1. Name, email:
2. Xtext Grammar:

(a)  Does your grammar support all of the example programs? If not what are the limitations?

(b)  How did you implement operator precedence and associativity?

(c)  How did you implement the syntax of variables (ID in rule Exp of the IF22 BNF), such that they can refer both to parameters of scenarios and local variables?

(d)  How did you implement the syntax of to statements, such that they refer to announcement/question/end/scenario? (first ID in rule Target of the IF22 BNF)

1. Scoping Rules

(a)  Did you implement scoping rules that allow variables to refer to variable definition and scenario parameters? If not, what are the limitations?

(b)  Did you implement scoping rules that allow announcement and question statements to call scenarios and reference the called scenario end statements? If not, what are the limitations?

(c)  Describe your implementation of any scoping rules included with your system.

1. Type Inference

(a)  What validation rule did you implement, if any?

(b)   Did you implement validation for the correct use of keywords such as this and Type keywords (number, text)? If not, what are the limitations? If yes, briefly describe your approach

(c) Did you implement validation with type inferencing? If yes, does it correctly check the types for all expressions (including input validation and conditions) in the provided IF22 examples? If not, what are the problems?

(d) If you implemented validation with type inferencing, briefly describe your approach.

1. Generator

(a)  Does your code generator correctly generate code for all of the ex- amples provided, and are you confident that it will also work for “similar” programs? If not, then what limitations are there?

(b)  Briefly describe how your code generator works.

1. Implementation: include your xtext grammar file and all implemented Xtend files (scoping, type inference, generator) verbatim.