**Reflection: 2048**

Rahul Berry

CID: 01217418

Dr. Max Cattafi

09.12.16

(e.g. which functions you defined and used and why you chose that structure for the program).

This program relies on one-dimensional vectors and nested for-loops to recreate the 2048 game.

The first version of my program aimed to determine the derivative of polynomials with up to four terms, as I hardcoded strings to differentiate each term individually. Further, it was designed to determine the value of the derivative of the function at a particular point as the user could select a value for the *x* variable However, this program was inefficient as it limited the number of terms that could be differentiated.. While my initial aim was to determine the actual value of the derivative at *x* of polynomials with up to four terms, this aim has since evolved to determining an expression for the derivative of a polynomial with *n* terms. This allowed me to focus more on the application of classes, vectors and loops to achieve the goal.

In order to complete this program, I had to learn about classes and their implementation in c++. Classes are used to define expressions and mathematical operations. For example, ‘ **class** AddExpr { ’ is used to define an addition operation. While attempting to create various classes and functions, I encountered several issues. For example, the ‘ **toString( ) {** ’ and ‘ **differentiate( ) {** ’ functions in each class were not functioning correctly, as they could not be redefined in each class. Therefore, these functions were converted to **virtual** functions, which enabled them to be redefined in each class. In addition to classes, which I learned to create this program, vectors, pointers and for-loops, all of which were taught in the autumn term, also play a vital role. Without using vectors and loops, I would have been unable to achieve the key goal of allowing the user to differentiate any number of terms.

The program is successful in achieving its key aim of determining any number of derivatives of a polynomial of degree *n*. However, the program is unable to simplify mathematical operations due to the fact that all outputs have been converted into strings using the **‘ toString()’** function. Hence, any operations, such as ‘ 5 – 1*’* will not be simplified to 4. In an attempt to solve this issue, I explored the ideaof creating a ‘ **simplify() ’** function in each class. However, due to the fact that each output is converted into a string, I was unable to successfully create a function to simplify the outputs.

While the program is successfully able to determine the derivative of polynomials, it can be extended by allowing the user to determine the derivatives for various functions. For example, by creating a ‘ **class** SinExpr {‘ the user will be able to determine the derivative of sine functions. In this case, within the class, the function ‘**differentiate**( ) {’ will return negative cosine expression. However, when adding new classes to allow for the differentiation of other functions, main.cpp must also be changed. In the submitted program the main will only return the derivative of a polynomial. This program can be extended in a number of ways, as there are many types of functions that can be differentiated, from all trigonometric functions to logarithmic and exponential functions.

By writing this program I have been able to understand the complex process behind creating websites such as Wolframalpha. It is fascinating to learn how mathematics can be embedded within programs, making them far more efficient than manual processes.