**GUI Calculator**

Sebastian Hall

CSC-18B-42514

June 5, 2019

**a. GUI design/implementation**

The calculator implements two containers to arrange the layout. A Vbox as the root container to vertically align the calculators controls (result area, input area, buttons) and a nested gridpane to align the calculators buttons similarly to what one might expect to see on a real calculator. The application also uses javafx css properties to style the size of buttons and a custom css font included in the style sheet to give the display a more traditional calculator feel.

**b. Advanced String Usage**

The input area of the calculator uses a stringbuilder when when inserting values or modifying values from any part of the string as that method uses the substring method to split the string apart and insert it in the desired position which would be less efficient using strings. When the input string is parsed for use in the shunting yard algorithm regular expressions are used to separate the string into a queue of separate tokens to evaluate. The regular expressions search for a digit that possibly has a decimal followed one or more other digits at the very beginning of the string or a mathematical operator. The input stringbuilder then has the matching part removed from the beginning and added to the token queue using the stringbuilder delete method.

**c. Files / Streams**

The calculator uses ObjectOutputStream initialized with a ByteArrayOutputStream to return a saved results object as a byte array. It also conversely uses ObjectInputStream initialized with a ByteArrayInputStream to read in a byte array and cast it as a result object. All of those streams are used to prepare the result object for the database.

**d. Object Serialization**

The saved results object is serialized and deserialized using the ObjectInput/OutputStreams’ and ByteArrayInput/OutputStream’s as listed above in the section on streams.

**e. Databases**

The user of the calculator can save their results along with a name of their result in a database to use later. This is intended to be used for frequently used results such as mathematical constants like PI, e, or g or the results of your own calculations that you would like to use in later calculations. The result is turned into a Result object and then serialized into a byte array and then stored into a database with a single table that has a field that represents the unique id and a blob field that holds the binary representation of the object. There should be a way to delete saved results from the database but there isn’t because I am lazy.

**Special Instructions**

Just start the java database in netbeans in order to use the save results feature. If it can’t connect to the database ensure that the database location (right click java DB, properties section) is set to the root folder of the calculator project where the database is located.