Create CUDA Project For CSC3150 Assignment 3

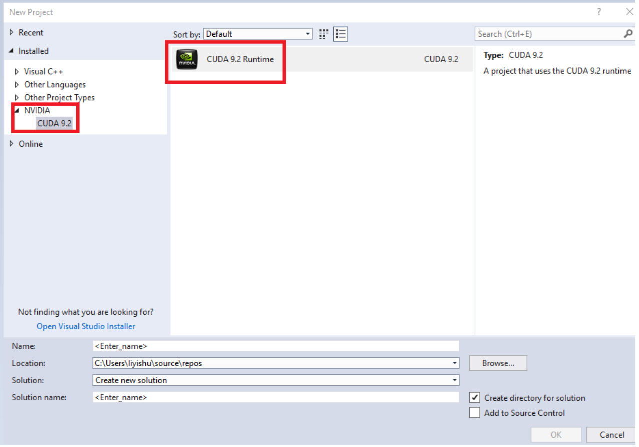
Environment:

- Win 10
- VS2017
- CUDA 9.2
- NVIDIA Geforce GTX 1060
- Compute capacity: 6.1

Configuration steps:

1. Launch VS2017

2. Create CUDA project
(If you're using PCs in TC301, change the project storing location in this step. Please ensure storing your CUDA project in D disk, rather than C disk, where will auto delete users' data every 30 days.)



