Notes to get started with VHDL.

- VHDL statements, keywords, identifiers, etc. are not case sensitive
- An identifier in VHDL is the name you give to a VHDL object such as an entity, architecture, signal, etc. Basic rules for identifiers are:
  - identifiers can contain only upper or lower case letters a-z, numerals 0-9, and underscore
  - the first character must be a letter
  - the last character cannot be an underscore
  - two successive underscores are not allowed
  - identifiers cannot contain spaces (i.e. the space character). Use underscore instead.
- Most statements end with a semi-colon
- Standard practice is to use indenting to increase readability
- Keywords that you immediately need to know and use are:
  - entity
  - is
  - port
  - in
  - out
  - end
  - architecture
  - of
  - begin
  - signal
- Logic operators are: and, or, nand, nor, xor, xnor
- Logic operators all have the same precedence and will be evaluated left to right unless parenthesis are used to show the desired order of evaluation.

An ENTITY is like a symbol on a schematic in that it defines the inputs and outputs of a component or circuit. Statements in the ARCHITECTURE define the functionality of the component in behavioral, functional, or structural fashion. For example, if a component is needed to implement this function: \( F = A \cdot B + C \) we can create the following entity and architecture for it.

```vhdl
entity firstFunction is
  port ( a, b, c : in std_logic;  
          f : out std_logic);
end firstFunction;

architecture behav_arch of firstFunction is
begin
  f <= ( a and b ) or c;
end behav_arch;
```

On the class web page under Notes and User Information, third line down, is an example VHDL file for an XOR gate created in a structural fashion using two AND gates and an OR gate that are defined first and then “wired” together using a netlist. That could be done for \( F = A \cdot B + C \) also (can you do it? Try it for practice).