COSC346 Assignment 1 Matrix/Vector Library and test suite for Matrix/Vector Library

Name: Ashley Midgley

ID: 7776496

Name: Kahurangi Cassidy

ID: 6118539

**Report:**

The Matrix Vector Library consists of the following classes: Complex.swift, Fraction.swift, Interface.swift, Matrix.swift, MatrixTest.swift, TestMatrixVector.swift, Vector.swift, VectorTest.swift and main.swift.

The Complex class allows the creation of a complex number, which has a real and imaginary part. The class provides methods to be able to perform arithmetic on the complex numbers as well as using operators and scalars. The same goes for the Fraction class, which allows for the creation of fraction objects and then you are able to perform arithmetic using fractions.

Interface defines all of the protocols for the matrix and vector classes. These allow for the use of operators on matrices and vectors. It also allows for matrices to be of Int, Double, Float, Fraction and Complex types. The BasicMatrix allows access rows and columns of the matrix as well as creating the transpose of the matrix. This protocol also lets you create a copy of the matrix. The basic vector protocol allows you to get the size of the vector, the dot product of the vector, allows the ability to change each index of the vector and also allows for the creation of a copy of a vector. The MatrixArithmetic is the protocol for using operators on the matrices to perform arithmetic on them, this is the same for VectorArithmetic and also means that the vector arithmetic can be inherited from the Matrix as a Vector can be a matrix and then can use the matrix arithmetic too. The arithmetic also allows for the use of scalars instead of just one vector plus another it can be one vector plus three (which adds three to each index of the vector). The MatrixToVector protocol allows for a matrix to be converted to a vector. This happens by choosing a matrix’s row or column to become a new vector. VectorToMatrix converts a vector to a matrix; this allows a vector to be able to inherit all the properties of a Matrix (as seen by the Matrix arithmetic being used on a vector that is converted to a matrix).

The Matrix class allows for the creation of a matrix of any size by definition of its number of rows and columns as well as being of Int, Double, Float, Fraction or Complex types. It allows for the use of the protocols defined in Interface.swift so a matrix can have a transpose, copy, be converted to a vector, can have its indexes changed, and it can also have its contents converted to a String. This allows us to check results for the arithmetic performed on a Matrix. The matrix also has methods that allow for arithmetic on other matrices provided their sizes are compatible. The arithmetic can come from another matrix of the same type or from a scalar of the same type. The Vector class is much the same, you are able to define a vectors size and type as well as get its dot product, create a copy and convert a vector to a Matrix. The arithmetic for vectors is also implemented as to perform arithmetic on vectors with other vectors or scalars (by the use of methods or operators) where the methods are inherited from the Matrix class as mentioned before. This means that any arithmetic done with a matrix can be done with a vector.

The TestMatrixVector class is split into two classes. So the TestmatrixVector class as two 2D Boolean arrays for the testing in matrices and vectors. This class sets up the test suite in order to run the tests on the matrices and vectors and records their results, which are displayed in the main. MatrixTest and VectorTest are very similar. All the tests use Int, Double, Fraction and Complex types for each different test. The tests cover all the methods that can be used on a matrix or vector including arithmetic with the same types of matrices/vectors scalars with the use of operators. Also, each test class creates matrices/vectors from each other, for example: a vector can be crated by defining its size, or by using matrixview on a row or column of a matrix. And a matrix can be created by using the vectorview on a vector or by defining a matrix by size. In order to know that the results for the test are correct, the use of a string array is used. This is why the matrix and vector classes have a method for converting their contents to a string and then into a string array. This allows the resulting arithmetic to be checked against the correct contents of the matrix/vector. Once the string arrays are compared they a method will fill the 2D Boolean array with either a true or false, false meaning that the test failed and true meaning that the test passed. As mentioned before these results are displayed in the main. The test classes also have a method that runs the test as every test is put in its own function. This makes it easier to test single cases without having to run every other test every time the program is run.