```vhdl
----------------------------------------------------------------------------------
-- Name: Capt Jeff Falkinburg
-- Date: Spring 2016
-- Course: ECE 281
-- File: JK_Flip_Flop.vhd
-- HW:   Lecture 19
-- Purp: Basic JK Flip-Flop
--    Write an HDL Module for a JK Flip Flop:
--       Inputs:  J, K, Clk
--       Output:  Q
--       ↑  -->   Q Keeps old value if J=K=0 (i.e. Latch)
--                Q = 1  if J=1 (i.e. Set)
--                Q = 0  if K=1 (i.e Reset
--                Q = Q' if J=K=1 (i.e. Toggle)
-- Doc:  None
-- Academic Integrity Statement: I certify that, while others may have
-- assisted me in brain storming, debugging and validating this program,
-- the program itself is my own work. I understand that submitting code
-- which is the work of other individuals is a violation of the honor
-- code.  I also understand that if I knowingly give my original work to
-- another individual is also a violation of the honor code.
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity JK_Flip_Flop is
    Port ( J : in  STD_LOGIC;
           K : in  STD_LOGIC;
           Clk : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           Q_not : out  STD_LOGIC);
end JK_Flip_Flop;

architecture Behavioral of JK_Flip_Flop is
   signal Q_Sig : STD_LOGIC;
begin
   process(clk)
   -- variable Q_sig : STD_LOGIC;  -- Could also assign a variable name
   begin
      if (rising_edge(clk)) then
         if (J = '0' and K = '0') then
            Q_Sig <= Q_Sig;      -- Latch...No Change
         elsif (J = '1' and K = '0') then
            Q_Sig <= '1';        -- Set
         elsif (J = '0' and K = '1') then
            Q_Sig <= '0';        -- Reset
         elsif (J = '1' and K = '1') then
            Q_Sig <= not Q_Sig;  -- Toggle
         end if;
      end if;
--    Q <= Q_Sig;          -- variable to output assignments would have to happen here since
--    Q_not <= not Q_Sig;  -- variables only exist within the process
   end process;
   Q <= Q_Sig;
   Q_not <= not Q_Sig;
end Behavioral;
```