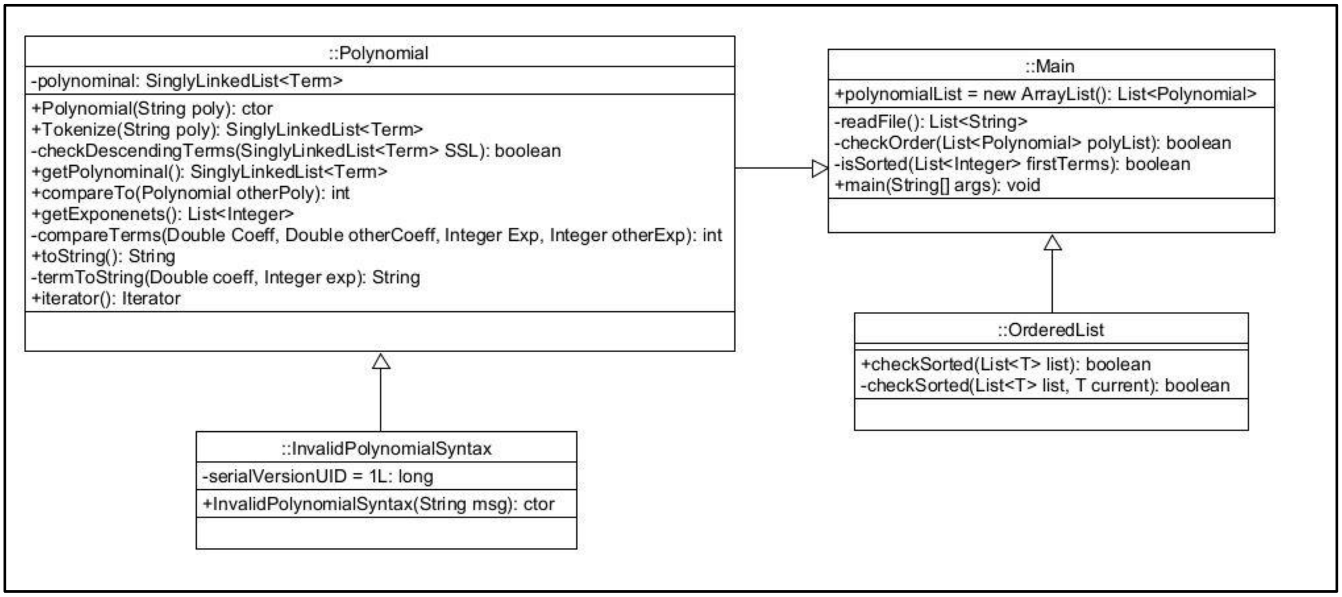
Allen Taylor
CMIS 350 6382
Project 2
2/4/2022

UML Diagram for Project 2


::Polynomial
-polynominal: SinglyLinkedList<Term>
+Polynomial(String poly): ctor
+Tokenize(String poly): SinglyLinkedList<Term>
-checkDescendingTerms(SinglyLinkedList<Term> SSL): boolean
+getPolynominal(): SinglyLinkedList<Term>
+compareTo(Polynomial otherPoly): int
+getExponenets(): List<Integer>
-compareTerms(Double Coeff, Double otherCoeff, Integer Exp, Integer otherExp): int
+toString(): String
-termToString(Double coeff, Integer exp): String
+iterator(): Iterator
::Main
+polynomialList = new ArrayList(): List<Polynomial>
-readFile(): List<String>
-checkOrder(List<Polynomial> polyList): boolean
-isSorted(List<Integer> firstTerms): boolean
+main(String[] args): void
::OrderedList
+checkSorted(List<T> list): boolean
-checkSorted(List<T> list, T current): boolean
::InvalidPolynomialSyntax
-serialVersionUID = 1L: long
+InvalidPolynomialSyntax(String msg): ctor

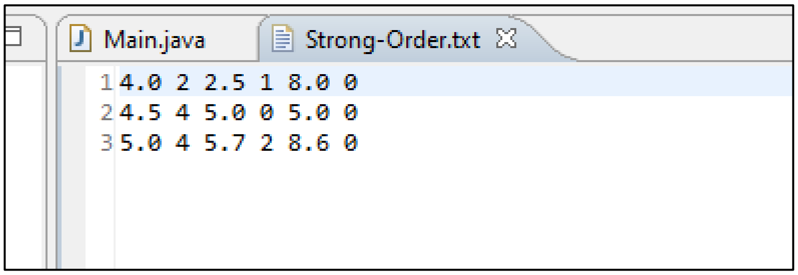
Test Case 1

Requirements:

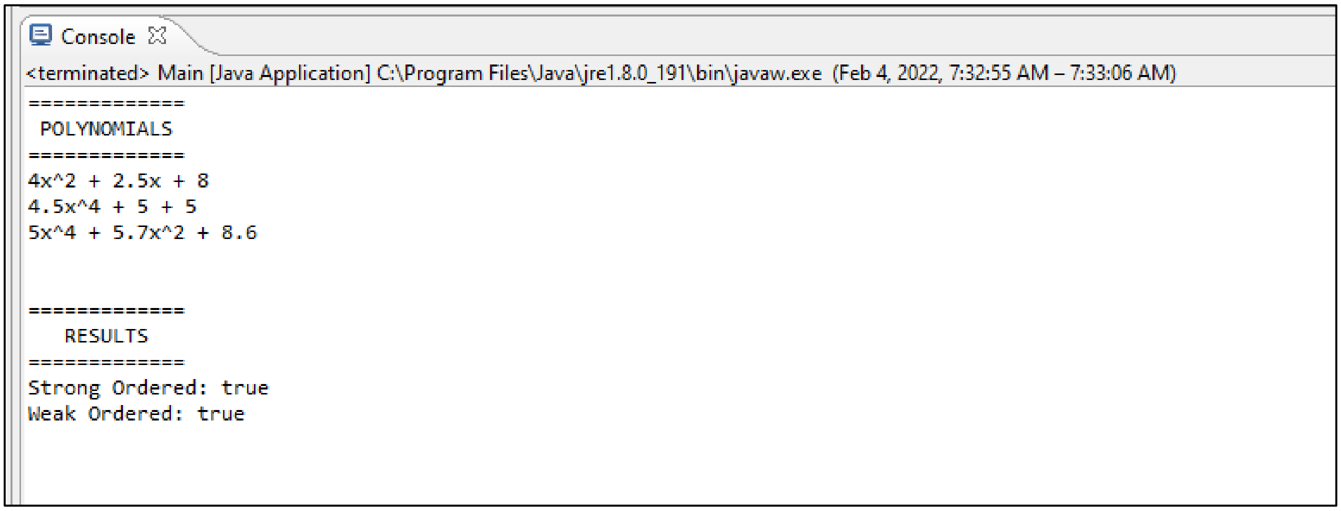
1. Test cases include a file in both strong and weak sorted order.
2. Test cases include a polynomial with exponents of 0, 1 and 2 or more.

Input:	4.0 2 2.5 1 8.0 0
	4.5 4 5.0 0 5.0 0
	5.0 4 5.7 2 8.6 0

Expected Output:	Strong Ordered: true Weak Ordered: true
Actual Output:	Strong Ordered: true Weak Ordered: true



```
4.0 2 2.5 1 8.0 0
4.5 4 5.0 0 5.0 0
5.0 4 5.7 2 8.6 0
```



<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (Feb 4, 2022, 7:32:55 AM – 7:33:06 AM)

```
=============
 POLYNOMIALS
=============
4x^2 + 2.5x + 8
4.5x^4 + 5 + 5
5x^4 + 5.7x^2 + 8.6


=============
   RESULTS
=============
Strong Ordered: true
Weak Ordered: true
```

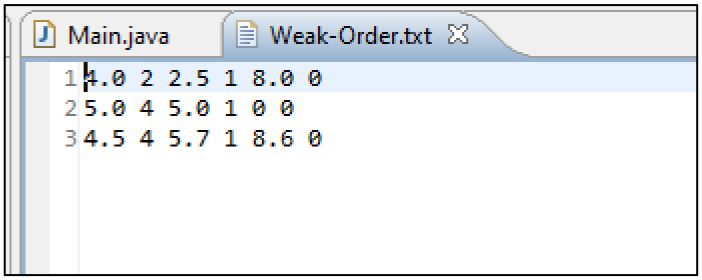
Test Case 2

Requirement:

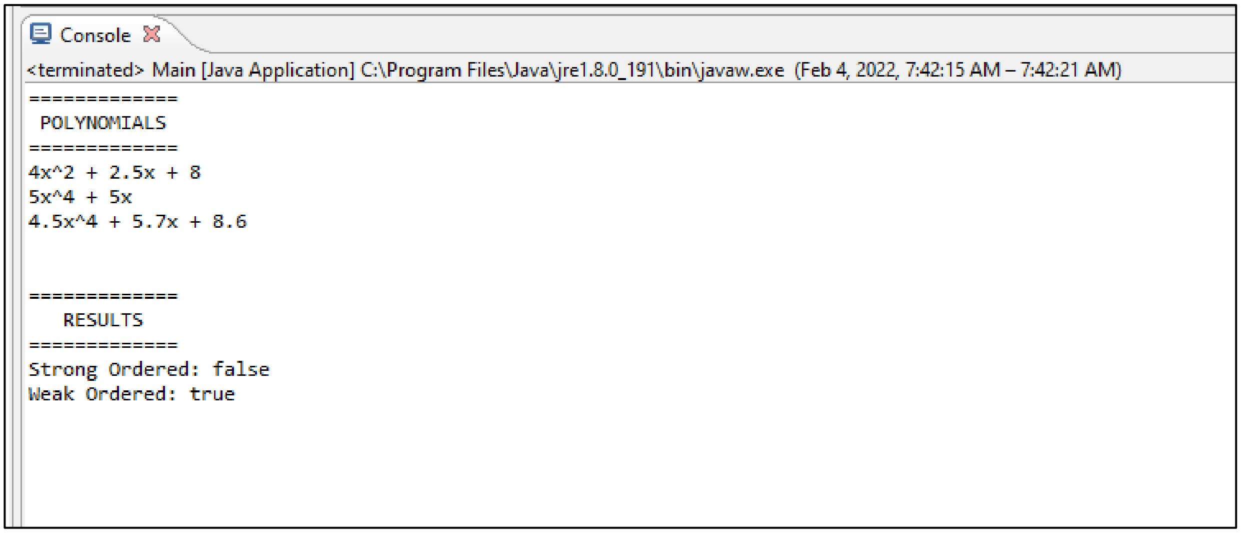
1. Test cases include a file in weak but not strong sorted order.
2. Test cases include a polynomial with exponents of 0, 1 and 2 or more.

Input:	4.0 2 2.5 1 8.0 0
	5.0 4 5.0 1 0 0
	4.5 4 5.7 1 8.6 0

Expected Output:	Strong Ordered: false Weak Ordered: true
Actual Output:	Strong Ordered: false Weak Ordered: true



```
4.0 2 2.5 1 8.0 0
5.0 4 5.0 1 0 0
4.5 4 5.7 1 8.6 0
```



<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (Feb 4, 2022, 7:42:15 AM – 7:42:21 AM)

```
=============
 POLYNOMIALS
=============
4x^2 + 2.5x + 8
5x^4 + 5x
4.5x^4 + 5.7x + 8.6


=============
   RESULTS
=============
Strong Ordered: false
Weak Ordered: true
```

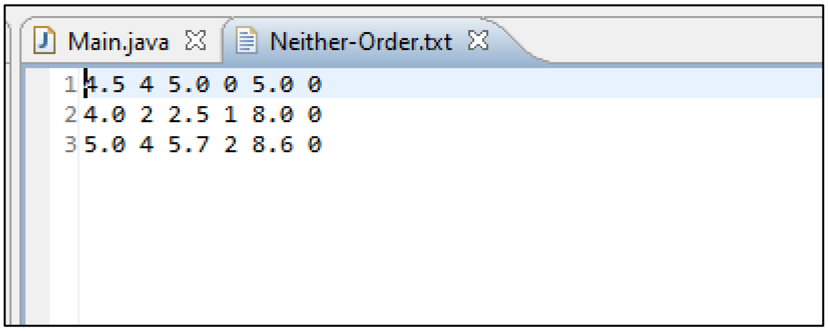
Test Case 3

Requirement:

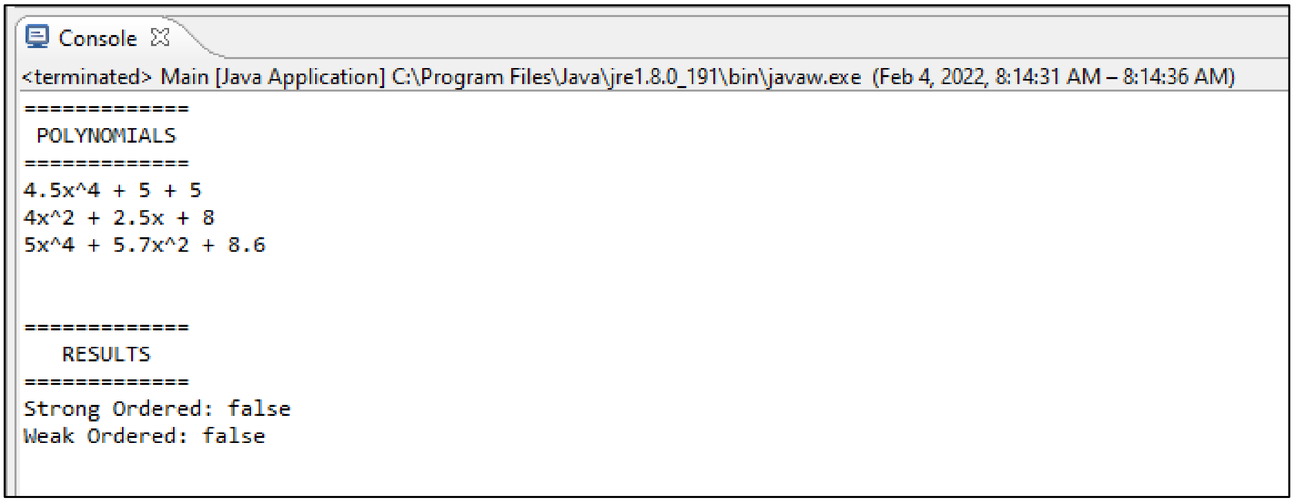
Test cases include a file in neither strong nor weak sorted order

Input:	4.5 4 5.0 0 5.0 0
	4.0 2 2.5 1 8.0 0
	5.0 4 5.7 2 8.6 0

Expected Output:	Strong Ordered: false Weak Ordered: false
Actual Output:	Strong Ordered: false Weak Ordered: false



```
4.5 4 5.0 0 5.0 0
4.0 2 2.5 1 8.0 0
5.0 4 5.7 2 8.6 0
```



<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (Feb 4, 2022, 8:14:31 AM – 8:14:36 AM)

```
=============
 POLYNOMIALS
=============
4.5x^4 + 5 + 5
4x^2 + 2.5x + 8
5x^4 + 5.7x^2 + 8.6


=============
   RESULTS
=============
Strong Ordered: false
Weak Ordered: false
```

Test Case 4

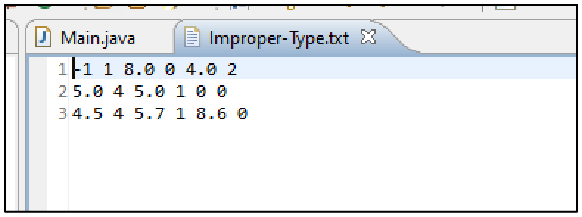
Requirement:

InvalidPolynomialSyntax thrown by the constructor of the Polynomial class when:

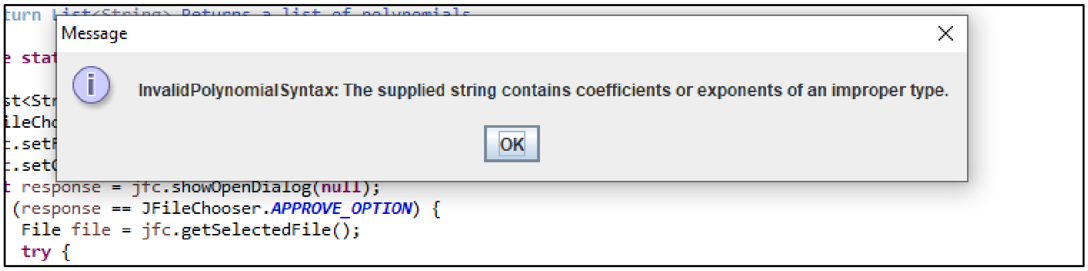
1. The supplied string contains coefficients or exponents of an improper type.
2. Exponents fail to be listed in strictly descending order.