Sort by: Default

Type: CUDA 9.2

A project that uses the CUDA 9.2 runtime

Name: <Enter_name>

Location: C:\Users\liyishu\source\repos

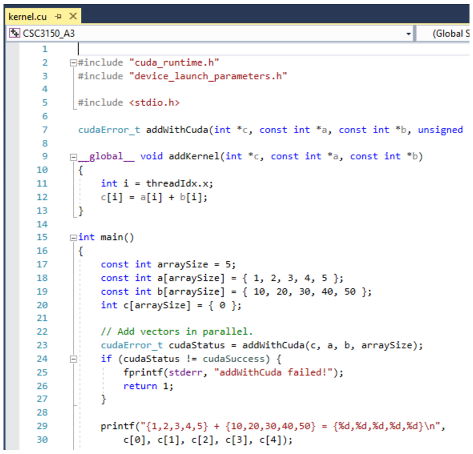
Solution: Create new solution

Solution name: <Enter_name>

☑ Create directory for solution

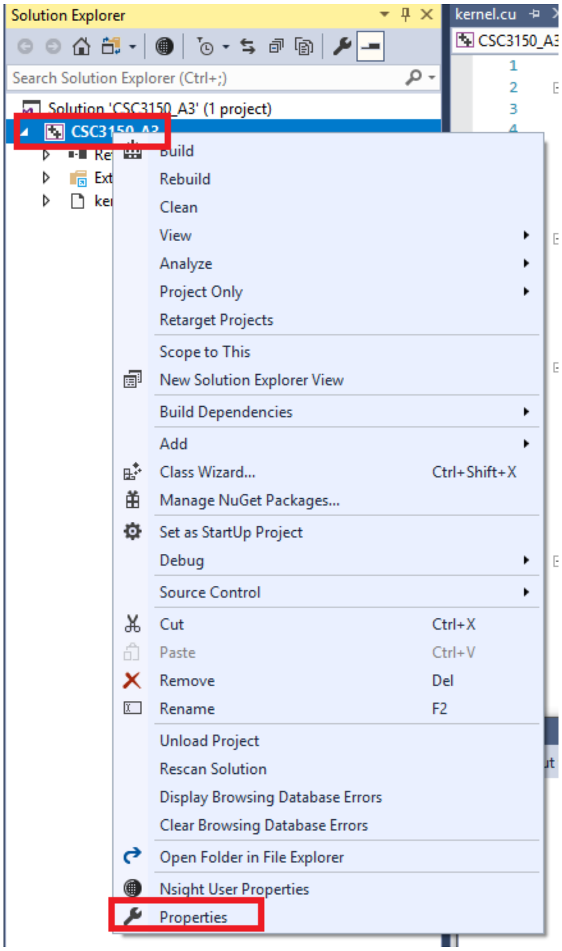
☐ Add to Source Control

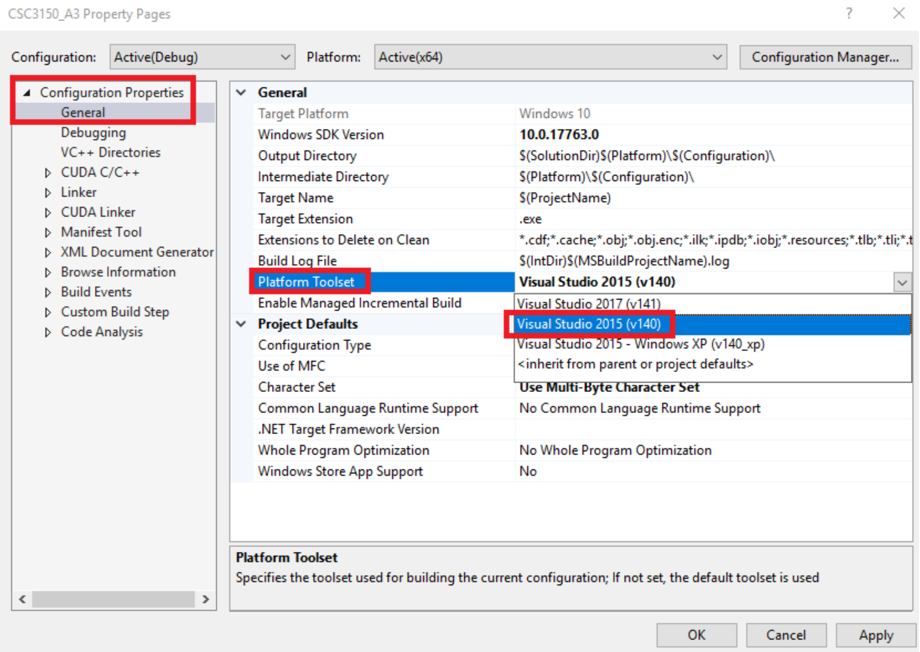
3. There will be a "kernel.cu" automatically created.



```

#include "cuda_runtime.h"
#include "device_launch_parameters.h"

#include <stdio.h>

cudaError_t addWithCuda(int *c, const int *a, const int *b, unsigned

__global__ void addKernel(int *c, const int *a, const int *b)
{
    int i = threadIdx.x;
    c[i] = a[i] + b[i];
}

int main()
{
    const int arraySize = 5;
    const int a[arraySize] = { 1, 2, 3, 4, 5 };
    const int b[arraySize] = { 10, 20, 30, 40, 50 };
    int c[arraySize] = { 0 };

    // Add vectors in parallel.
    cudaError_t cudaStatus = addWithCuda(c, a, b, arraySize);
    if (cudaStatus != cudaSuccess) {
        fprintf(stderr, "addWithCuda failed!");
        return 1;
    }

    printf("{1,2,3,4,5} + {10,20,30,40,50} = {%d,%d,%d,%d,%d}\n",
        c[0], c[1], c[2], c[3], c[4]);
```

4. Select and right click the project, open its properties. Update the platform toolkit as 'Visual Studio 2015(v140)'





CSC3150_A3 Property Pages

Configuration: Active(Debug) Platform: Active(x64) Configuration Manager...

