well into 20^{th} century

Early Computing Devices

- Blaise Pascal
  – 1642: First numerical calculating machine (addition and subtraction)
- Gottfried Leibniz
  – 1673: 4-function mechanical calculator (addition, subtractions, multiplication, division)
- Used cogs and gears
- Showed mechanization can simplify and speed up numerical calculations



Are These Devices Computers?

- CANNOT be considered as general-purpose computers
- They are lack of
  - memory
  - ability to be programmed

First Programmable Device with Memory

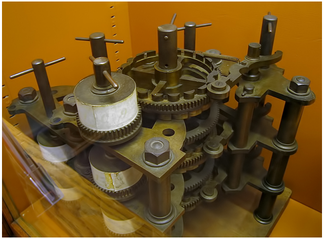
- Used to weave cloth with patterns
- Invented by **Joseph Jacquard**, France, 1804
- Automated loom using punched cards to create pattern
  - hole in card at a certain place causes change in the weave at corresponding place in the fabric

Jacquard Loom

- memory: the cards
- programmable: change the cards
  – first fully automated and programmable loom
  – used punch cards to *program* the pattern to be woven into cloth
- capture human expertise in a machine

England, 1822-1830: Designed and worked on a "Difference Engine" for calculations

- Compute tables of logarithms
- Never finished it: current manufacturing technology not able to provide required precision in cogs and gears
- Others later built one: 7 feet by 11 feet, 3 tons, 4000 moving parts



Part of Babbage's difference engine, assembled after his death by Babbage's son, Henry Prevost Babbage (1824-1918), using parts found in his laboratory

Charles Babbage & Analytical Engine

- 1833: Designed the "Analytical Engine" (a.k.a Difference Engine No. 2)
  1. English mathematician, engineer, philosopher and inventor.
  2. Could not get funding, since never finished first machine, but fully designed to be steam-powered
  3. This was the first general purpose computer!
- Separate storage from calculation
- Used punched cards

Ada Lovelace

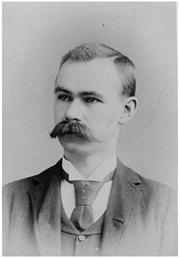
- Augusta Ada Byron, Countess of Lovelace
  – Friend of Charles Babbage
- Translated, edited, and commented on document describing Babbage's Analytical Engine
- Created a program for the (theoretical) Babbage analytical engine which would have calculated Bernoulli numbers.
- Widely recognized as the first programmer.
- Described its potential as a general purpose computer
  1. Wrote "programs" that could be run on it. As a result, she is often considered the world's first computer programmer
  2. Wrote about potential uses, even for computer music

Following Babbage

- General purpose computing waited
  - Instead, several different specific devices
- Most computational devices still mechanical
  - Typewriters (1868)
  - Adding machines (1875) and calculators
  - Cash registers (1879)

Herman Hollerith

- Herman Hollerith developed tabulating machine
    1. Developed machines for encoding information on punched cards
    2. Cards could be sorted and tabulated
- The "1890 Census" was completed in 2 years with Hollerith's machines
  – Also saved millions of dollars



Further Development

- Work continued on machines to add, tabulate, record.
  1. Charles R. Flint: Computing, Tabulating, Recording (CTR) company, followed up on Hollerith's work.
  2. Thomas J. Watson renames CTR to International Business Machines (IBM) in 1924.
- Individual machines were created for each stage of a process
  – For example, separate machines to count, sort.
  – Most machines encoded information on punched cards.

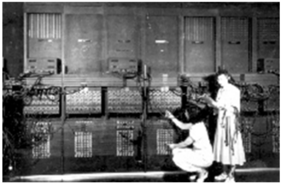
Howard Aiken & MARK I (ASCC)

- Funded by Navy and IBM, at Harvard
- 1930's and 40's
- general-purpose programmable computer
- used relays, magnets and gears
- used binary values (0/1) instead of decimal (0 to 9)
- used vacuum tubes and electric current (on/off) instead of 10-toothed gears
- memory: 72 numbers
- speed: 23-digit multiplication in 4 seconds

