Berklee Online
OLMSC-150
Lesson 1
Terms and Definitions

- **Requirements:** What a piece of software *should* do
- **Design:** How a piece of software will go about accomplishing its requirements
- **High-Level Language:** A programming language that abstracts away many details of the hardware to make programs easier to write and maintain. Examples: Javascript, Python, Swift
- **Low-Level Language:** A language that gives the developer more control over how code will actually be executed, but generally less programmer-friendly.
- **Machine Language:** The lowest level way of directing a computer, specific to the type of computer it's running on.
- **Compiled Language:** Code that must be *compiled* in advance to run on a particular type of computer. Generates an *executable* containing the machine language version of your program.
- **Interpreted Language:** A high-level language that runs code directly, rather than first compiling it. Requires an *interpreter* to run the code.
- **Literal:** Pre-existing data in the code, as opposed to something loaded at run time. "Literal" because they sometimes require special syntax (such as quotes) to indicate that they're data, as opposed to code.

Berklee Online
OLMSC-150
Lesson 1
Terms and Definitions

- **Data Type:** The kind of information that will be stored. Examples: numbers, strings
- **String:** A data type to store text
- **Integer:** A data type for storing whole numbers
- **Float:** A data type for storing any number
- **Variable:** A named piece of data that can be used and changed at run time
- **Function:** A chunk of named re-usable code, often taking parameters as input and returning a value
- **Parameter:** Data passed into a function. Also known as an argument.
- **And (&&)**: Boolean statement that evaluates to true when both of its arguments are true
- **Or (||):** Boolean statement that evaluates to true when either (or both) of its arguments are true
- **Exclusive Or (XOR):** Boolean statement that's true when exactly one of its arguments is true
- **Not (!):** Boolean statement that negates a single value, so false becomes true and vice-versa.
- **De Morgan's Law:** Boolean equivalence that says (!A && !B) is equal to !(A || B)