

Code Generation for Assignment 4

This table specifies the code to generate for Assignment 4. Any grammar elements that are crossed out will be left for Assignment 5—you can ignore them for now.

This specification is for the class itself—you may also need to include a package declaration and/or import statements.

AST Node	Code to add to StringBuilder (shown in red) _X_ means that visiting _X_ should result in code being added to represent X.
Program ::= Type IDENT NameDef* Block	<pre>public class _IDENT_ { public static _Type_ apply(_NameDef* _) _Block }</pre> <p>Note: parameters from _NameDef*_ are separated by commas</p>
Block ::= BlockElem*	{ _BlockElem* _ }
BlockElem ::= Declaration Statement	Note: Declarations and Statements are terminated with ;
NameDef ::= Type Dimension² IDENT	_Type_ _name_ Where _name_ is the Java name of the IDENT.
Type ::= image pixel int string void boolean	int String void boolean
Declaration ::= NameDef	_NameDef_
Declaration ::= NameDef Expr	_NameDef_ = _Expr_
Expr ::= ConditionalExpr	
BinaryExpr	
unaryOp Expr PostFixExpr	
StringLitExpr	_StringLitExpr_.getText
NumLitExpr	_NumLitExpr_.getText
IdentExpr ConstExpr	_IdentExpr_.getNameDef().getJavaName()
BooleanLitExpr	true or false
ExpandedPixelExpr	
ConditionalExpr ::= Expr _{GuardExpr} Expr _{TrueExpr} Expr _{FalseExpr}	(_ Expr _{GuardExpr} _ ? _ Expr _{TrueExpr} _ : _ Expr _{FalseExpr} _)
BinaryExpr ::= Expr _{leftExpr} op Expr _{rightExpr}	If Expr _{leftExpr} .type is string and op is EQ _ Expr _{leftExpr} _.equals(_ Expr _{rightExpr} _)

	<p>If op is EXP <code>((int)Math.round(Math.pow(_ Expr_{leftExpr} _ , _ Expr_{rightExpr} _))</code></p> <p>Otherwise <code>(_ Expr_{leftExpr} _ _op_ _ Expr_{rightExpr} _)</code></p>
UnaryExpr ::= op Expr	<code>(_op_ _Expr_)</code> Note: you do not need to handle width and height in this assignment
PostfixExpr ::= Expr PixelSelector ² ChannelSelector ²	
ChannelSelector ::= red green blue	
PixelSelector ::= Expr _{xExpr} Expr _{yExpr}	
ExpandedPixelExpr ::= Expr _{red} Expr _{green} Expr _{blue}	
Dimension ::= Expr _{width} Expr _{height}	
LValue ::= IDENT PixelSelector ² ChannelSelector ²	<code>_IdentExpr_.getNameDef().getJavaName()</code>
Statement ::=	
AssignmentStatement	
WriteStatement	
DoStatement	
IfStatement	
ReturnStatement	
StatementBlock	
AssignmentStatement ::= LValue Expr	<code>_LValue_ = _Expr_</code>
WriteStatement ::= Expr	<code>ConsoleIO.write(_Expr_)</code> Note: you will need to import <code>edu.ufl.cise.cop4020fa23.runtime.ConsoleIO</code>
DoStatement ::= GuardedBlock ⁺	
IfStatement ::= GuardedBlock ⁺	
GuardedBlock ::= Expr Block	
ReturnStatement ::= Expr	<code>return _Expr_</code>
StatementBlock ::= Block	<code>_Block_</code>