CIS*4650 (Winter 2018) --- Marking Scheme for Checkpoint One

Group	Questions	Comments
	Documentation (20)	
	Scanner (20): 1. Major token types: 2. Row Numbers: 3. Using Lex/Flex/JFlex:	
	Parser (40): 1. Parsing w/o Output: 2. Generating AST's: 3. Using Yacc/Bison/CUP:	
	Error Recovery (20): 1. Basic Reporting: 2. Major Components: 3. Extensive Recovery:	

Scanner: 1. Major token types: keywords, symbols, white spaces, identifiers, numbers, comments, and invalid characters. 2. Row numbers: required for error reporting 3. Must use one of the scanner tools	- Run fac.cm - Check Lex/Flex/JFlex file to verify the use of a scanner tool.
Parser: 1. Parse w/o output 2. Generate abstract syntax trees 3. Must use one of the parser tools	- Run fac.cm, gcd.cm, and sort.cm - Check abstract syntax trees for these programs - Verify the tree is displayed after being completely built - Check Yacc/Bison/CUP file to verify the use of a parser tool
Error Recovery: 1. Basic reporting: first error token with type, value, and row number. 2. Major components: recover with dec sequence, exp sequence, and expressions with multiple binary operations 3. Extensive recovery: recover with other refined structures.	- Introduce errors in some of the test files and verify the results.