Real Time system Assignment 1:

By: Sanket Patel and Thomson Philip  
Prof: Jeff Fortuna  
  
**Introduction:** We will be working on the reverb effect for this assignment.

**Challenges faced:   
Asio4All setup-** we were only getting garble noises from asio for all the options and not able to solve it properly due to inexperience with audio system related to programming.  
**Reverb Setup**- functional problems faced while setting up the reverb effect where implementation of formula of comb and all pass filter in c# and understanding of the whole project contains driver and different types of classes which executes as one.

**Process:** We derived the formula for the comb and all pass filter from the delay effect.

* The comb filter formula was y[n] = x[n – d] + gy[n - d]
* in c# Comb Formula is delayOut = \_delayBuffer[index, (\_counter - \_FBDelay)] + (float)0.708 \* \_delayFBbuffer[index, (\_counter - \_FBDelay)];
* the all pass filter formula y[n] = -gx[n] + x[n - d] + gy[n – d]
* in c# all pass filter is delayIn = -(float)0.708 \* \_delayRBbuffer[index, \_counter] + \_delayRBbuffer[index, (\_counter - \_FBDelay)] + (float)0.708 \* \_delayFBbuffer[index, (\_counter - \_FBDelay)];
* according to diagram of reverb, the typical value of g is 0.708

We saved the result from the comb filter inside delayOut which we later transferred to a delay buffer \_delayRBbuffer[index, \_counter]. We used this buffer while we calculated the value for the all pass filter, which was later stored in \_delayFBbuffer[index, \_counter] . We passed the values to **OutputL** and **OutputR** as delayOut + 2 \* delayIn;

**Result:**

The result of our code was somewhat white noise which we have attached along with the submission. Due to challenge we faced after4All setup, we were not completely verified whether is software issue or it is part of the code.