ENIAC

Electronic Numerical Integrator and Computer

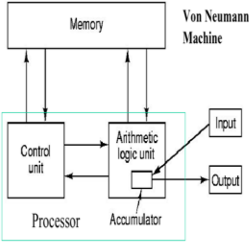
- 1940's
- Motivating application: calculate firing tables (how to aim gun depending on distance, wind speed, temp, etc.)
- Funded by Army at Univ. of Penn.
- John Mauchly & Presper Eckert lead designers
- First fully electronic general-purpose computer
- Vacuum-tube based
- Required rewiring to change program originally
- 100 feet long, 10 feet high, 30 tons
- 1000 times faster than Mark I



Von Neumann Architecture

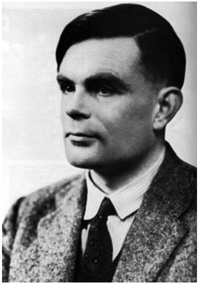
- John Von Neumann $\Rightarrow$ mathematician, physicist, chemist, computer scientist,. . . at Princeton
- worked on ENIAC
- realized shortcoming
- Key idea:

> - encode instructions as binary values and store in memory along with data
> - to change program, rewrite sequence of instructions



Where Did the Idea Come From?

- Andrew Hodges credits Alan Turing's work on the concept of the Universal Turing Machine
- Notion of one machine for all tasks, although it seems obvious to us now, was not at all obvious:



"If it should turn out that the basic logics of a machine designed for the numerical solution of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regard this as the most amazing coincidence that I have ever encountered."

Howard Aiken (1956)

The Modern Era, 1950–Present

- Changes more evolutionary than revolutionary
- Focused on making computers
  1. faster
  2. smaller
  3. cheaper
  4. more reliable
  5. easier to use
- Conventionally divided into rough "generations"

First Generation

1950–1959

- First commercial computers
- First symbolic programming languages
- binary arithmetic
- vacuum tubes for storage
- punched card I/O

Second Generation

1959–1965

- transistors and core memories
  – reduced size and cost, increased reliability
- first disks for mass storage
- first high-level programming languages and programmers
  – FORTRAN, COBOL
- first operating systems

```
C    AREA OF TRIANGLE
     READ INPUT TAPE 5, 501, IA, IB, IC
501 FORMAT (3I5)
C    CHECK THAT SUM OF 2 SIDES IS > THIRD SIDE
     IF (IA) 777, 777, 701
701 IF (IB) 777, 777, 702
702 IF (IC) 777, 777, 703
703 IF (IA+IB-IC) 777, 777, 704
...
```

Third Generation

1965–1975

- Integrated circuits
  – components are photographically etched onto pieces of silicon
  – further reduction in size and cost
- first mini-computers
  – desk-sized instead of room-sized
- time-shared operating systems
- appearance of software industry
- introduction of computing standards for compatibility

Fourth Generation

1975–1985

- Very large scale integrated circuits (VLSI)
  – complete system on one circuit board
  – further reduction in size and cost, increased reliability
- first micro-computer
  – desk-top machine, instead of desk-sized
- further growth of software industry
- computer networks
- graphical user interfaces (GUI)

fifth Generation

1985–?

- Ultra-large scale integrated circuits (ULSI)
  – more than 1,000,000 elements on one chip
- supercomputers and parallel processors
- laptops and hand-held computers
- wireless computing
- on-line terabyte storage devices
- global networks and distributed systems
- artificial intelligence
- hi-resolution graphics, visualization, virtual reality
- multimedia user interfaces

Part 2

Overview of Computer Science Research

Computer Science – What?

Theory

- Algorithms
- Computation

Practice

- Software Engineering
- Application Areas

Systems

- OS
- Architecture

Definition of Computer Science

Definition

The development of *new techniques* using computers & computation to *solve problems* that were fundamentally not solvable before.

- Contrast to IT: there are no fundamental challenges in setting up a billing system that have not yet been solved - it's clear that it's achievable.
- Contrast with programming: though programming is by no means trivial, it is usually clear at the onset that the program can be written.
- Computer programming represents the means by which (empirical) computer science research is conducted.

Computer Science

Computer science is (*for the most part*) an engineering science:

- We have a task we would like to accomplish.
- We don't know if it is solvable, let alone how to solve it.
- We (*attempt to*) find computational methods to accomplish our task.

