When using the Xilinx ISE 13.4, there is a process to complete before a person can begin to code the Spartan 6 FPGA, this is a simple list going over said process.

0. Open Xilinx ISE 13.4 via the Desktop Icon (13.4 for this walk-through)

1. Creating a new source file (and associated processes) by choosing Open Project from the ISE start menu in the lower left hand corner.

To create a new project, go to “File” and select “New Project”. Give it a name and choose a directory for both the location of the project and the working directory, both can go to the same place. For the Top-level source type select HDL. Select next and your will see a new window where many options will be available; this allows you to choose the the board, product, family, package, and speed of the fpga in use. In this case, choose the SP605 evaluation platform for the Evaluation Development Board, which will fill most of the options for you. Leave everything else alone and press ok.

2. Programming

1. The code I used was found online on the web video: <https://www.youtube.com/watch?v=7GgAukBxEmQ> Although it is coded for a different board, it provides code that can work for the sp605 evaluation kit. However, it is inaccurate in one aspect, the pins for the constraints code. (See part b). This code needs to be entered into the main verilog file of the project.
2. To add a constraints file, right click the main file of the project in the menu at upper left, and press ““Implementation Constraints File.” You must create a constraints file in order to define variables in the code representing parts. To do this you must find the pins of your given board to properly code for the board. In this case, the LED and Dip switch pins can be found here: <https://www.xilinx.com/support/documentation/boards_and_kits/sp605-schematic-xtp067-rev-e.pdf>
3. Example code is below (save with Ctrl-S)

NET "switch[0]" LOC = "Y6";

NET "switch[1]" LOC = "W6";

NET "switch[2]" LOC = "E4";

NET "led[0]" LOC = "D17";

NET "led[1]" LOC = "AB4";

NET "led[2]" LOC = "D21";

3. Synthesizing, implementation, and generation

After completing the code, you must synthesize your design, implement it, and then generate the a programming file. To do this, select the .v file from the left hand menu and go to the box below to select the “Synthesize-XST” (double-click), “Implement Design” (double click) and “Generate Programming File”(double-click). Click each of these in this order and wait for each to complete before clicking the next one as only one process can be executed at a time.

4. Loading code into FPGA

To begin loading your program to the FPGA you must select “Configure Target Device” from the same menu used in Step 3. Open the iMPACT program, or select “ok” if no iMPACT file exists yet. Connect the JTAG USB cable to the PC and power on the Xilinx board. In the iMPACT program double-click “Boundary Scan”. Right click the white environment and select “add a xilinx device” and navigate the windows explorer window to find your project .BIT file. This is the first file created in Step 1. Click the new device on the board, and it will turn green. Click “Initialize Chain” from the icon menu (four green circles by the bug and four gray squares) close the two new windows that pop up and then double click the device with the same name as the chip on your FPGA (in this case: xc6slx45t). Navigate the new windows explorer window that appears to find the .BIT file of your project and select it. Close the window that pops up. Finally, right click the device with your chip’s name on it and select “Program.” If successful, a blue box stating “Program Succeeded” should appear.

5. Test program

You are now ready to test your program on the FPGA. To do this, toggle the DIP switches defined in the constraints file. They are located near the USB connection to the Xilinx embedded kit board. The LED corresponding to the individual switch should light when you toggle the switch.