General

Target Platform	Windows 10
Windows SDK Version	**10.0.17763.0**
Output Directory	$(SolutionDir)$(Platform)\$(Configuration)\
Intermediate Directory	$(Platform)\$(Configuration)\
Target Name	$(ProjectName)
Target Extension	.exe
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.obj.enc;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*.tli;*.t
Build Log File	$(IntDir)$(MSBuildProjectName).log
Platform Toolset	**Visual Studio 2015 (v140)**
Enable Managed Incremental Build	

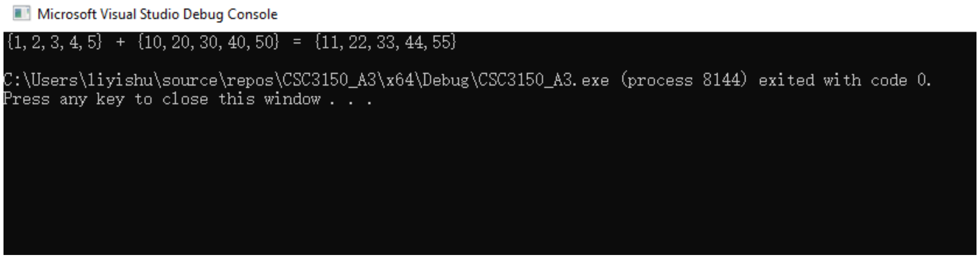
Visual Studio 2017 (v141)
Visual Studio 2015 (v140)
Visual Studio 2015 - Windows XP (v140_xp)
<inherit from parent or project defaults>

Project Defaults

Configuration Type	
Use of MFC	
Character Set	**Use Multi-Byte Character Set**
Common Language Runtime Support	No Common Language Runtime Support
.NET Target Framework Version	
Whole Program Optimization	No Whole Program Optimization
Windows Store App Support	No

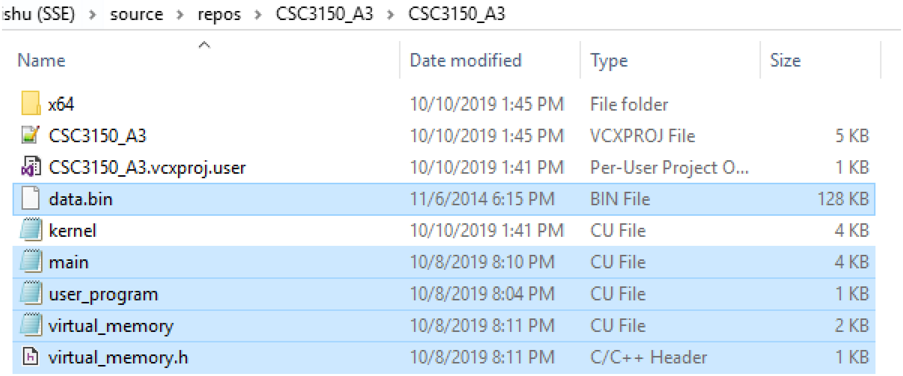
Platform Toolset
Specifies the toolset used for building the current configuration; If not set, the default toolset is used

5. Press “Ctrl”+”F5” to run the project. If the calculation results being displayed correctly in console, it means your CUDA project settings is correct.



```
{1, 2, 3, 4, 5} + {10, 20, 30, 40, 50} = {11, 22, 33, 44, 55}

C:\Users\liyishu\source\repos\CSC3150_A3\x64\Debug\CSC3150_A3.exe (process 8144) exited with code 0.
Press any key to close this window . . .
```

