UMGC – CMSC 315
Project 2

Indications and recommendations regarding assumptions, constraints and defining the Comparator

1. You may consider the simplification assumption that the polynomial coefficients are positive decimal values with one only decimal value.

2. You may also assume that the exponents are positive integer values (including zero).

3. The instructions define **two types of strictly ascending order** of polynomials: **strong order** and **weak order**.

4. Test input files and their content should be created by the students using a simple text editor such as Notepad.

5. Regarding defining a custom sorting comparator by implementing the Comparator interface, either of the following three techniques will be considered:

5.1 Defining a custom sorting comparator by using a lambda expression:

```
Comparator<Polynomial> customSortingComparator =
     (Polynomial p1, Polynomial p2) -> { // ... implementation code };
```

5.2 Defining a custom sorting comparator class that implements the interface Comparator:

```
public class CustomSortingComparator implements Comparator<Polynomial> {
    @Override
    public int compare(Polynomial p1, Polynomial p2) {
        // ... implementation code
    }
}
```

5.3 Using an anonymous custom sorting comparator class to generate a Comparator object:

```
Comparator<Polynomial> customSortingComparator =
    new Comparator(<Polynomial>)() {
           @Override
           public int compare(Polynomial p1, Polynomial p2) {
                // ... implementation code
           }
    };
```

Note. For Web references about lambda expressions please check the attached file “Lambda Expressions – Web references”.