Project 1 - Team1 Documentation

## Overview

Team 1 (ie The Wookie Workgroup) created a command line tool that adds polynomials. The group consists of Daniel Mitchel, Joshua Neustrom, and Wang Chen.

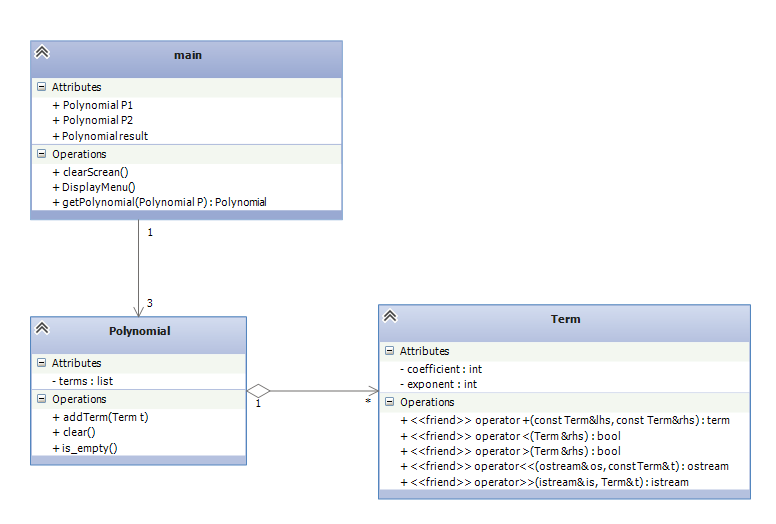
The report gives an overview of our solution including the following

1. Assumptions
2. UML Class Diagram
3. Github Project Link
4. Algorithm
5. Error Handling
6. Known Issues
7. Efficiency Analysis of Algorithms
8. References

## Assumptions

1. Only X variables used (can accept x but is changed to upper case)
2. No history stored past the last polynomial entered
3. Polynomials are sorted and simplified
4. User enters two polynomials before adding
5. If get bad input, ask user for new polynomial
6. Coef is assumed to be 1 if not entered
7. ^ entered before exponent greater than 1
8. Wookies rule

## UML Class Diagram



## Github Site

<https://github.com/WookieWorkgroup/Project1>

## Algorithm

1. User interface – continuous loop that displays a command line menu
2. Option 1 - Ask for polynomial #1
   1. Clear old polynomial
   2. Store line
   3. Add a + in front of – symbols
   4. Parse Terms into separate strings using + symbol
   5. Read in Term String
      1. Discard +
      2. Read Coefficient
      3. Discard Variable
      4. Read Exponent
      5. Add term to the polynomial (list of terms)
   6. Combine similar terms
   7. Sort the list
3. Option 2 - Ask for polynomial #2 (same process as #2)
4. Option 3 – Add polynomials
   1. Go through the ordered list
   2. Find similar terms (by exponent) and add coefs
   3. Add remaining terms
   4. Sort
5. Option 4 - Display polynomial #1
6. Option 5 – Display polynomial #2
7. Option 6 – Display result
8. Option 7 – Clear all polynomials
9. Option 8 – Exit - Done and Done
10. Other inputs – prompt user again

## Error Handling

1. Bad Input – Use try and catch to discard polynomial with improper formatting
2. Variables – Program accepts X and changes lower case x to a capital. Other variables result in an error (see #1)
3. Lack of coefficient or exponent – assumes coef is 1. Exponent is assumed to be one if X is in term and zero otherwise
4. Improper menu option entered – prompt user for input again

## Known Issues

1. Terms zeroed by addition still appear in polynomial with coef of zero
2. Excessive filtering of user input – non X variables can be incorrectly converted to X terms (example is 3Y -> 3X)

## Efficiency of Algorithms

1. Sort of polynomials – O(n^2) (built in sort function of the list library)
2. Add polynomials – O(n^2) – sort called at the end of an addition
3. Input polynomial – O(n^2) due to built-in list sort function being called
4. Display polynomial – O(n)

## References

1. Built in sort function - http://www.cplusplus.com/reference/list/list/sort/
2. Method for eliminating user input that causes overflow - http://stackoverflow.com/questions/3826281/how-do-i-make-a-c-program-that-filter-out-non-integers
3. The Force