Carlos Hernandez

Professor Dinesh Sthapit

CSC120

10 April 2021

The code downloaded from Devoney makes public classes of different logic gates like simple not, and, or and makes them callable with return values. NOT returns the opposite of the Input, OR checks the input state to be true in either of the input values and returns true if it does, and AND only returns true if both InputA.State AND InputB.State are true. The composite functions are a bit more complicated, NAND returns true only if the values are different, by going through an AND gate and then a NOT gate. NOR does something similar, it goes through an OR gate and then inverts it, so in the end it only returns a true output or 1 if none of the values are true. XOR gates check if the input has an odd amount of 1 or true inputs, in which case it returns true, and the final composite gate, XNOR inverts the end of that, so it would only return true in cases where the number of true inputs are even, including zero.

Devoney uses object oriented programming to make it so he can instantiate his Simple class gates like AND or NOT in his composite gates, and combines those different gates to make the composite ones, like NAND for example. You can see this if you look into his source code.