Computer Science
Engineering School

Software
Engineering

Lab 13
Code Generation III

Francisco Ortín Soler

University of Oviedo

Survey

- Please fill in the survey at (2 minutes)
  https://encuestas.uniovi.es/calidad

Objective

- Generate code for
  - Function invocation

    (function definition was done in lab 10)
  - Return statement

Function Invocation

- Analyze input.txt and output.txt to identify the code to be generated for
  1. Function invocation
  2. Return statement

Questions

1. What code functions should be defined for **`Return`**?
2. What code functions should be defined for **`FunctionInvocation`**?

Question

- Define code generation for **`Return`**

execute[[Return: statement → exp]] = ?

Activity 1

- Write your **code templates** as multiline comments at the beginning of
  a. AddressCGVisitor.java for Address templates
  b. ValueCGVisitor.java for Value templates
  c. ExecuteCGVisitor.java for Execute templates
  - Be careful with the productions of the abstract grammar
  - Show the AG to the lecturer before its implementation (mandatory)

Activity 2

- You must have your code templates validated by the lecturer before the implementation
- **Implement** the code templates in your C-- compiler
  - At least test it with input.txt and big-input.txt files provided
  - **Always run the generated code with MAPL**
    1. Making sure the execution is the expected one
    2. No errors or warnings are shown

Lab Exams

- **VERY IMPORTANT NOTICE**

  For any lab exam, your implementation must compile the two examples given in Lab 13 and generate correct MAPL code

  You **must** bring the two files to the exam