CSC 345 Lab – Abstract Syntax Tree

Overview

In this lab you will draw diagrams for abstract syntax trees (AST).

Setup and Assumptions

Assume the following abstract node classes exist:

- Expr (abstract).
- Stmt extends ASTBase (abstract).
- Id extends Expr. Use this for variable names.
- IntLiteral extends Expr. Use this for integer literals (constants).
- Sum extends Expr. Has two Expr children.
- Assign extends Stmt. Normal assignment. Has two children. The left side is an Id and right side is an Expr.
- Equals extends ASTBase. Use for equality comparison. Evaluates to true or false. Should have two Id children.
- LessThan extends ASTBase. Use for a less than comparison. Evaluates to true or false. Should have two Id children.
- StmtCollection extends ASTBase. Has multiple Stmt children.
- If extends Stmt. Should have Equals and StmtCollection as children.

Problem 1

Draw an AST diagram for a print abstract node class. You can only print ids.

Problem 2

Draw an AST diagram for a while abstract node class. Assume it only uses less-than comparisons.

Problem 3

Draw an AST diagram for an if-else abstract node class. Assume it only uses equals comparisons.

Problem 4

Draw an AST diagram for the following code:

```
While x < y
   print x
   x = x + 1
endWhile
```

Problem 5

Draw an AST diagram for the following code:

```
If (x  = y)
   y = 1
   While y < z
      print y
      y = y + 1
   endWhile
else
   print y
endIf
```

Problem 6

Draw an AST diagram for the following code:

```
If (x  = y)
  If (w = z)
     a = b + c + d
  else
    a = 1
  endif
else
  While y < z
     print y
     y = y + 1
  endWhile
endif
```