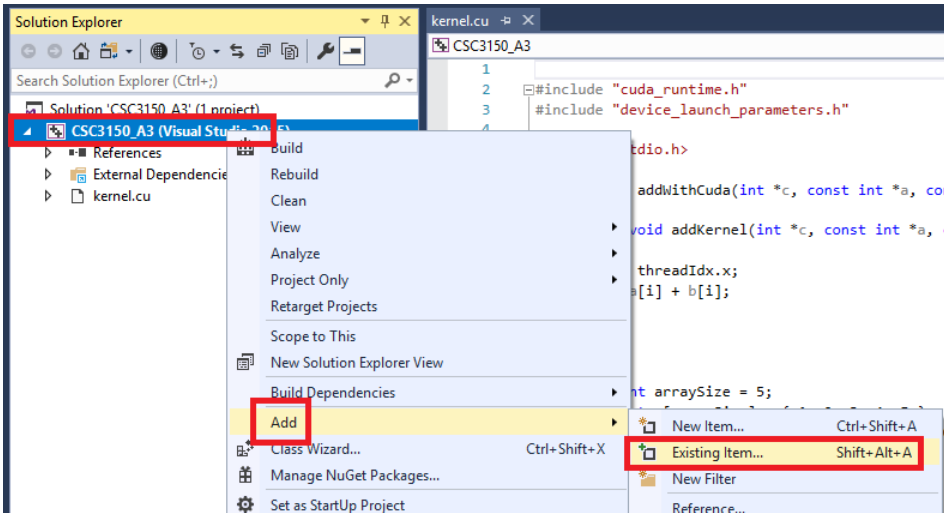
6. Copy provided source code into your project folder. (‘main.cu’, ‘user_program.cu’, ‘virtual_memory.cu’, ‘virtual_memory.h’, ‘data.bin’)



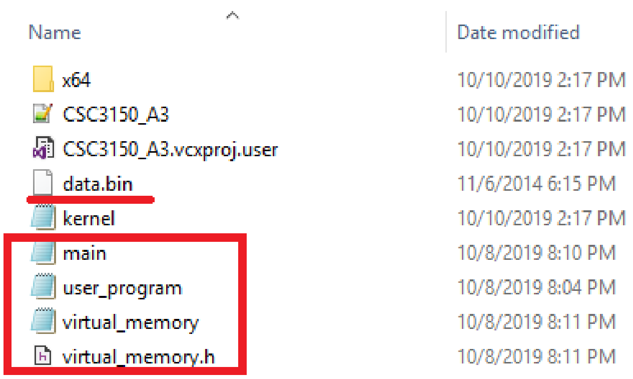
ishu (SSE) › source › repos › CSC3150_A3 › CSC3150_A3

Name	Date modified	Type	Size
x64	10/10/2019 1:45 PM	File folder	
CSC3150_A3	10/10/2019 1:45 PM	VCXPROJ File	5 KB
CSC3150_A3.vcxproj.user	10/10/2019 1:41 PM	Per-User Project O...	1 KB
data.bin	11/6/2014 6:15 PM	BIN File	128 KB
kernel	10/10/2019 1:41 PM	CU File	4 KB
main	10/8/2019 8:10 PM	CU File	4 KB
user_program	10/8/2019 8:04 PM	CU File	1 KB
virtual_memory	10/8/2019 8:11 PM	CU File	2 KB
virtual_memory.h	10/8/2019 8:11 PM	C/C++ Header	1 KB

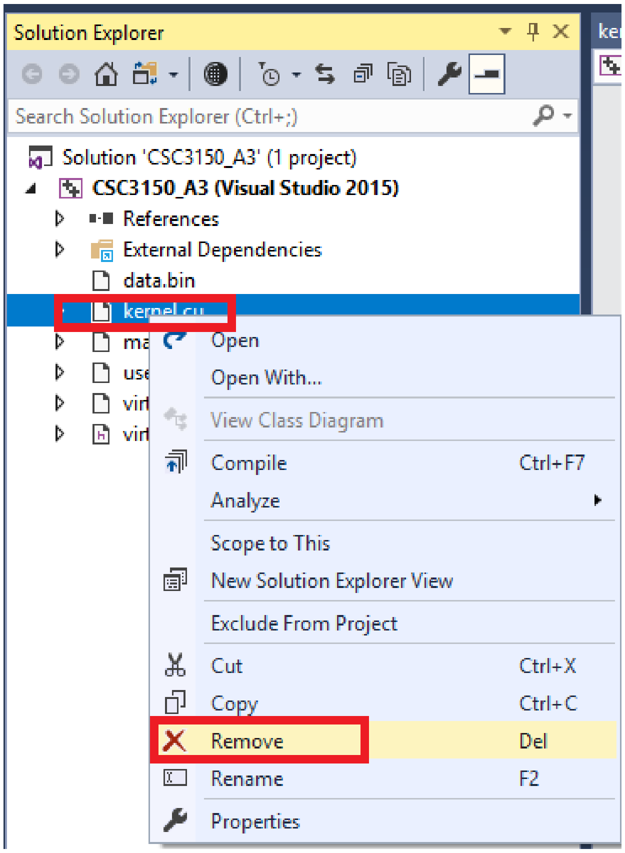
7. In Visual Studio, select and right click your project, add exiting items (the source codes) into your project. And remove the ‘kernel.cu’ from your project.

