

# COMP 3004 - SCAPES Assignment #2

Due: Tuesday, October 8, 2019 at 4:00 PM (afternoon)

Collaboration: This assignment must be completed **individually**

## Object and Dynamic Model Document (partial)

### Table of content:

1. Object model
2. Dynamic model

### Document content:

#### *Object model:*

- UML class diagram:
  - You must provide **one** UML class diagram that shows the **entity objects** involved in the *compiling and running a program* features of the SCAPES system
  - Every class must show attributes and basic operations, but do not show getters, setters, constructors or destructors; every attribute and operation must show visibility, and operation parameters must show their role type (in, out, inout)
  - All class associations must be shown, with directionality and multiplicity where applicable

#### NOTES:

- You **must** design your object model so that it can store the objects that represent the compilation output, as well as all the associations between those objects. The compilation phase will perform a *thorough syntax checking* of the SCAPL program, including verifying that all the variables and labels referenced in the instructions are actually declared in the program. The relationships between individual statements and the variables and labels that they reference **must** be represented in your object model, as you will be implementing this in your Deliverable #1.
- Because of the above, your object model **must** be able to store a program that is separated into different objects based on its syntax. Your entity objects **must** include the `Statement` (or `Instruction`) class, as well as the `Variable` and `Operand` classes, with the correct associations between them. You will note that the `Statement` class is involved in an inheritance hierarchy of the different kinds of instructions allowable in SCAPL, and **there is polymorphism in this group of classes!** Your class diagram must show all the related classes, it must demonstrate the polymorphism in the correct classes by showing the class operations that are polymorphic.

#### *Dynamic model:*

- UML sequence diagrams:
  - You must provide a UML sequence diagram for the *compiling a program* use case
  - Your sequence diagram must show the entity objects being created

## Grading

### Grading breakdown:

- *Object and Dynamic Model Document:*

Class diagram	50%
Sequence diagrams	50%

## **Format**

Documentation deliverables must be submitted as a **PDF document**. They must be typed and legible, and they must be **professional**, including a cover page, page numbers, as well as section numbers and names. All UML diagrams must be produced using a drawing tool, and not hand-drawn. Documents that do not conform to these specifications will not be graded.