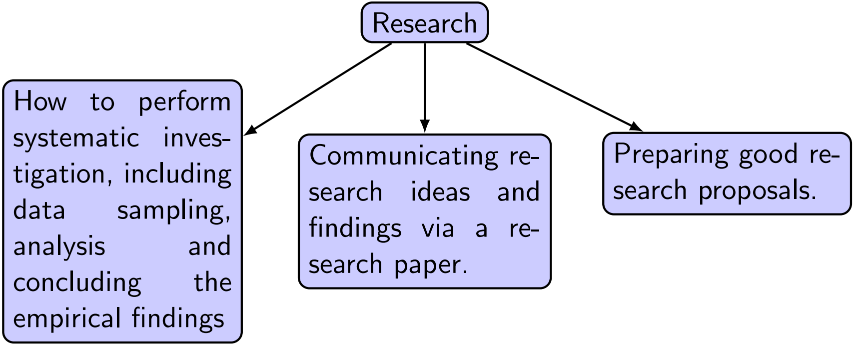
Computer Science: Sub-Areas

Not Limited to...

- machine learning
- software engineering
- data mining
- database
- compiler
- networking
- operating systems
- distributed systems
- computer vision
- numerical computing
- human-computer interaction
- graphics
- computer architecture
- security
- cryptography algorithms
- computational geometry
- image & video processing
- bioinformatics

Research

What is it?



Are you interested to do research?

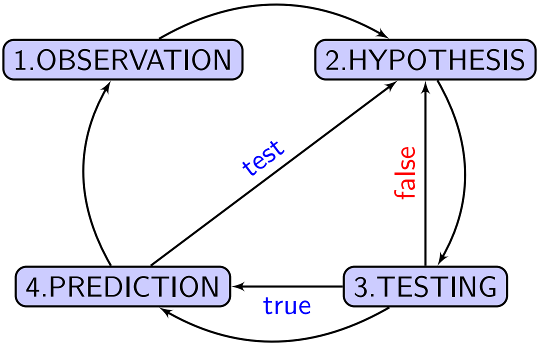
What is Research Method?

- The Research Method can be described as a sequence of steps for systematically analyzing scientific questions; designing and executing research to answer those questions, and producing reproducible results.
- The method is expressed in the scientific research that constantly adds new knowledge and discoveries to the various realms and endeavors of the sciences.

What is Research Method?

Sample steps are;

1. Researching previous studies and current data to pose a question or hypothesis for current research.
2. Designing a research "protocol" to answer the question, or prove/disprove the hypothesis.
3. Executing the research to create reproducible results.


1.OBSERVATION
2.HYPOTHESIS
3.TESTING
4.PREDICTION
test
false
true

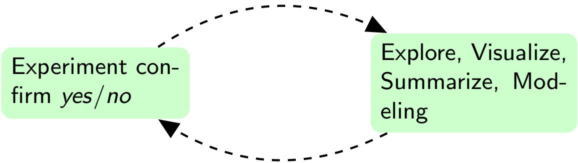
- Observation: an act of recognizing and noting a fact or occurrence often involving experiment(s)
- Hypothesis: tentative assumption made in order to draw out and test its logical or empirical consequences
- Testing: to apply a test on algorithm(s) or model(s) as a means of analysis
- Prediction: to predict the outcomes of algorithm(s) or model(s)

Theoretical Research

use mathematics/logical reasoning to prove what you are proposing is definitely true.

Empirical Research

run many experiments to show that what you are proposing is probably true.



Example

- Proof theoretically that a particular casino gambling strategy will give you the highest possible winnings.
- Demonstration over many experiments that a particular strategy works better than others

Important: It's Not always easy/possible to prove something *mathematically*.

Hypothesis

- The first stages of a research involve identifying a specific issue to investigate $\rightarrow$ develop a specific question to answer.
- In computer science, a hypothesis is usually about whether a proposed approach fits for a certain purpose.

Example:

1. Using algorithm X we can reduce the number of memory accesses and make the program faster
2. Sorting algorithm Y can be improved if we replace a tree-based structure with an array-based structure

- The goal of the research will be to test this hypothesis
- The hypothesis should be clear, precise, and unambiguous.
- Often is important to state what the limitations of the hypothesis (research work).

Consider that there are two data Artificial Intelligence methods: *Deep Learning* and *XGBoost*. You believe that *Deep Learning* is superior than *XGBoost*.

Hypothesis 1

Deep Learning is superior to XGBoost

- Wrong Hypothesis (un-testable) → Success has to apply in all applications, in all conditions, for all times

Hypothesis 2

As an in-memory search structure for large data-sets, XGBoost is faster and more compact than Deep Learning

- Good Hypothesis (testable) → the scope is limited to a domain that can feasibly be explored

- An algorithm is a tool for solving some well-specified computational problem
- An algorithm is a well-defined computational procedure that takes some value(s) as input and produces some value(s) as output.
- An algorithm can be expressed in natural language (e.g., English), as a computer program (e.g., Python, R, Java, C# and many more), or even in a hardware design.