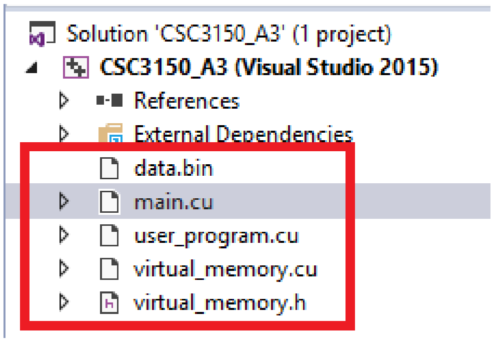


```
#include "cuda_runtime.h"
#include "device_launch_parameters.h"

tdio.h>

addWithCuda(int *c, const int *a, co

void addKernel(int *c, const int *a,

threadIdx.x;
[i] + b[i];

nt arraySize = 5;
```

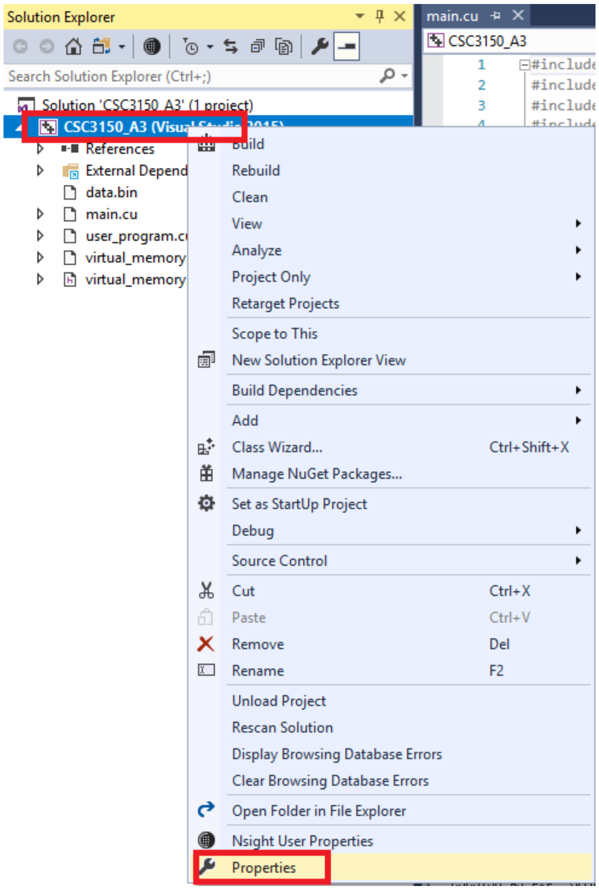


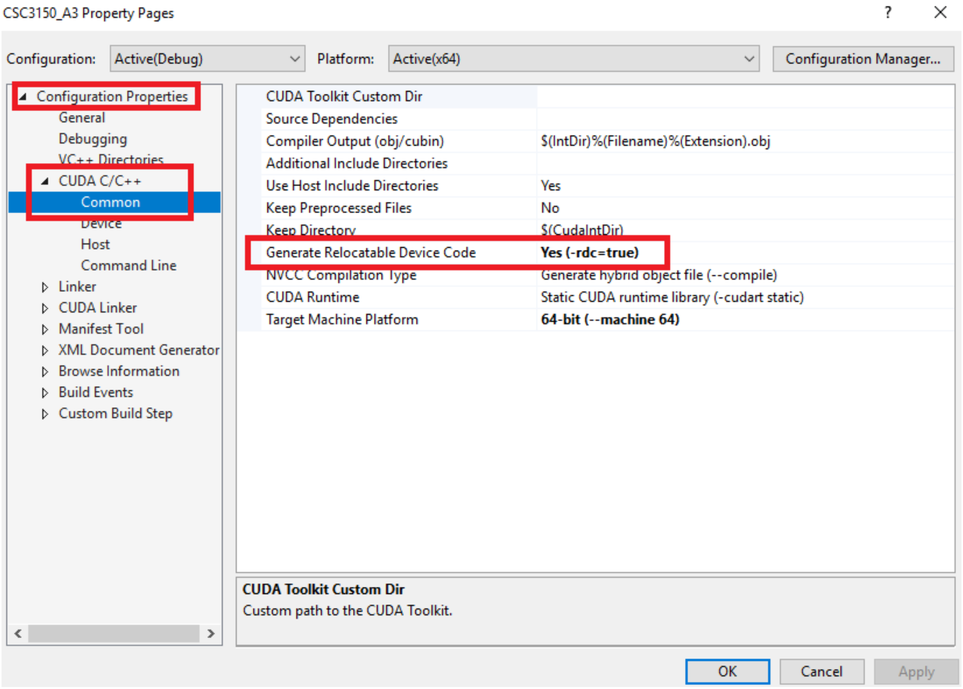
Name	Date modified
x64	10/10/2019 2:17 PM
CSC3150_A3	10/10/2019 2:17 PM
CSC3150_A3.vcxproj.user	10/10/2019 2:17 PM
data.bin	11/6/2014 6:15 PM
kernel	10/10/2019 2:17 PM
main	10/8/2019 8:10 PM
user_program	10/8/2019 8:04 PM
virtual_memory	10/8/2019 8:11 PM
virtual_memory.h	10/8/2019 8:11 PM





8. Select the project and right click it. Open its properties, allow the CUDA files to generate relocatable device code.



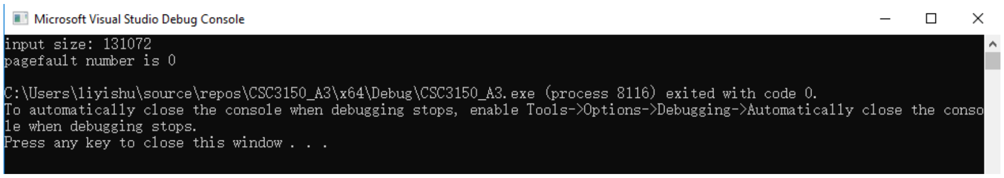


Configuration: Active(Debug)   Platform: Active(x64)

CUDA Toolkit Custom Dir	
Source Dependencies	
Compiler Output (obj/cubin)	$(IntDir)%(Filename)%(Extension).obj
Additional Include Directories	
Use Host Include Directories	Yes
Keep Preprocessed Files	No
Keep Directory	$(CudaIntDir)
Generate Relocatable Device Code	**Yes (-rdc=true)**
NVCC Compilation Type	Generate hybrid object file (--compile)
CUDA Runtime	Static CUDA runtime library (-cudart static)
Target Machine Platform	**64-bit (--machine 64)**

CUDA Toolkit Custom Dir
Custom path to the CUDA Toolkit.

9. For those three cuda files (.cu), right click to compile them. (Or use ‘Ctrl’+’F7’).
   Then press “Ctrl”+”F5” to run your program.
   For the provided template, you will get output as below:



```
input size: 131072
pagefault number is 0

C:\Users\liyishu\source\repos\CSC3150_A3\x64\Debug\CSC3150_A3.exe (process 8116) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

10. In “virtual_memory.cu”, start your programming.