



University of the Philippines Diliman
College of Engineering
Department of Computer Science

CS 150 MACHINE PROBLEM SPECIFICATIONS

Title:	Own Programming Language
Objective/s	to design and implement a new imperative programming language to apply learned concepts of programming language to create a debugger and interpreter/compiler for the designed language
General Instructions:	Design and implement your own programming language. All parts of your programming language are to be implemented in Python or Java.

MINIMUM SPECIFICATIONS

Implement an interpreter/compiler/translator for the designed language. For parsers, you can choose between LR or LL parsers as discussed in class.	
Implement a debugger that will check for syntax errors and display debugging information such as line number and details on errors	
Create an interface for coding, debugging and running the programs	
Design Requirements	Provide set of reserved and key words and their uses
	Define rules for naming identifiers (variables, subprograms and etc.)
	Variable scoping and binding will be your preference
	Support at least integers and floating points as primitive data types - may also include some other data type
	Statements such as assignment statement with basic arithmetic expressions should be implemented - follow the MDAS precedence
	Provide way or method for user input and outputting data
	Implement control structures for sequence, selection and iteration - should also include means of blocking compound statements and nesting possibilities.
	Assume that boolean expressions are limited to single relational expression that supports coercion
	Provide a way on implementing subprograms - passing parameters rule will be your preference.
Provide way of including comments in your program	

DELIVERABLES	
Part 1 - 6:	<p>Provide a written and detailed documentation of the constructs and designs that you will include in your programming language.</p> <p>Deadline: November 16, 2016 (Wednesday) - 11:59PM. SOFT COPY. You can submit all parts or by part before the deadline.</p>
Part 1: Lexical and Syntax Analysis	<p>What would be the name of your programming language?</p> <p>Short Introduction about your programming language (inspiration)</p> <p>What lexical and syntax analyzer you will use in your programming language and how do you plan on implementing it?</p>
Part 2: Names, Binding and Scoping	<p>Are your names case sensitive?</p> <p>What are the reserve words in your language? how about keywords?</p> <p>What will be the name form?</p> <p>Binding Type of your programming language</p> <p>Lifetime and scoping of your identifiers</p> <p>How to represent blocks?</p> <p>Grammar Definition of Identifiers - valid identifiers for variables and constants</p>
Part 3: Data Types	<p>Primitive data types (numeric, boolean, character types) available</p> <p>Will your language support strings? if yes would it be primitive or special kind? static or dynamic length? what operations are available?</p> <p>Will your language support user-defined data types? how about arrays, associative arrays, records, tuples, list, unions and pointers? give details on this if it's available in your language.</p> <p>Type checking - allows coercion and typecasting</p> <p>Grammar Definition for Data Types</p>
Part 4: Expression and Assignment Statement	<p>Arithmetic Expressions - operators, operands, groupings, unary/binary/ternary, infix/prefix</p> <p>Operator overloading</p> <p>Type conversions - narrowing and widening; coercions and type casting</p> <p>Relational and Boolean expressions</p> <p>Assignment Statements - simple/multiple assignments, conditional targets, assignment operators</p> <p>Grammar Definition of Expressions and Assignment Statements</p>
Part 5: Statement Level Control Structures	<p>Selections Statements - form and type of expressions that controls the selection; then-else clauses, means of blocking in nested selectors; multiple selection structure (switch or multiple if)</p> <p>Iterative Statements - counter controlled loops; logically controlled loops; user-located loop control; nested iterations</p> <p>Grammar Definition of Statement Level Control Structures</p>
Part 6: Subprograms	<p>Subprogram definition (header, parameters, return types)</p> <p>Subprogram calls and returns</p> <p>Global and local variables - scoping and allocation</p> <p>Location of subprogram definitions</p> <p>Parameter Passing methods</p> <p>Parameters format and types</p> <p>Overloaded or Generic subprograms availability</p> <p>Allows recursions?</p> <p>Grammar Definition of Subprograms</p>

Part 7: Final Output	Deadline: Dec 1, 2016 (Thursday) - 11:59PM. SOFT COPY Please make sure you sign-up in the demo schedule sheet.
Part 7.1: User Manual	Step-by-step instruction on how to use your compiler/interpreter of your own programming language. Include also instructions on how to use your interfaces for coding, debugging and running
Part 7.2: Source Code, Executable/Installable	Zipped your source codes with filename: <Section>_<ProgrammingLanguage>.zip
Part 7.3: Test Files	Provide 3 sample programs that tests the capability of the implemented compiler/interpreter. The first sample program should be a program that outputs all leap years within a range of years inputted by the user. The second sample program will be your preference but make sure it can show the capability of your programming language. The third sample program will be a program to test your debugger. Pepper your test files with comments.

SUBMISSION DETAILS		
Format:	A4 bond paper, Arial, Font-size: 10. PDF File	
Email Address:	All submissions should be sent to teacher.ada.submissions@gmail.com as an attachment. Submissions made other than to this email will NOT be checked.	
Email Subject:	CS 150 <Section> <Group Name> <Item>	
Filename:	All soft copy submissions should have a file name with the following format: <Group Name>_<Item>.pdf e.g ltoAngGroupKo_DataTypes.pdf	
Grading Breakdown	Part 1 - 6 (10 points each)	60
	Functionality	50
	Error Checking and Debugging Info	25
	Demo and Sample Programs	40
	Language Manual	15
	Evaluation	10
	Total	200
Deductions:	Late submission of MP will be penalized with 10 PTS per day. A maximum of 1 week will be given for late MPs. Afterwards the MP will merit a grade of 0.	
	Please follow all instructions. There will be 5 points deduction for every instruction not followed.	
	Even if the MP will merit a grade of zero, students are required to submit all parts of the MP.	
	The coursework cannot be completed at the last minute. Spread the work over the time provided and to the team members accordingly.	

Reminders	<p>Do not assume anything, If there are questions about the specifications, ask your teacher. This is very important. Please NEVER hesitate to approach or ask . You can ask via the facebook group; facebook private message, email or consultations.</p>
	<p>The requirement is for group work. Committing plagiarism or assisting another group committing plagiarism shall be dealt with accordingly.</p>