COMP 304B – Object-Oriented Software Design
Assignment 5 – Visitor Pattern

Due date: Friday April 11, 2003 before 23:55

Practical information

- Team size == 2 (pair design) !
- Each team submits only *one full solution*. Use the `index.html` template provided on the assignments page. Use **exactly** this format to specify names and IDs of the team members. The other team member *must* submit a single `index.html` file containing only the coordinates of both team members. This will allow us to put in grades for both team members in WebCT. Beware: after the submission deadline there is no way of adding the other team member's `index.html` file and thus no way of entering a grade !
- Your submission must be in the form of a simple HTML file (`index.html`) with explicit references to *all submitted files* as well as inline inclusion of images. See the general assignments page for an `index.html` template.
- The submission medium is WebCT.

The assignment

In this assignment you will use UML *Class Diagrams*, Pseudocode and UML *Sequence Diagrams* to describe a simple design based on the *Visitor Pattern*.

The design uses the `Formula` class structure of the Spreadsheet design found in the solution of assignment 2.

Your design removes the `evaluate` methods from all classes and puts it in an appropriate `Evaluate` visitor. A `PrettyPrint` visitor is also required. It makes use of the `__str__` methods which are assumed to *not* recursively call their children (in case of operators, for example).

Your assignment solution should contain:

1. A Class Diagram depicting all relevant classes, their attributes and methods, as well as all relationships between the classes.

   Pseudocode (in sticky notes) must be provided for *all* important methods (for example, for `accept()`).

2. Pseudocode for a *Use Case* (scenario) in which `program`, and instance of the `Program` class has access to the tree corresponding to the formula `$A$12*2+10`. The `program` first evaluates and then prettyprints this formula.

3. A Sequence Diagram depicting the above Use Case.

The Class Diagram, Pseudocode, and Sequence Diagram must be *consistent*.

Add short explanations where necessary.