Assignment 4: Higher-order Functions in Java

1 Implementation

The goal of this assignment is to implement a simple function in Java that would only take a few lines in functional languages.

1. You are to implement in Java class Reduction<T> that provides a method reduce. The class is to be constructed with a reduction operator (a binary function of the form $T \times T \rightarrow T$, defined in an interface Operator<T>) and an a neutral element (of type $T$). The reduce method is given a List of elements of type $T$ and reduces\(^1\) the list, returning the resulting value (of type $T$).

2. Based on this generic reduction function, you must then implement reduction operators for int addition (Add) and multiplication (Mul).

All access modifiers for all classes, interfaces and methods must be set to be public.

2 Submission

You must submit the implementations to your subversion repository to the directory comp3351/f2007/$USER/P4/.

The files must be called

- src/edu/du/cs/comp3351/p4/Reduction.java
- src/edu/du/cs/comp3351/p4/Add.java
- src/edu/du/cs/comp3351/p4/Mul.java
- src/edu/du/cs/comp3351/p4/Operator.java

You must check that the submitted code compiles by invoking make. Verify that the output of your program matches the expected output using your own testcases.

\(^1\)The meaning of reduce is defined in “Why Functional Programming Matters”.

1