SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

❖ **A readable, dynamic, pleasant, flexible, fast and powerful language**

OBJECTIVES:

- To understand the basics of algorithmic problem solving.
- To learn to solve problems using Python conditionals and loops.
- To define Python functions and use function calls to solve problems.
- To use Python data structures - lists, tuples, dictionaries to represent complex data.
- To do input/output with files in Python.

OUTCOMES:

Upon completion of the course, students will be able to

- ❑ Develop algorithmic solutions to simple computational problems.
- ❑ Develop and execute simple Python programs.
- ❑ Write simple Python programs using conditionals and loops for solving problems.
- ❑ Decompose a Python program into functions.
- ❑ Represent compound data using Python lists, tuples, dictionaries.
- ❑ Read and write data from/to files in Python Programs.

19IT103 COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

UNIT I COMPUTATIONAL THINKING AND PROBLEM SOLVING 9

Fundamentals of Computing – Identification of Computational Problems - Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS 9

Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT III CONTROL FLOW, FUNCTIONS, STRINGS 9

Conditionals:Boolean values and operators, conditional (if), alternative (if-else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

UNIT IV LISTS, TUPLES, DICTIONARIES 9

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

UNIT V FILES 9

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100).

TOTAL : 45 PERIODS

TEXT BOOKS:

Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016.

Karl Beecher, "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017.

REFERENCES:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.
2. G Venkatesh and Madhavan Mukund, “Computational Thinking: A Primer for Programmers and Data Scientists”, 1st Edition, Notion Press, 2021.
3. John V Guttag, "Introduction to Computation and Programming Using Python: With Applications to Computational Modeling and Understanding Data‘‘, Third Edition, MIT Press , 2021
4. Eric Matthes, “Python Crash Course, A Hands - on Project Based Introduction to Programming”, 2nd Edition, No Starch Press, 2019.
5. https://www.python.org/
6. Martin C. Brown, “Python: The Complete Reference”, 4th Edition, Mc-Graw Hill, 2018.

UNIT I COMPUTATIONAL THINKING AND PROBLEM SOLVING 9

Fundamentals of Computing – Identification of Computational Problems - Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

1.1 Fundamentals of Computing:

A computer is:

- An electronic machine that can be programmed to accept data (***input***), and process it into useful information (***output***). Data is put into secondary storage (***storage***) for safekeeping or later use.
- The *processing* of input into output is directed by the software, but performed by the hardware.

1.1 Fundamentals of Computing:

History of computer:

- 1822: English mathematician **Charles Babbage** conceives of a steam-driven calculating machine that would be able to compute tables of numbers.

- 1890: Herman Hollerith designs **a punch card system** to calculate the 1880 census, accomplishing the task in just three years and saving the government $5 million. He establishes a company that would ultimately become IBM.

- 1936: **Alan Turing** presents the notion of a universal machine, later called the Turing machine, capable of computing anything that is computable. The central concept of the modern computer was based on his ideas.

- 1939: **Hewlett-Packard** is founded by David Packard and Bill Hewlett in a Palo Alto, California, garage, according to the Computer History Museum.

1.1 Fundamentals of Computing:

History of computer:

•1943-1944: Two University of Pennsylvania professors, John Mauchly and J. Presper Eckert, build the Electronic Numerical Integrator and Calculator (ENIAC).

•1946: Mauchly and Presper leave the University of Pennsylvania and receive funding from the Census Bureau to build the UNIVAC, the first commercial computer for business and government applications.

•1953: Grace Hopper develops the first computer language, which eventually becomes known as COBOL.

•1971: Alan Shugart leads a team of IBM engineers who invent the "floppy disk," allowing data to be shared among computers.

1.1 Fundamentals of Computing:

History of computer:

- 1976: Steve Jobs and Steve Wozniak start Apple Computers on April 1st and roll out the Apple I, the first computer with a single-circuit board, according to Stanford University.
- 1981: The first IBM personal computer, code-named "Acorn," is introduced.
- 1990: Tim Berners-Lee, a researcher at CERN, the high-energy physics laboratory in Geneva, develops HyperText Markup Language (HTML), giving rise to the World Wide Web.
- 1993: The Pentium microprocessor advances the use of graphics and music on PCs.

1.1 Fundamentals of Computing:

History of computer:

•1999: The term Wi-Fi becomes part of the computing language and users begin connecting to the Internet without wires.

