# Name: Corinne Green

# PE12 - Thread Control

## Objective:

Create a simple threaded program that uses join to control thread execution.

## Child Class:

1. Create a class named Child. This class will serve to hold the threaded code.
2. Create an attribute to hold a child’s name. This attribute will hold a string value used to identify the thread.
3. Create a “saying” attribute that will hold the child’s saying
4. Create a parameterized constructor that will accept two string values and will assign them to the name and saying attributes.
5. Within the class create a method named whine. This method will be public, will return nothing, and will have no parameters. The method will contain a loop that will print the child’s name and saying on ONE line. For example: “Sue: That’s my toy!” The loop will execute 5 times.

## Test Class:

1. The test class will create an array that can hold 3 thread objects.
2. To set up to test, do the following:
   1. Create three Child objects with their own unique name and saying
   2. Create three new Threads in the array. Each one should use a different child object’s whine method.
   3. DO NOT START THE THREADS YET.
3. Create a second loop to start all of the threads.
4. After starting all of the threads, have the main method print the message “Knock it off!” (as if it were the parent of these children).

(continued on back)

1. Where does the parent’s comment appear in the midst of all the whining?

The parent’s comment appears *once*. Its location amidst all of the wining changes each time the program is run. Sometimes it appears after a child’s 5 saying iterations, sometimes it appears before one child gets to finish its 5 iterations.

1. Now change the Child class and get the children to whine in sequence by using Sleep to cause a thread to give up the CPU. What do you see in the output after this change?

Each child is able to successfully complete their thread (5 saying iterations) without interruptions.

1. Now in the main class, use join on each thread to cause the “Knock it off” comment to appear after all the whining. What code did you need to add to do this?

In my foreach loop I added

t.Join();

Console.WriteLine("Knock it off!");

Now each child is told to “Knock it off!” once their thread is finished.

**Make sure you follow the coding standards for all code you create.**

Submit your code and completed questions to the PE12 dropbox

***Rubric:***

Note: code that fails to compile gets you an automatic zero for the entire assignment. Check your work before you submit it!

|  |  |  |
| --- | --- | --- |
| Criteria | Max Pts | Earned Pts |
| Code compiles and executes | 30 |  |
| Code solves the stated problem | 30 |  |
| Coding standards are followed | 20 |  |
| Questions answered correctly (if applicable) | 20 |  |
| Deductions:   * Code does not compile * Nothing submitted | -100  -100 |  |
| Total Grade: | 100 |  |