***The 4 Stages of Computational Thinking:***

* ***Decomposition***

This consists of breaking down problems to smaller sections which further assist on solving issues quicker and making them more manageable.

An example of this in the real world is when you set up a chronological goal list with check boxes so when you get stuck on one of tasks you can always move to the next one without missing out on something because you forgot to go back to it.

* ***Abstraction***

This also consists of getting goals completed as soon as possible by looking at the most important tasks at hand when dealing with a to-do list and highlighting key details of the plan by using brainstorming as well.

An example of this in the real world is the true minimalism of the tube map

* ***Algorithm Design***

Algorithms is developing a step-by-step solution to the problem, a sequence of instructions for carrying out a task. In computing, algorithms are needed to design computer programs. They allow things like computers, smartphones, and websites to operate and make decisions.

An example this in the real world can be the recipe for baking a cake, the method we use to solve a long division problem or the process of doing laundry.

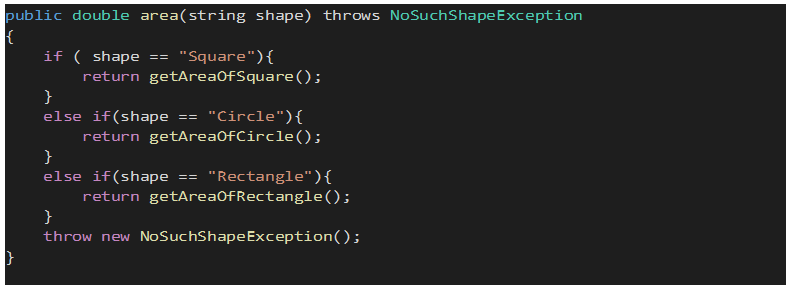
* ***Pattern Recognition***

When you find a pattern of something being repeated so for example you find a sequence of numbers in a pattern, if you find pattern of repetition for example you can problem solve easier than it would be to go over the same information. It is also easier to limit how many patterns there are so the work rate would be faster as you would have to go through less information.

An example of this in the real world can be things such as biometrics or just a set of information that is directed towards you like a license plate.

***Procedural Programming Language***

Procedural programming is breaking down a problem into sections in order to make it easier to solve problems due to them being segmented. The program data is in forms of variables and each function consists of computational thinking statements in order to solve a problem.

An example of Procedural Programming:

Examples of computer procedural languages are BASIC, C, FORTRAN, Java, and Pascal. Procedural languages are some of the common types of programming languages used by script and software programmers. They make use of functions, conditional statements, and variables to create programs that allow a computer to calculate and display a desired output.

***Object Orientated Language***

Object-oriented programming is a programming language paradigm (paradigm is a new method of thinking about a problem or situation). In OPP the code can be broken down and reused in a code where some properties or behaviours may be the same. This can be useful in improving the developer’s ability to quickly prototype software and just increase its life span and functionality since its being updated instead of being constantly remade.

Examples of object orientated languages are Java, C#, Python, Ruby, PHP and TypeScript. Object orientated languages are more advanced than procedural orientated programming and is commonly used by web designer for websites or even for databases for banks.

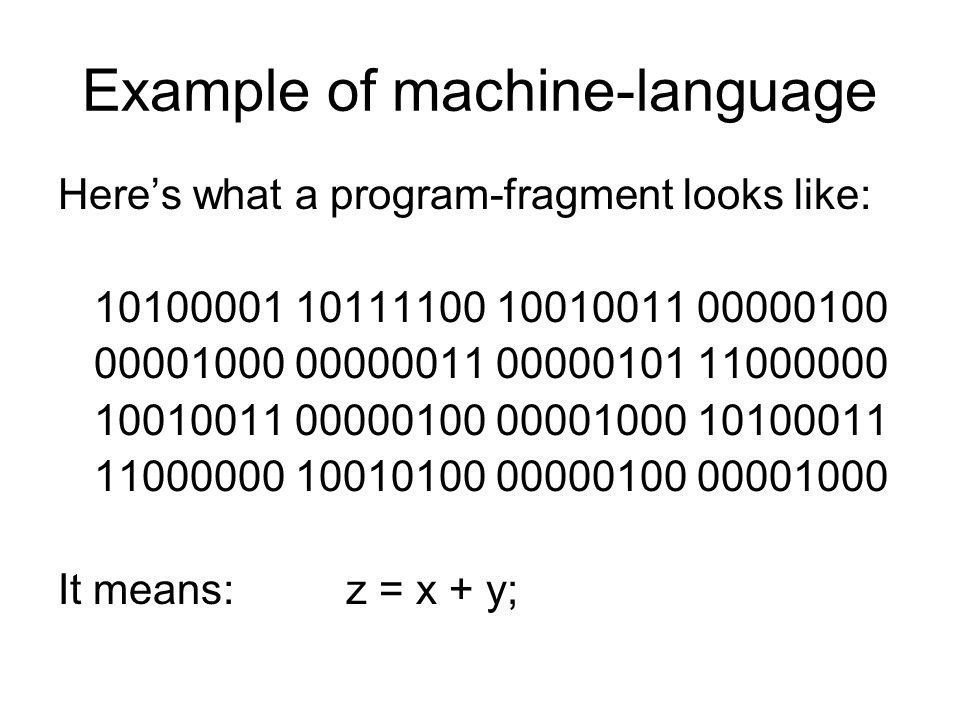
An Example of Object Orientated Languages:



Machine Language

Machine Languages is a collection of binary digits or bits that a computer reads and interprets. Machine language is the only language a computer is capable of understanding. Machine language is mostly used for physical hardware for it communicate with each component and let them work together.

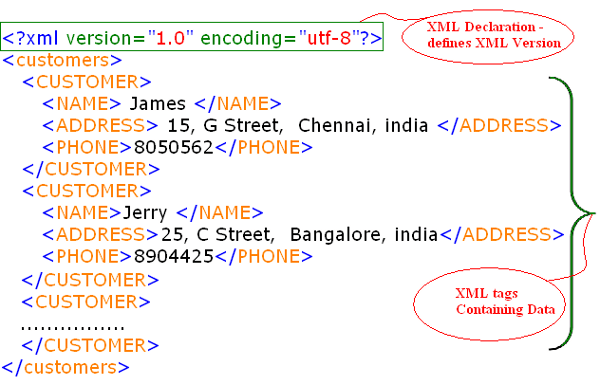
Examples of Machine Languages are C++, Java, and Virtual Basic. A computer cannot directly understand the programming languages used to create computer programs, so the program code must be compiled. Once a program's code is compiled, the computer can understand it because the program's code is turned into machine language.

An example of Machine Languages:

Mark-Up Language

Mark-Up Language is a computer language that consists of easily understood keywords, names, or tags that help format the overall view of a page and the data it contains.

To create any mark-up language file, any text editor can be used. A mark-up language is not a programming language. It is special markings, interspersed with plain text, which, if removed or ignored, leave the plain text as a complete whole. Mark-Up is mostly used to create websites that’s why the heavy use of HTML.

An example of Mark-Up Language: