# CIS 343 – Structure of Programming Languages
## Winter 2016, 3/22/2016

# Programming Assignment #4
# Matrix Operations in Ruby
## Due Date: Tuesday, April 12, 2016

### Project Goals

- Implement classes in Ruby
- Implement operator overloading
- Implement closures
- Custom exceptions and exception handling
- Unit testing

### Description

A matrix is a collection of numeric values arranged in rows. In a given matrix, each row has the same number of columns. In this project, you will implement a class called `Matrix` in Ruby language. **You are not allowed to use or refer to the `Matrix` class that is already part of the Ruby API in your implementation of the Matrix class**.

You are provided with two files – `Matrix.rb` and `MatrixTest.rb`. Your task is to complete the following methods in the `Matrix` class. You are not allowed to make changes to signature of the methods in the `Matrix` class. Remember, the unit tests in the `MatrixTest` class assume the method signatures as specified in the supplied `Matrix` class.

- `initialize()`
- `get()`
- `set()`
- `add()`
- `subtract()`
- `scalarmult()`
- `multiply()`
- `transpose()`
- `identity()`
- `fill()`
- `clone()`
- `==()`
- `each()`
- `to_s()`

The `MatrixTest` class contains unit tests for testing the functionality of the `Matrix` class. Your goal is to implement the `Matrix` class to get all the unit tests in the `MatrixTest` class to pass. You **MUST NOT MODIFY** the `MatrixTest` file.

## Executing Ruby Programs on EOS

To run the `main()` method in the Matrix class during your development, uncomment the call to main and do the following on EOS:

```
$ ruby ./Matrix.rb
```

To run the unit tests for Matrix class, do the following on EOS. **Remember, I will use only unit tests to grade your submission**:

```
$ ruby ./MatrixTest.rb
```

## Deliverables

1. **Upload only `Matrix.rb` file on Blackboard by midnight on due date.**
2. **I will use the submission date/time on Blackboard as your official submission date/time.**
3. **It is your responsibility to make sure the submission on Blackboard went through successfully.**
4. **I will compile, run, and test your program on EOS when grading.**
5. **Late penalty (10% per day) applies after due date.**