Group: Feynman Members: Xukun Qin, Dalton Stokke Note: Dalton completed Expr concrete classes and all abstract clases, Xukun completed Stmt, Decl, Stmts, and program concrete classes. UML was created on one computer via UMLet while trading off in a pair programming fashion. Node Note: Node is Super Class, Abstract class unparse is pure virtual and +unparse() requires unparse to be implemented in all sub-classes. Program Stmts Decl Stmt Expr exprExpr -_e1: Expr* +exprExpr (e:Expr*) emptyStmts variableStmt matrixExpr program lex1: std::string -_lex1: std::string -_lex1: std::string +emptyStmts ifExpr -_s1: Stmts* -_e1: Expr* -_e1: Expr* () -_e2: Expr* - e1: Expr* -_e2: Expr* +program -_e2: Expr* -_e3: Expr* (lex:std::string, +matrixExpr seqStmts -_e3: Expr* - boolean:int s:Stmts*) (lex:std::string, -_s1: Stmt* +ifExpr e1:Expr*,e2:Expr*) +variableStmt -_s2: Stmts* (e1:Expr*,e2:Expr* (lex:std::string, e3:Expr*) +seqStmts e1:Expr*) varExpr (s1:Stmt*, +variableStmt -_lex1: std::string s2:Stmts*) (lex:std::string, letExpr +varExpr e1:Expr*,e2:Expr*, (lex:std::string) -_s1: Stmts* e3:Expr*) -_e1: Expr* +letExpr intDecl constExpr stmtsStmt (s:Stmts*,e:Expr*) -_s1: Stmts* -_lex1: std::string -_lex1: std::string +intDecl +stmtsStmt +constExpr (lex:std::string) (s:Stmts*) (lex:std::string) notExpr -_e1: Expr* floatDecl declStmt booleanExpr +notExpr (e:Expr*) -_lex1: std::string -_d1 : Decl* -_lex1: std::string -_boolean:int +floatDecl +declStmt (lex:std::string) (d:Decl*) +booleanExpr andExpr (lex:std::string, e1: Expr* boolean:int) -_e2: Expr* stringDecl ifStmt -_lex1: std::string +andExpr _e1: Expr* mulExpr (e1:Expr*,e2:Expr*) -_s1: Stmt* +stringDecl - e1: Expr* (lex:std::string) +ifStmt -_e2: Expr* orExpr (e:Expr*,s:Stmt*) +mulExpr -_e1: Expr* matrixDecl (e1:Expr*,e2:Expr*) -_e2: Expr* ifElseStmt -_lex1: std::string +orExpr -_e1: Expr* -_lex2: std::string divExpr (e1:Expr*,e2:Expr*) -_s1: Stmt* -_lex3: std::string -_e1: Expr* -_s2: Stmt* e1: Expr* -_e2: Expr* nestedExpr +ifElseStmt -_e2: Expr* +divExpr -_e3: Expr* (e:Expr*,s1:Stmt*, -_lex1: std::string (e1:Expr*,e2:Expr*) s2:Stmt*) -_e1: Expr* +matrixDecl (lex1:std::string, +nestedExpr whileStmt plusExpr lex2:std::string, (lex:std::string .e:Expr*) lex3:std::string, -_e1: Expr* -_e1: Expr* e1:Expr*,e2:Expr*, -_s1: Stmt* -_e2: Expr* e3:Expr*) +whileStmt +plusExpr relationshipExpr (e:Expr*,s:Stmt*) (e1:Expr*,e2:Expr*) - lex1: std::string matrixDecl2 -_e1: Expr* printStmt minusExpr -_lex1: std::string -_e2: Expr* -_e1: Expr* -_e1: Expr* -_e1: Expr* +relationshipExpr +matrixDecl2 -_e2: Expr* (lex:std::string, +printStmt (lex:std::string, e1:Expr*,e2:Expr*) (e:Expr*) +minusExpr e1:Expr*) (e1:Expr*,e2:Expr*) semiStmt booleanDecl +semiStmt() -_lex1: std::string +booleanDecl (lex:std::string) repeatStmt -_lex1: std::string -_e1: Expr* -_e2: Expr* -_s1: Stmt* +repeatStmt (lex:std::string,

e1:Expr*,e2:Expr*,

s:Stmt*)