**COMP 3002 Winter 2025 Assignment #1b**

**Building the class Relation**

**Due: Sun Midnight Jan 12.**

**Basic goals**: To implement the class Relation described in the notes. The descriptions below is for the Smalltalk version.

**Be simple if possible…**

Even though the class is known to contain triples, it’s not obvious how to store the triples. You could store them in an ordered collection or a set of 3-element arrays. Hint: Would there be any use to looking at the implementation of collection method partitionsUsing:?

One nice way to work is to create example class methods that can be used to show that the class works. For example, run “Relation example1” and “Relation example2”. Provide other example methods for showing off other aspects of your class.

example1

"Relation example1"

| relation |

"First, build a relation (don’t store duplicates)."

relation := Relation from: #((2 < 3) (1 = 1) (3 > 1) (2 < 4)(1 < 5) (5 < 6) (2 < 5) (1 = 1)).

"Second, show that the entire relation can be printed…"  
 Transcript cr; << 'Let relation = '; << relation.

"Third, show that the 3-parameter do: works…"  
 Transcript cr; << 'One triple per line, version1 of relation is'.

relation do: [:a :b :c |Transcript cr; << ' ('; << a; space; << b; space; << c; << ')']

"Fourth, show that the 1-parameter do: works…"

relation do: [:triple |Transcript cr; << triple]

example2

"Relation example2"

| relation |

"First, build a relation."

relation := Relation from: #((2 < 3) (1 = 1) (3 > 1)  
 (2 < 4)(1 < 5) (5 < 6) (2 < 5) (1 < 6) (1 < 7) )]).

"Second, print the relation"

Transcript cr; << 'Let relation = '; << relation.

"Third, show that from:relationsDo: works…"  
 Transcript cr; << 'Starting from {1 2 3},'.

relation from: #(1 2 3) relationsDo: [:relationship :subrelation |   
 Transcript cr; << 'The class of the subrelation is '; << subrelation class.   
 Transcript cr; << 'There is a relationship '; << relationship;   
 << ' with subrelation'; cr; tab.  
 subrelation do: [:triple |Transcript space; << triple]]

"Without debugging information and for different from sets."

#((1 2 3) (1 2) (2) ()) do: [:fromCollection |  
 Transcript cr; << 'Starting from {'.  
 fromCollection do: [:from | Transcript space; << from].

Transcript < ' }.

relation from: fromSet keysDo: [:relationship :subrelation | Transcript cr; << relationship; ' => '; << subrelation]].