•2003: The first 64-bit processor, AMD's Athlon 64, becomes available to the consumer market.

•2005: YouTube, a video sharing service, is founded. Google acquires Android, a Linux-based mobile phone operating system.

•2007: The iPhone brings many computer functions to the smartphone.

•2010: Apple unveils the iPad, changing the way consumers view media and jumpstarting the dormant tablet computer segment.

1.1 Fundamentals of Computing:

History of computer:

1.1 Fundamentals of Computing:

Generations of computer:

- The computer has evolved from a large-sized simple calculating machine to a smaller but much more powerful machine.
- The evolution of computer to the current state is defined in terms of the generations of computer.
- Each generation of computer is designed based on a new technological development, resulting in better, cheaper and smaller computers that are more powerful, faster and efficient than their predecessors.
- Currently, there are five generations of computer.

1.1 Fundamentals of Computing:

First Generations computer (1940-56):

- The first Generation computers used **vacuum tubes** for circuitry and **magnetic drums** for memory.
- They were often enormous and taking up entire room.
- First generation computers relied on machine language.
- They were very expensive to operate and in addition to using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions.
- The **UNIVAC** and **ENIAC** computers are examples of first-generation computing devices.

1.1 Fundamentals of Computing:

Second Generations computer (1956-63):

- **Transistors** replaced vacuum tubes in the second generation of computers.
- Second-generation computers moved from cryptic binary machine language to symbolic.
- High-level programming languages were also being developed at this time, such as early versions of COBOL and FORTRAN.
- These were also the first computers that stored their instructions in their memory.
- The **IBM 1620** and **UNIVAC 1108** are examples of Second-generation computing devices.

1.1 Fundamentals of Computing:

Third Generations computer (1964-71):

•The development of the **integrated circuit** was the hallmark of the third generation of computers.

•Transistors were miniaturized and placed on silicon chips, called **semiconductors**.

•Instead of punched cards and printouts, users interacted with third generation computers through keyboards and monitors and interfaced with an operating system.

•Allowed the device to run many different applications at one time.

•**IBM-360 series** and **Honeywell-6000** series computers are example for third generation computers.

1.1 Fundamentals of Computing:

Fourth Generations computer (1971-present):

- Computers of fourth generation used Very Large Scale Integrated (**VLSI**) circuits.
- VLSI circuits having about 5000 transistors and other circuit elements with their associated circuits on a single chip made it possible to have microcomputers of fourth generation.
- Fourth generation computers became more powerful, compact, reliable, and affordable. All the high-level languages like **C, C++, DBASE** etc., were used in this generation.
- **PDP 11,CRAY-1(Super Computer),CRAY-X-MP(Super Computer)** are example for fourth generation computers.

1.1 Fundamentals of Computing:

Fifth Generations computer (Present -Next):

- In the fifth generation, VLSI technology became **ULSI** (Ultra Large Scale Integration) technology, resulting in the production of microprocessor chips having ten million electronic components.

- This generation is based on parallel processing hardware and AI (Artificial Intelligence) software.

- AI try to simulate the human way of thinking and reasoning.

- All the high-level languages like C and C++, Java, .Net etc., are used in this generation.

1.1 Fundamentals of Computing:

Types of computers:

Computers for Individual Use:

- Desktop computers
- Workstations
- Notebook computers
- Tablet computers
- Handheld computers, Palm computers
- Smart phones

1.1 Fundamentals of Computing:

Types of computers:

