What does the computer does best?

- Perform operations
  - Add, sub, multiply, copy from data from location to another

- Store information
  - Depending on memory capacity

- Interact with outside world
  - Input/Output

- Fast
  - E.g. 2 GHz processor can roughly carry out 2 billion instruction in 1 second

Computing in general

- Automate a known solution
- How we can best do it
- Investigate things we don’t know

- Its theoretical foundation lies in
  - Mathematics (discrete math)
  - Electrical Engineering
Digital System

- Computer is a digital system

- Digital System
  - Uses discrete (discontinuous) values
  - By contrast, non-digital (or analog) systems use a continuous range of values to represent information
    - E.g. Temperature at a given location with respect to time

- A discrete value is a sampled version of a continuous value
  - Data is noted at fixed intervals
  - E.g. every microsecond rather than continuously

Digital vs. Analog

What does the Computer Understand?

- At the lowest level, a computer has electronic "plumbing"
  - Operates by controlling the flow of electrons through very fast tiny electronic devices called transistors

- The devices react to presence or absence of voltage
  - Could react actual voltages but designing electronics then becomes complex

- Symbolically we represent
  1. Presence of voltage as "1"
  2. Absence of voltage as "0"

What does the Computer process & store?

- An electronic device can represent uniquely only one of two things
  - Each "0" and Each "1" is referred to as a Binary Digit or Bit
  - Fundamental unit of information storage

- To represent more things we need more bits
  - E.g. 2 bits can represent four unique things: 00, 01, 10, 11
  - k bits can distinguish 2^k distinct items

- Combination binary bits together can represent some information or data. E.g. 01000001 can be
  1. Decimal value 65
  2. Alphabet (or character) ‘A’ in ASCII notation
  3. Command to be performed e.g. Performing Add operation

**Machine Language**

- Computers understand only 0’s and 1’s
  - A.k.a. *Machine (hardware) Language*
  - Each machine has its unique language
    - E.g. combination of 00101001 is not the same meaning on different machines
    - Difficult for humans to program the computer in machine language

**Programming to Machine Language**

- Write in language almost like English
- Another special software (compiler) translates the programming language into a *specific machine language*
  - Specific Machine: Electronic Hardware + Operating System
- Once translated (Programming -> Machine)
  - The same program *cannot run* on different machines
- Java avoids the above problem
  - Code is portable - Write one, run anywhere!
  - One of the features for popularity of Java
  - Caveat: The code we write for Boe-Bots is not portable

**Programming Languages**

- Computer Programming
  - Is telling the computer how to do something
  - Wikipedia Definition: Applies specific *programming languages* to solve specific computational problems with solutions

- *Programming Languages*
  - Unlike human languages
    - Designed for *instructing* computers to solve problems
  - Like human languages
    - They have a *grammar*
    - We will be learning grammar for *Java* Programming Language
    - Caveat: Stripped down version of the regular Java

**Programming Language Syntax**

- Syntax is the *grammar* of the language
- Analogous to rules in English Language:
  - Missing a period after sentence
  - Rules using verbs, nouns etc..
- Compiler (special software) will point out every syntax error
- Error messages may be helpful
  - Often, they are not
  - You gain experience with error messages after a while
    - Helps to have a good editor that provides highlights for language feature
Program

- A written set of instructions in a particular language

- The first step in writing any program is to analyze the work/job to be done
  - Think Algorithmically

Algorithm

- Step-by-step procedure that is guaranteed to terminate, such that each step is precisely stated and can be carried out

Algorithm Example

(1) INPUT A NUMBER N
(2) MULTIPLY N TIMES ITSELF
(3) MULTIPLY N TIMES 2
(4) ADD THE RESULTS OF (2) AND (3)
(5) ADD 10 TO THE RESULT OF (4)

Programming Language Semantics

- When the computer carries out your instructions
  - Running or Executing a program

- Semantics is the meaning of the program
  - We learn the semantics after we run or execute the program
  - Basically we observe the output

- After the executing program, the semantics of the program may or may be correct

- Semantic errors cause your answers to be wrong
  - You may or may not get error messages
    - E.g. Error Message – Dividing a number by zero
  - If your program is not doing what you want it to do, though it runs, the error is semantic

Programming - The Craft

- People have different tastes in programming, but many values are held in common
  - Programming is an art as well as a craft

- Craft
  - Learn concepts fundamental to all programming languages
  - Learn about Object Oriented Programming (OOP) model using Java
  - Learn a variety of data structures and algorithms
Programming Recurring Themes

- Modularity
  - Break big problem into sub problems and compose

- Algorithm Thinking
  - A well defined procedure

- Abstraction
  - Look for and encode common features