____ 5. Call the OpenCL function to **Create** the cl::Kernel object from the **program** after the comment "Exercise 2 Step 2.5."

*You can use the kernel_name variable in your argument which is set to "SimpleKernel" and should match your kernel name.*

____ 6. Call the function to setup the Kernel arguments four times, once for each buffer: **Buffer_In**, **Buffer_In2**, and **Buffer_Out**, and another time to pass in the variable vectorSize. Do this after the comment "Exercise 2 Step 2.6."

____ 7. After the comment "Exercise 2 Step 2.7," **launch** the kernel using **enqueueTask**.

*Use the command queue we created in the first exercise.*

*This command will execute the kernel on the OpenCL device when you are not in emulation mode.*

____ 8. Following the comment "Exercise 2 Step 2.8," read back the results of the kernel execution by copying the contents of the Buffer_Out buffer to the array **Z**.

*The contents of Z will be verified later in **main**.*

*Make sure the blocking argument of the call is set to CL_TRUE. This guarantees that after read function, Z will be ready to be used.*

____ 9. After the comment "Exercise 2 Step 2.9," write the **same** math operation you have in the kernel. This time, however, it should operate on X[i] and Y[i] and store the result into CalcZ[i].

*Here we're doing the same calculation the kernel is doing, except that we're using regular C++. The contents are then stored in CalcZ, which will be compared to the contents of Z.*

____ 10. Save **main.cpp.**



Copyright © 2018 Intel Corporation