- The notion of algorithm is fundamental to all of computer science.
- An algorithm is a step-by-step procedure ("recipe") for how to complete a particular task or solve a problem.
- Computers can perform tasks very quickly, but they require a very precise description of what to do.

Algorithm

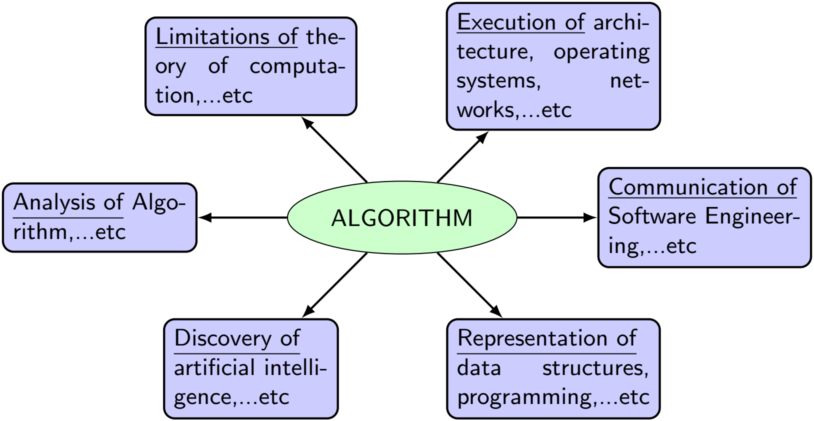
Sample in English & C

Algorithm 1 Counting 1 → 100

```
1: procedure MyCounting
2: Start with number 1
3: Print the number
4: Add 1 to the number
5: Go back to Step 2 (line 3) until reaching 100
6: end procedure
```

Algorithm 2 Counting 1 → 100

```
1: procedure MyCounting
2: int number = 1;
3: while (number <= 100){
4: printf("%d\n", number);
5: number += 1;
6: end procedure
```



ALGORITHM
Limitations of theory of computation,...etc
Execution of architecture, operating systems, networks,...etc
Analysis of Algorithm,...etc
Communication of Software Engineering,...etc
Discovery of artificial intelligence,...etc
Representation of data structures, programming,...etc

Function

Definition

A correspondence between a collection of possible input values and a collection of possible output values so that each possible input is assigned a single output

- Computing a function = determining the output value associated with a given set of input values
- Noncomputable function = a function that cannot be computed by any algorithm

Definition

Computation is any type of calculation or use of computing technology in information processing

- Some computer science tasks, even if there exists algorithms to solve them, may take prohibitively long to finish.
- Other algorithms may finish their work in less than a second.
- Which tasks can be solved quickly, and which take more time?

Definition: complexity of that algorithm

The amount of time that an algorithm takes to finish a task.

Literature Review

- To review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic.
- Writing a literature review lets you gain and demonstrate skills in two areas:
  1. information seeking: the ability to scan the literature efficiently, using manual or computerized methods, to identify a set of useful articles and books
  2. critical appraisal: the ability to apply principles of analysis to identify unbiased and valid studies.

Technical Writing

- Writing should be part of the research process
  1. It's really hard to "Do The Work" and then "Write It Up"
  2. For one thing, The Work is never done, and It is constantly changing
  3. Writing helps to pin down the details, and helps to focus your ongoing research
- Important for it to be accurate and clear
- Tools = Microsoft Word, LibreOffice, LaTeX, and etc ...

In this course, you shall learn preparing documents using LaTeX.

Refereeing

- criticism and analysis of papers written by some other authors/scientists.
- main mechanism for identifying good research and eliminating bad.

Ethics

- Science is built on trust.
- Ethics deserves our attentions.
- Published research:
  1. should be: new, objective, and fair.
  2. shouldn't: present opinion as fact, distort truths, and plagiarize others.

Ethics

The forms of unethical behaviors :

- Plagiarism
- Abuse of power
- Fraud: fake experiment result

The contents presented in this slide are meant for teaching purpose only. No commercialized component exist. The texts are partially copied from the textbooks and images are downloaded from the internet.