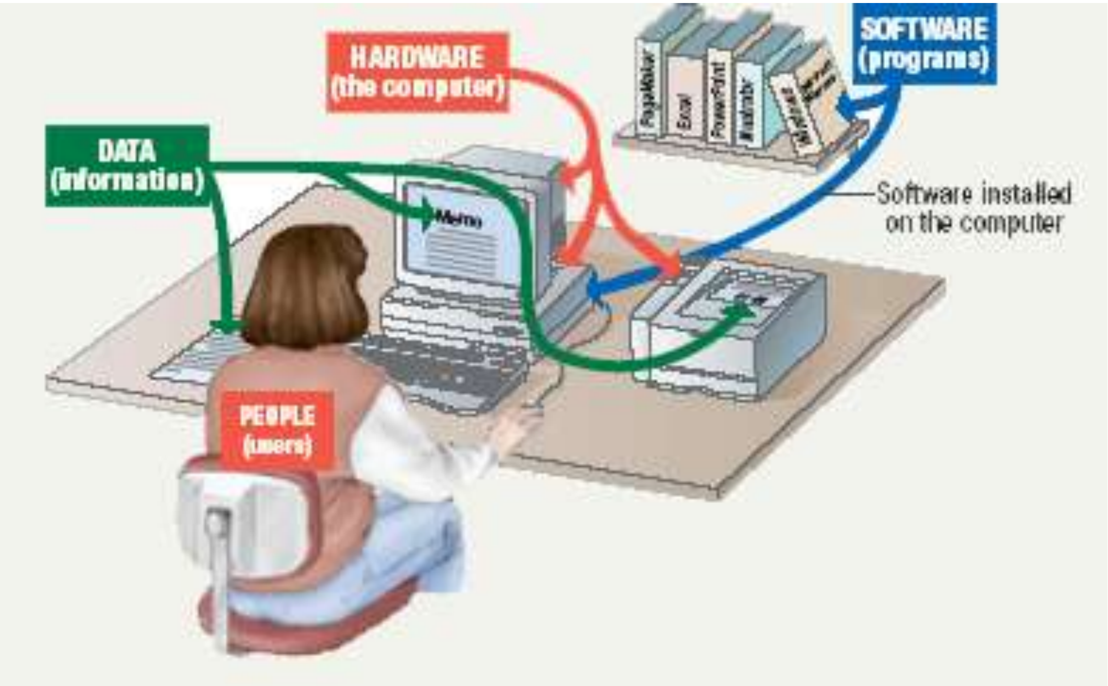
Computers for Organization:

- Network servers
- Mainframes
- Mini computers
- Super computers

1.1 Fundamentals of Computing:

Parts of Computers:

- Hardware
- Software
- Data
- User



1.1 Fundamentals of Computing:

Parts of Computers:

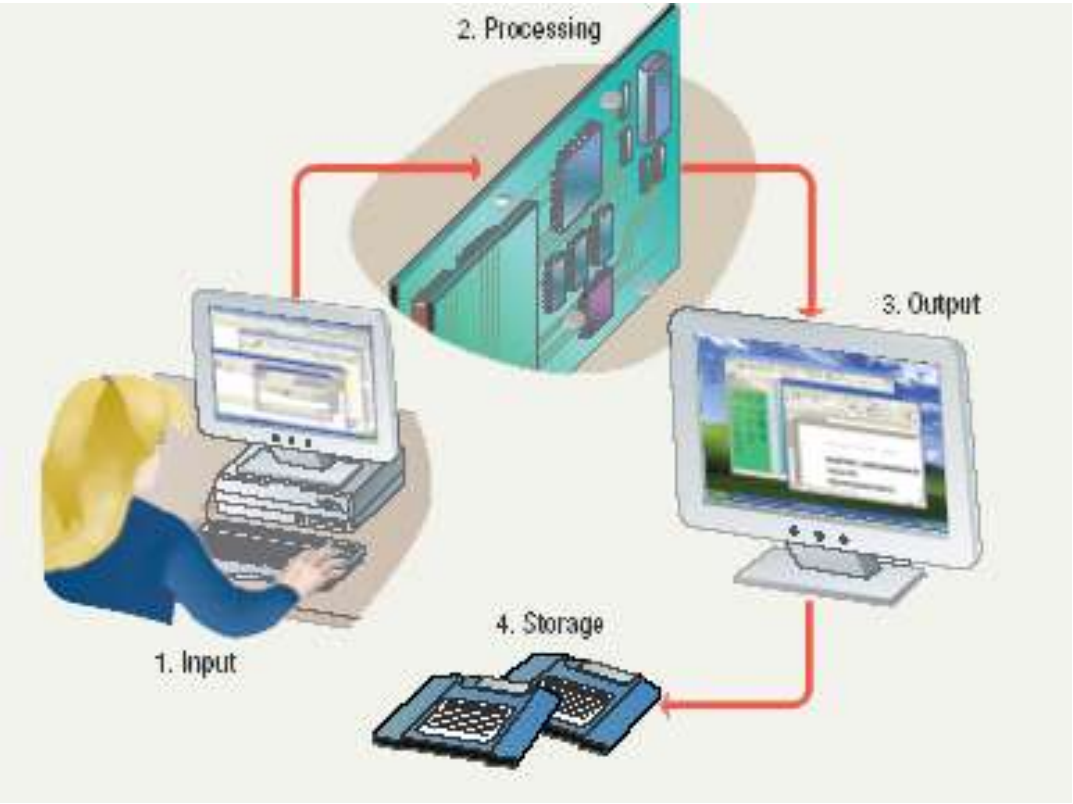
- Hardware - Mechanical devices in the computer
- Software - Tell the computer what to do ,also called a program
- Data - Pieces of information , Computers organize and present data
- User - People operating the computer , Most important part, Tell the computer what to do

