# Lab Assignment 3
# Due Oct 22, 2023 at 11:59pm

## 1 Objective

The purpose of this lab is to introduce the student to the pinned memory APIs long with CUDA Streams by implementing vector addition. The student will implement pinned memory allocation and experience the usage of CUDA Streams by writing the CUDA calls in the host code.

## 2 Instructions

The code template in `template.cu` provides a starting point and handles the import and export as well as the checking of the solution. Students are expected to insert their code to where demarcated with `//@@`. Students are expected to leave the other code unchanged.
Edit the skeleton code to perform the following:

- Call pinned memory allocators
- Partition data into per-stream segments
- Copy data from host to device asynchronously
- Initialize grid and block dimensions
- Call the CUDA kernel
- Copy data from device to host asynchronously
- Synchronize
- Free streams and device memory

Compile the template with the provided `Makefile`. The executable generated as a result of compilation can be run using the following code:

```
./StreamVectorAdd_Template -e <expected.raw> -i <input1.raw>,<input2.raw>
-o <output.raw> -t matrix
```

where `<expected.raw>` is the expected output, `<input0.raw>,<input1.raw>` is the input dataset, and `<output.raw>` is an optional path to store the results.

`README.md` has details on how to build `libgputk`, `template.cpp` and the dataset generator.

## 3 What to Turn in

Submit a report that includes the following:

1. How many bytes of data (both read and write) are moved from host to device when using CUDA Streams?
2. When should one use pinned memory? Explain.
3. Your version of `template.cu`.

4. Execution times of the kernel with the input data generated by the dataset generator (in a table or graph). Please include the system information where you performed your evaluation. For time measurement, use `gpuTKTime_start` and `gpuTKTime_stop` functions (You can find details in `libgputk/README.md`).

5. Execution times of the kernel for the largest input size (96000 elements) with different numbers of CUDA Streams (up to 32). Please include the system information where you performed your evaluation. For time measurement, use `gpuTKTime_start` and `gpuTKTime_stop` functions (You can find details in `libgputk/README.md`).