Assignment #1

Goal of Assignment

1. Programming `vectorAdd()` on the GPU
   - Skeleton code is given
     - Including Programming Flow Guide
   - Must run `vectorAdd()` function on GPU
2. Draw Solid Teapot in an OpenGL Window
   - Skeleton code is given
     - Including OpenGL initialization routines
   - Call “Drawing **Solid Teapot**” function
   - Change the title of window to your student ID and name
     - Ex) 2015000000 박지혁

※ **Submission Due Date : 09/23**

Compile and Run on GPU Server

Both assignments must run on GPU Server

1. Connect to GPU server

2. Transfer your code to GPU server or write code on GPU Server

3. Compile and run your assignment

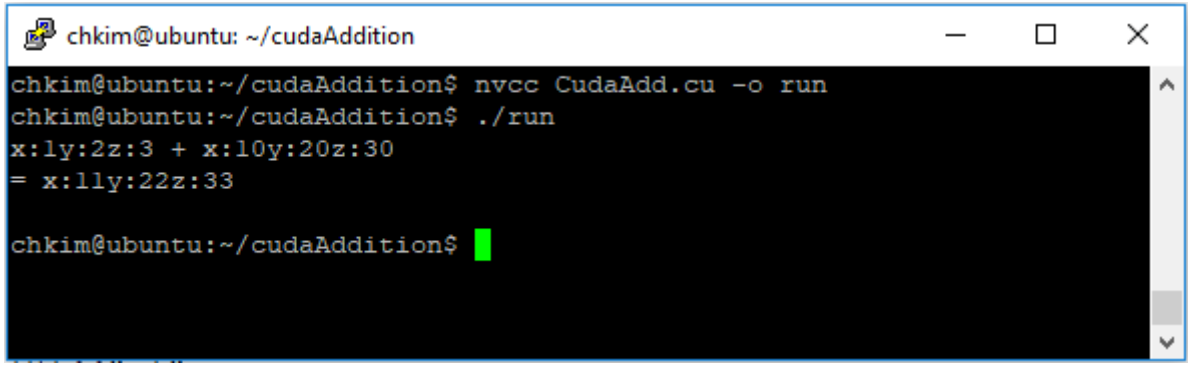
4. Submit your source code and running image

CUDA Assignment

- **Programming `vectorAdd()` on GPU**

$$\vec{A} + \vec{B} = \vec{C}$$

  - `vectorAdd()` function must run on the GPU
  - Compile and print result on GPU Server



Write a Device Kernel Function

- **Add two 3D vectors**
  - Use 3D vector structure.

```
struct vec3
{
    float x,y,z;

};
```

