

What is Computer Science?

Computer science is the study of computers and the way they work. It involves the following:

- Data storage and processing
- Network communication between systems
- Programming (writing code)
- and problem solving using efficient algorithms and programming.

Computer science is much more but for this purpose we'll leave it at that.

What is programming?

Programming is a way to provide computers instruction or an algorithm to complete a problem or a set of tasks.

What kind of tasks or problems can you think of?

- Basic math calculations.
- Web browsing.

These are two examples that are used often on a daily basis.

In our case we will be programming for the web, in other terms we will be developing code that will make our websites be more than just "static", boring, pages with more function and behaviors.

Core Programming Concepts

Programming languages and programming in general must span a minimum set of required ideas. The following, with the exception of algorithms (the developer comes up with these), are features that all program languages should have to be considered a programming language.

1. Algorithms

A step by step procedure or set of instructions design to reach an end result (solve a problem).

For example making coffee can be written as an algorithm. To keep it simple I will cover the main coffee making steps not including prerequisite steps. Feel free to elaborate on these to make the algorithm more complex.

- 1) Fill coffee maker with water
- 2) Set coffee carafe on coffee maker
- 3) Add Coffee
- 4) Brew coffee

2. Variables

Named containers for data

Example:

```
let score = 100;
```

3. Data Types

- Integer: 42, 3.14
- Float: 1.2, 5.4, 0.5
- String: "Hello"
- Boolean: true, false
- Array: [1, 2, 3]
- Object: { name: "Alex" } (JSON – similar but not the same and will be covered in greater detail later as necessary)

4. Operators

Arithmetic (math): +, -, *, /, %

Comparison: ==, !=, >, <, >=, <= (determines validity of a relationship)

Logical: &&, ||, ! (determines the combination validity of relationships)

Comparison: is x= 1 equal to y=1 – yes it is true

Logical: if x is true and y is true then it must be true

5. Control Flow

Conditionals: if, else (controls decisions, not the most accurate description but the simplest)

Loops: for, while (controls repetition)

5.1 If/else

If and else are exactly what they read like.

If something is true then do this, else if is not true do something else.

There are cases where you might want to check for the negation (false) of something.

5.2 for and while loops

These control repetition.

For loops are great when you have a predetermined number of cycles(loops) something should be done.

If you are unsure how many times you have to loop, a while loops works better.

*Instructor : use board to show *

6. Functions

Reusable section of code (a function can be an algorithm)

Example:

```
function greet(name) {  
  return "Hello, " + name;  
}
```

** A function is an algorithm if: **

- It has a clear input
- Set of clearly defined steps
- Has a clearly determined output

Reminder: Not all algorithms are functions, some span a diverse set of variables, data structures and other dependent functions.

Thinking Like a Programmer

1. Understand the problem (break the problem down to small steps)
3. Write pseudo-code (or code that does not use language specific syntax)
4. Translate pseudo-code into programming language specific syntax
5. Test and debug