Input:	-1 1 8.0 0 4.0 2
	5.0 4 5.0 1 0 0
	4.5 4 5.7 1 8.6 0
	
	2.5 1 8.0 0 4.0 2
	5.0 4 5.0 1 0 0
	4.5 4 5.7 1 8.6 0

Expected Output:	JOptionPanes display error messages.
Actual Output:	JOptionPanes display error messages.

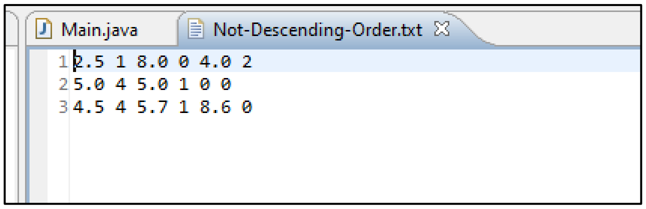


```
-1 1 8.0 0 4.0 2
5.0 4 5.0 1 0 0
4.5 4 5.7 1 8.6 0
```

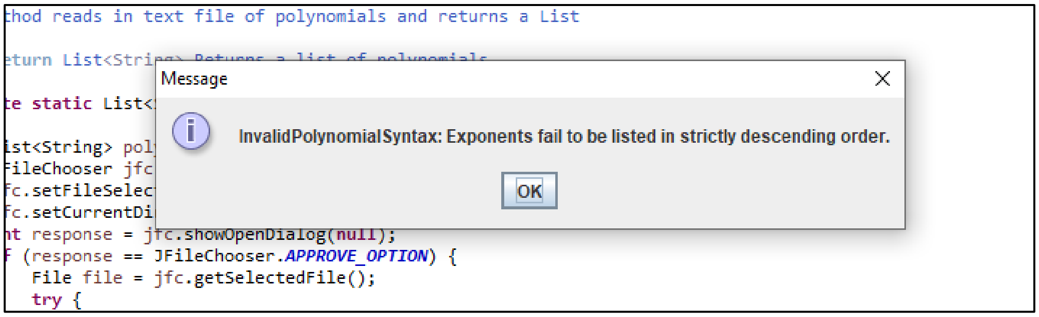


InvalidPolynomialSyntax: The supplied string contains coefficients or exponents of an improper type.

```
turn List<String> Returns a list of polynomials
e stat
st<Str
ileCho
:.setF
:.setC
t response = jfc.showOpenDialog(null);
 (response == JFileChooser.APPROVE_OPTION) {
  File file = jfc.getSelectedFile();
  try {
```



```
2.5 1 8.0 0 4.0 2
5.0 4 5.0 1 0 0
4.5 4 5.7 1 8.6 0
```



InvalidPolynomialSyntax: Exponents fail to be listed in strictly descending order.