- **Function must run on the GPU**

CUDA Assignment

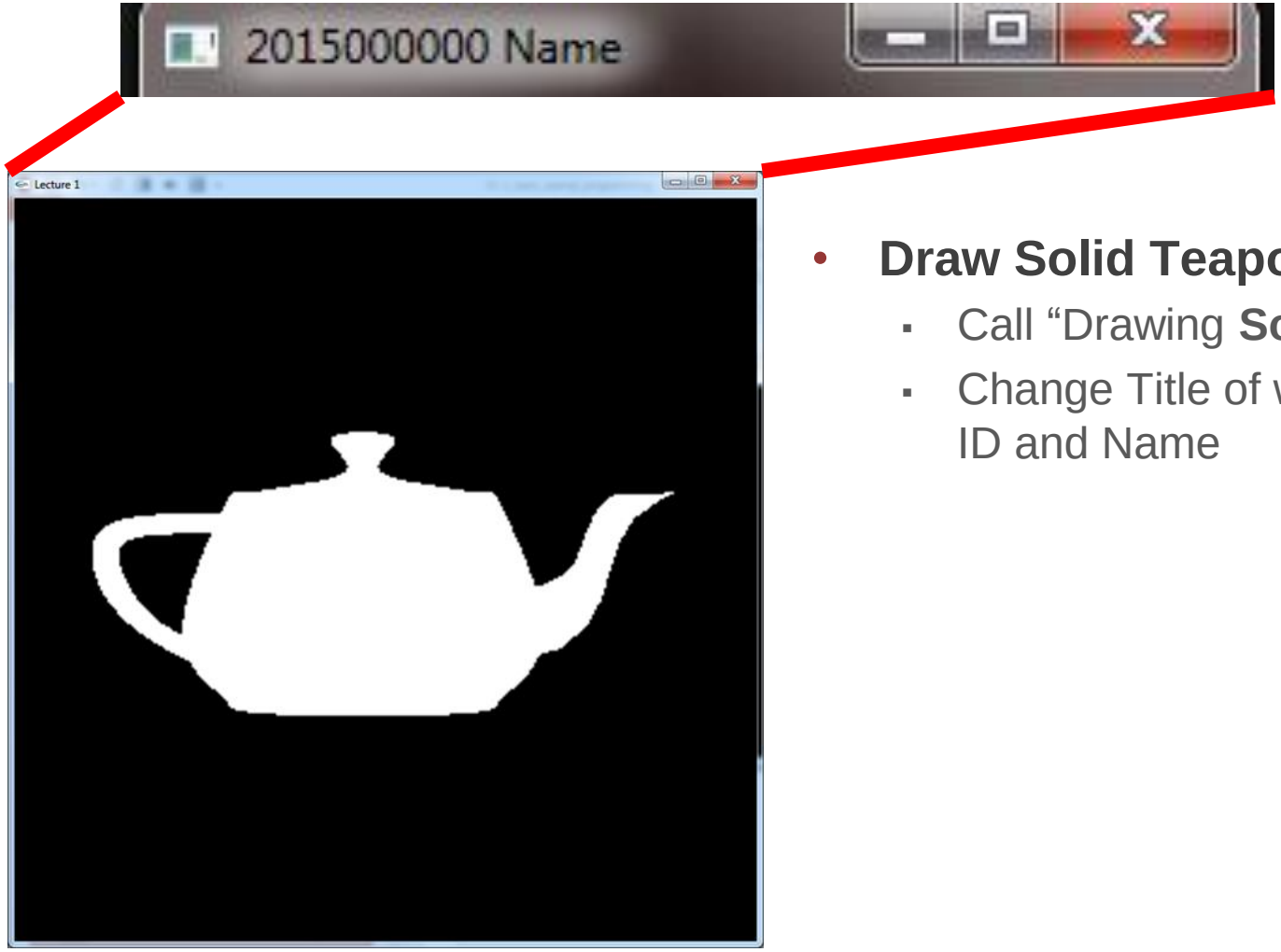
Skeleton Code (1)

```
#include "cuda_runtime.h"
#include <stdio.h>
#include <stdlib.h>


__global__ void vectorAdd(/*in&out arguments*/)
{
        int tid = threadIdx.x;
        /*      1-1. write vector addition function                     */
}


int main( void )
{
        /*      2-1. Check whether a proper device is mounted           */

        /*      2-2. Declare Host and Device pointer variables          */

        /*      2-3. Allocate Host memory                               */

        /*      2-4. Allocate Device memory                             */
```

Skeleton Code (2)

```
/*    2-5. Check that memory is allocated well on Device                    */

/*    2-6. Setup Input values to host array                                 */

/*    2-7. Copy memory for Input array from Host to Device                  */

/*              2-8. Call Kernel Function with <<<1, 1>>>                   */
vectorAdd<<<1,1>>>(/*in&out arguments*/);

/*    2-9. Copy memory for Result from Device to Host                       */

/*    2-10. Print Results                                                   */

/*    2-11. Release Host and Device memory                                  */

return 0;

}
```

OpenGL Assignment



- **Draw Solid Teapot on the X Window**
  - Call “Drawing **Solid Teapot**” Function
  - Change Title of window to your Student ID and Name

