**Assignment 2 – ITC 322**

**Discussion:**

This Program took a while longer to create than expected. The node and its fields were hard to get around. I understood them on paper but not in java implementation. The forum explanations helped me with extending linked list. I ending up making a separate class for the fields called node, not sure if that was the correct way to do so.

The constructors were also a bit strange. How would one tell the difference between a string of file name and string to be made into polynomial? I eventually settled for semi constructor, a method that still uses the other constructor but with extra handles for reading the text file. I guessing this is allowed due to the forums response of creating the object and reading the file being in different places been given approval.

The actual constructor is very complicated, especially with the extra features demanded. These being the double digit and the potentially more digits, so I have coded for the extra. This has led to some convoluted code, specifically when the already converted to integer is converted back to string, concatenate the strings and back to integer. This isn’t too bad I suppose however when many digits are used, this would make a lot of unnecessary computing. Unfortunately the only way I see to avoid this is to make the code extra complicated so this converting isn’t done until all the digits are collected. It is good we are not tested on the speed of creation of these polynomials. The other feature being the negatives, so adding extra Boolean checks were annoying.

The adding method seemed straight forward. The only issue how to actually do it. My response to this was that since the polynomial must be sorted at all times, why not just combine them? It does look like undocumented feature but without overcomplicating things with duplicate code, making a private function seems right.

The divertive function was very easy; I only had to brush up on the rules of multiplying powers. One gripe I had though was that there was no particular easy way to create a polynomial on the fly and add to it later. My already complete constructor makes empty a single link if I pass an empty string, so instead of changing the constructor which was complicated enough already, I made separate constructor. Hope I don’t lose points on that.

For the driver program, without adding extra choices, I decided to make some default choices for the user with the adding, get value and product. I justify this as the program is supposed to be, more or less, a proof of concept and should test all methods and functions, which it does. However you can see that it could be easily changed to ask for additional input.

You will see at the end of the source code is a sorting algorithm. This took a while to make. It was primarily because I was trying to make a custom implementation of sorting similar to insertion. When this got done I then decided to change it to the previous assignments way, using comparable, which makes cleaner code. However this presented its own problems, which was that I was not competent using two separate classes with this interface. A tried implementing it directly in the linked list which didn’t work, so far as my efforts went anyway. Doing it only on separate class for the node did however, so that how I did it at last.

**How to run (command line):**

The java programs are designed to be run with current directory set as the folder of the program in the command prompt, complied with javac and then run with java. JDK must also be installed with it’s directory set in the environmental variables. If javac doesn’t work, a temporary workaround is to use code similar to this:  
set PATH=%PATH%;C:\Program Files\Java\jdk1.7.0\_55\bin

**Eg:  
Programs location(source):**C:\Dropbox\ITC322\_assign2\src\Polynomials.java **Programs location(complied):**C:\Dropbox\ITC322\_assign2\bin\Polynomials.class  
C:\Dropbox\ITC322\_assign2\bin\Node.class

**Command Prompt input:**🡪cd “C:\Dropbox\ITC322\_assign2\src\” (compiling optional)  
🡪javac Polynomials.java (compiling optional)  
🡪cd “C:\Dropbox\ITC322\_assign2\bin\” (run if not compiling)  
🡪java “Polynomials”

**How to run (eclipse):**

1. Open **Eclipse** IDE for Java EE Developers
2. Open the “**file**” menu
3. Click “**Import**”
4. Expand “**General**”
5. Double click “**Existing projects into workspace**”
6. Browse for the **root directory** (the one with .project files etc.)
7. Import “**ITC322\_assign2**”
8. **Tick** the box
9. Click **Finish**
10. **Select the project** on the left side
11. Click **Run** (top toolbar) > click **Run**
12. Choose an options presented by the program to **select polynomial**

**Java Documents:**

The java docs were written into the code and generated using the Java Development Toolkit’s JavaDoc.exe. The generated html files reside in “**ITC322\_assign2\doc**” and can be opened by accessing “**index.html**” with your preferred web browser.

A screenshot of the program output is located at the bottom of this document.

