# CS 7310 Python for CS and AI Midterm Exam

## NumPy Measurement

Compare the performance of the built-in function sum() that sums all the values in a 2D NumPy array vs. using two loops to visit all the cells of the array and iteratively adding the values.

Steps:

* Use the numpy arange ( ) function to create a 1D numpy array (call it arr1d) of 100 million floating point numbers. The array will contain 100\_000\_000 floating point numbers.
* Convert the 1D array to a 2D array with 10 columns using:
  + ncols = 10
  + arr2d = np.reshape(arr1d, (-1, ncols))
* Use the timing technique shown in the NumPy module to measure the time it takes NumPy to compute the sum of all values in arr2d
* Write a function called compute\_sum(arr) that takes a numpy 2d array as parameter and uses two loop to iterate over all the values in the rows and columns and compute the sum and return the sum
  + Hint: use the shape property of numpy arrays to determine the number of rows and columns
  + Hint: use NumPy’s arr[row, col] to get the value in cells of the array and compute the sum
* Use the timing technique to measure the time it takes to compute the sum iteratively
* Be sure to check that the sum is the same for both methods

Display

* How much faster (or slower) is the built in sum() function vs computing the sum with loops
* Use matplotlib to display a bar chart that shows the 2 time values. If one value is significantly greater than the other, use the log display feature of matplotlib to get a better visualization of the difference.

## 2. CSV Data Set Processing

Tip: review the section in the Deitel book section 9.12.3 Reading the Titanic Disaster Dataset

* Load the Titanic CSV dataset into a pandas DataFrame.
* Determine the number of individuals who died vs. those that survived
* Plot these values with a Bar Chart using pyplot
* Determine the average age of survivors vs. those who died
* Plot these values with a Bar Chart using pyplot

## Debugging

Rollo wrote the following program and needs your help

def foo(n):  
 sum = 0  
 for idx in range(n):  
 sum = sum + 0.01 \* idx  
 return sum  
  
sum = foo(2000)  
  
print(sum)

For some reason, Rollo want to use a debugger to set a conditional breakpoint when the index variable idx has a value of 99.

Use a conditional debugger to stop the debugger when idx has as value of 99 before idx gets added to the sum.

What is the value of the sum before idx (99) gets added? \_\_\_\_

Using the debugger STEP function to step thru the loop, what is the value of sum AFTER 99 gets added to the sum? \_\_\_\_\_-