1.1 Fundamentals of Computing:

Information Processing Cycle:

Steps to be followed to process data:

- Input
- Processing
- Output
- Storage



1.1 Fundamentals of Computing:

What is computing?

- **Computing** is any goal-oriented activity requiring, benefiting from or creating computers.
- **Computing** includes designing, developing and building hardware and software systems; processing, structuring and managing various kinds of information; doing scientific research on and with computers; making computer systems behave intelligently.

Subfields of computing includes :

  - Computer Engineering, Computer Science
  - Software Engineering
  - Information Systems, Information Technology

1.2 Identification of Computational Problems :

Problem:

- Problem is a thing that requires logical thought and /or mathematics to solve.

Problem Solving:

- Problem solving is the systematic approach to define the problem and creating number of solutions.
- The problem solving process starts with the problem specifications and ends with a Correct program.

1.2 Identification of Computational Problems :

Problem Solving with Computers :

- Computers are built to solve problems with algorithmic solutions, which are often difficult or very time consuming when input is large.
- Solving a complicated calculus problem or alphabetizing 10,000 names is an easy task for the computer.
- So the basis for solving any problem through computers is by developing an algorithm.

1.2 Identification of Computational Problems :

Problem Solving with Computers :

- Field of computers that deals with heuristic types of problems is called Artificial Intelligence (AI)
- Artificial intelligence enables a computer to do things like human by building its own knowledge bank
- As a result, the computer’s problem-solving abilities are similar to those of a human being.
- Artificial intelligence is an expanding computer field, especially with the increased use of Robotics.

1.2 Identification of Computational Problems :

Problem Solving Techniques:

- Problem solving technique is a set of techniques that helps in providing logic for solving a problem.

Problem solving can be expressed in the form of:

1. Algorithms.
2. Flowcharts.
3. Pseudo codes.
4. Programs

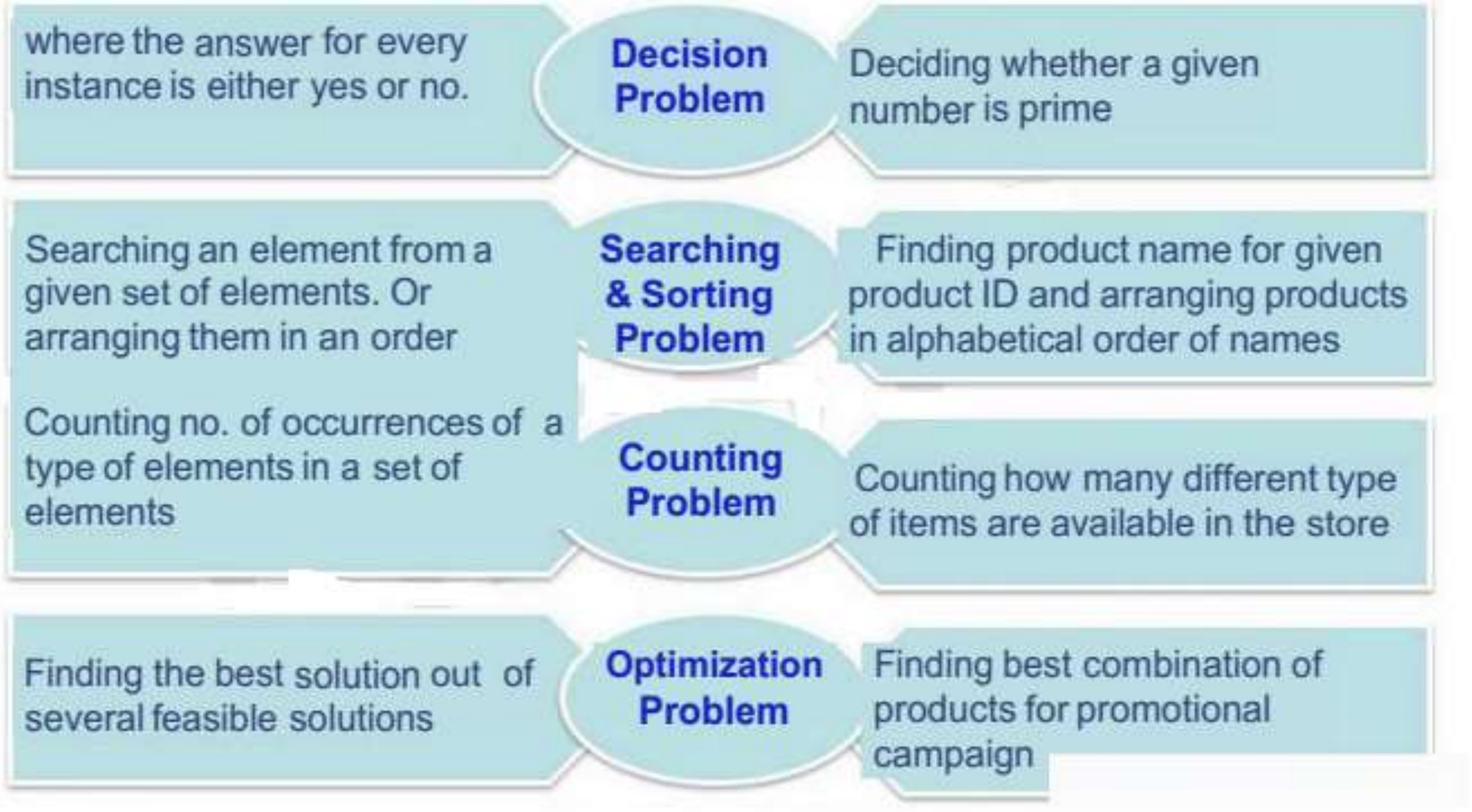
1.2 Identification of Computational Problems :

