Filename	Functionality Tested	Points Possible	Mostly or completely incorrect (0% of points possible)	Needs improvement (50% of points possible)	Mostly or completely correct (100% of points possible)
ifSTR	If-expression's 1st expression can't be FUNCTION, LIST, NULL, or STR				
ifFUNCTION	If-expression's 1st expression can't be FUNCTION, LIST, NULL, or STR				
ifLIST	If-expression's 1st expression can't be FUNCTION, LIST, NULL, or STR				
ifNULL	If-expression's 1st expression can't be FUNCTION, LIST, NULL, or STR				
ifThenFUNCTION	If-expression's "then" expression can't be FUNCTION				
ifElseFUNCTION	If-expression's "else" expression can't be FUNCTION				
assignParamNULL	assignment-expression can only assign INT expression to a param variable				
assignParamSTR	assignment-expression can only assign INT expression to a param variable				
assignParamFLOAT	assignment-expression can only assign INT expression to a param variable				
assignParamLIST	assignment-expression can only assign INT expression to a param variable				
assignParamFUNCTION	assignment-expression can only assign INT expression to a param variable				
functOneArgTooFew	function-call cannot have too few args (e.g., no args)				
functMultArgsTooFew	function-call cannot have too few args				
functNoArgsTooMany	function-call cannot have too many args				
functOneArgTooMany	function-call cannot have too many args				
functMultArgsTooMany	function-call cannot have too many args				
functArgFLOAT	function-call must pass INTs (or BOOLs) as params				
functArgLIST	function-call must pass INTs (or BOOLs) as params				
functArgNULL	function-call must pass INTs (or BOOLs) as params				
functArgSTR	function-call must pass INTs (or BOOLs) as params				

functArgFUNCTION	function-call must pass INTs (or BOOLs) as params				
functFUNCTION	Function can't return a function				
forFUNCTION	for-expression 1st expression can only be LIST				
forNULL	for-expression 1st expression can only be LIST				
forINT	for-expression 1st expression can only be LIST				
forFLOAT	for-expression 1st expression can only be LIST				
forSTR	for-expression 1st expression can only be LIST				
forBOOL	for-expression 1st expression can only be LIST				
forIdentLIST	for-expression ident must be compatible with INT, BOOL, FLOAT, and STR if it exists before the for-expression				
forIdentFUNCTION	for-expression ident must be compatible with INT, BOOL, FLOAT, and STR if it exists before the for-expression				
forIdentNULL	for-expression ident must be compatible with INT, BOOL, FLOAT, and STR if it exists before the for-expression				
ifThenGood_INT	A correct if-expression (no else); result is INT				
ifThenGood_LIST	A correct if-expression (no else); result is LIST				
ifThenElseGood_INT	A correct if-expression (with else); result is compatible with INT				
ifThenElseGood_LIST	A correct if-expression (with else); result is compatible with LIST				
ifThenElseGood_Combo	A correct if-expression (with else); result is big combo type				
functNoArgsGood	A correct function call				
functOneArgGood	A correct function call				
functMultArgsGood	A correct function call				
functArgsINT	A correct function call				
assignParamGood	Correct assignments to function parameters				
functReturnsINT	Functions returns type of value which is correct for how it is then used				
functReturnsLIST	Functions returns type of value which is correct for how it is then used				
functReturnsFLOAT	Functions returns type of value which is correct for how it is then used				
functReturnsSTR	Functions returns type of value which is correct for how it is then used				
functReturnsBOOL	Functions returns type of value which is correct for how it is then used				

functReturnsNULL	Functions returns type of value which is correct for how it is then used				
functReturnsComboType	Functions returns type of value which is correct for how it is then used				
inputVarINTCompatible	Input variable is compatible with INT				
inputVarINTFLOATBOOLCompatible	Input variable is compatible with INT, FLOAT, BOOL				
forGoodIdentOld	Correct for-loop where ident previously existed				
forGoodIdentNew	Correct for-loop where ident didn't previously exist				