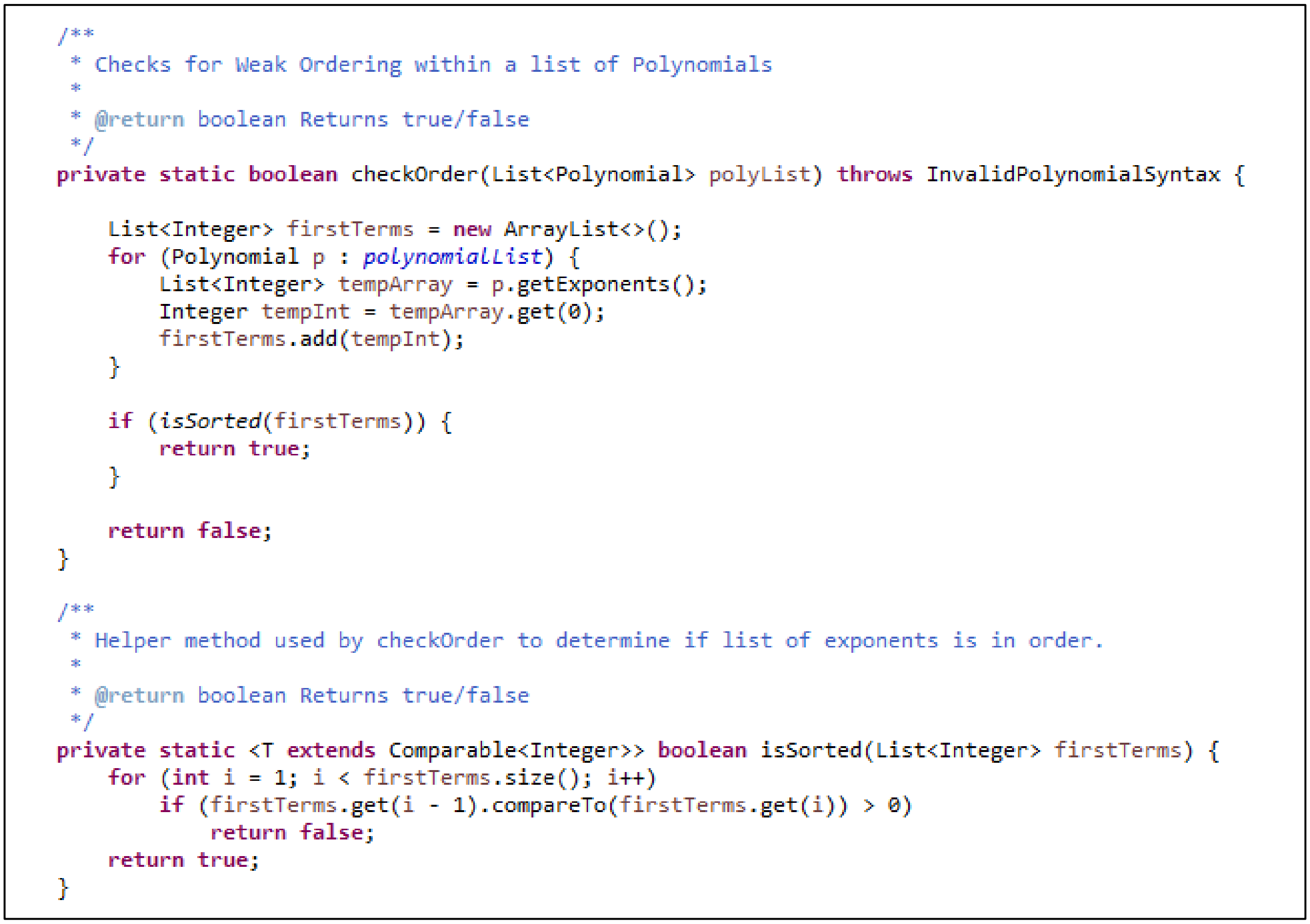
```
thod reads in text file of polynomials and returns a List

eturn List<String> Returns a list of polynomials

te static List<

ist<String> pol
FileChooser jfc
fc.setFileSelec
fc.setCurrentDi
nt response = jfc.showOpenDialog(null);
f (response == JFileChooser.APPROVE_OPTION) {
  File file = jfc.getSelectedFile();
  try {
```

Lessons Learned



```
/**
 * Checks for Weak Ordering within a list of Polynomials
 *
 * @return boolean Returns true/false
 */
private static boolean checkOrder(List<Polynomial> polyList) throws InvalidPolynomialSyntax {

    List<Integer> firstTerms = new ArrayList<>();
    for (Polynomial p : polynomialList) {
        List<Integer> tempArray = p.getExponents();
        Integer tempInt = tempArray.get(0);
        firstTerms.add(tempInt);
    }

    if (isSorted(firstTerms)) {
        return true;
    }

    return false;
}

/**
 * Helper method used by checkOrder to determine if list of exponents is in order.
 *
 * @return boolean Returns true/false
 */
private static <T extends Comparable<Integer>> boolean isSorted(List<Integer> firstTerms) {
    for (int i = 1; i < firstTerms.size(); i++)
        if (firstTerms.get(i - 1).compareTo(firstTerms.get(i)) > 0)
            return false;
    return true;
}
```

One of the hardest parts of this assignment was using the Comparable Interface. I finally found a solution in the provides coding examples in the Week 1 Generics. This provided me with the isSorted method.

To determine the Weak Order of a list of Polynomials I did the following:

1. Read all exponents from the Polynomial.
2. Take only the exponent from the FIRST term.
3. Do this for all Polynomials in the list and add them to a List<Integer>.
4. Lastly, use the isSorted to check if they are in Weak Order.