OpenGL Assignment

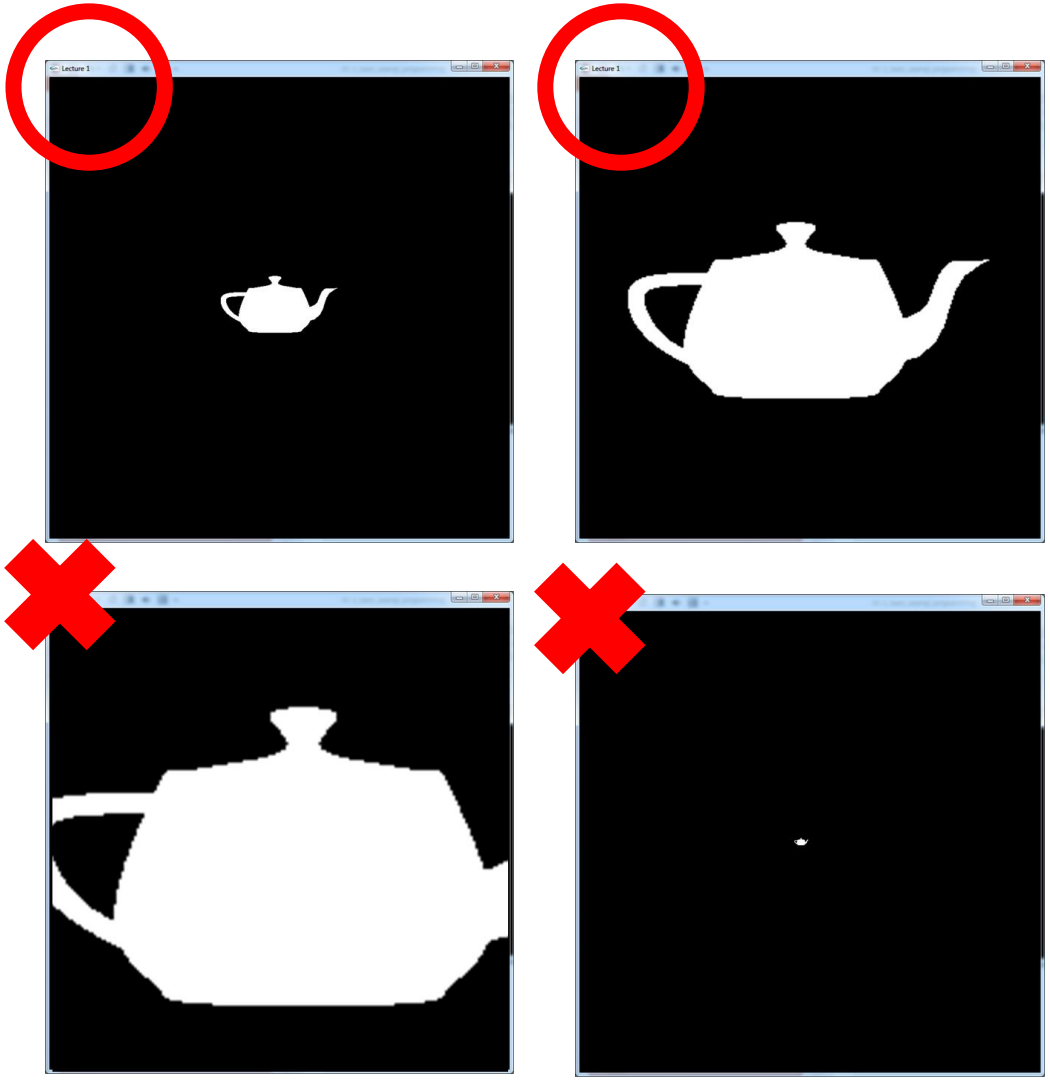
Call “Drawing Solid Teapot” Function

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
        //Draw Call Here
    glXSwapBuffers(dpy, win);
}

int main(int argc, char *argv[])
{
        /*Create Window*/


        while(1)
        {
            display();
        }
}
```

- **Write draw function**
  - You can draw solid teapot with two(**Draw, Initialize)** function.
    - Hint: **GLUT**

Caution

- Whole part of Teapot must be on the Window
- Teapot must be **visible** size



Submit the Assignment

- **Submit the zip file @ Blackboard**
  - File name must be “Assignment1_StudentID_Name.zip”
    - Ex. Assignment1_2015000000_박지혁.zip
  - Zip file must include two folder
    - CUDA
    - OpenGL
  - Each assignment folder must include
    - Src file
    - Result running Image file