Computational Problems:

- **Computation** is the process of evolution from one state to another in accordance with some rules.
- **A computational problem** is a problem that a computer might be able to solve or a question that a computer may be able to answer.

1.2 Identification of Computational Problems :

Types of Computational Problems :



where the answer for every instance is either yes or no.

Decision Problem

Deciding whether a given number is prime

Searching an element from a given set of elements. Or arranging them in an order

Searching & Sorting Problem

Finding product name for given product ID and arranging products in alphabetical order of names

Counting no. of occurrences of a type of elements in a set of elements

Counting Problem

Counting how many different type of items are available in the store

Finding the best solution out of several feasible solutions

Optimization Problem

Finding best combination of products for promotional campaign

1.2 Identification of Computational Problems :

Computational Thinking:

- ‘Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.’
- Computational thinking is an approach to problem-solving that involves using a set of practices and principles from computer science to formulate a solution that’s executable by a computer.
- It’s not just for programmers. In fact, it’s applicable in a diverse array of fields.

1.2 Identification of Computational Problems :

Some examples of Computational Thinking:

Predicting climate change:

Predicting global climate change is only possible because of advanced computer models. According to the UK Met Office, ‘The only way to predict the day-to-day weather and changes to the climate over longer timescales is to use computer models.’

1.2 Identification of Computational Problems :

Some examples of Computational Thinking:

Assisting police, lawyers and judges:

- Computational Thinking has a long tradition in influencing the law, especially in the dream of providing a set of logical rules that can automate the process of reaching a verdict, its desire to minimize human discretion and maximize predictability of outcome.
- legal reasoning systems have been making inroads where they merely try to assist those making legal decisions.

Summary:

- A Computer is ***an electronic machine*** that can be programmed to accept data (*input*), and process it into useful information (*output*). Data is put into secondary storage (*storage*) for safekeeping or later use.

- The computer has evolved from a large-sized simple calculating machine to a smaller but much more powerful machine.

- ***Problem*** is a thing that requires logical thought and /or mathematics to solve.

- ***Problem solving*** is the systematic approach to define the problem and creating number of solutions.

Summary:

- Computers are built to solve problems with algorithmic solutions, which are often difficult or very time consuming when input is large.
- ***A computational problem*** is a problem that a computer might be able to solve or a question that a computer may be able to answer.
- ***Computational thinking*** is an approach to problem-solving that involves using a set of practices and principles from computer science to formulate a solution that's executable by a computer.

THANK YOU