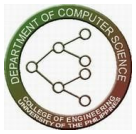
DEPARTMENT OF COMPUTER SCIENCE
College of Engineering
University of the Philippines – Diliman



CS 150 Machine Problem Specs

MP Grade Breakdown:

Language Design and Functionalities	**120 pts**
Implementation	**100 pts**
Platform/Environment for dev & exec	20 pts
Interpreter/Compiler	50 pts
Debugger	20 pts
Source Code documentation	10 pts
Language & User Manual	**10 pts**
Demo & Sample Programs	**20 pts**
TOTAL	**250 pts**

Student's Grade **AP * TOTAL**

where AP is the average percentage score from the peer evaluation (0-100%)

Guidelines:

1. This requirement is to be accomplished by group (grouping is the same as the group report). Committing plagiarism or assisting another group committing plagiarism shall be dealt with accordingly.
2. Format for papers (soft/hardcopy):
    a. A4 bond paper
    b. Font: Arial
    c. Font size: 10
    d. File type: PDF
    e. Binding: Ring (for hardcopy)
    f. Special comments: Include page numbers.
3. The coursework cannot be completed at the last minute. Spread the work over the time provided and to the team members accordingly.
4. 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.
5. The demo will be on **Dec 4, 1pm-6pm, 30 mins per group**. Reserve a slot for your group's demo. Sign-up sheet will be posted at Rm. 307 on Oct 1, 2015.

DEADLINE: Submit the requirements below:

1. Language Design & Functionalities (PDF)
2. Source Code (include in-code comments for documentation)
3. Source Code Documentation (PDF)
4. Language Manual (PDF)
5. Sample Programs

on or before Dec 3, 2015- 5PM-

- **Soft copy** of req 1-5 at rapineda1@upd.edu.ph
  - Email Subject: **CS150 <SECTION> - MP - <PROGRAMMING LANGUAGE>** (Ex: CS150 TGI/HVW - MP - JOSE RIZAL LANGUAGE)
  - **Do not** compress all your files. Attach each file individually.
  - Submit only **ONCE**. Your email should contain all the files for submission and the names of your groupmates.
- **Hardcopy** of req 1,3,4 at RM 307
- **Peer evaluation** per group member sealed in a white envelope. If there are six members in your group, I should receive 6 envelopes from your group. Submit this along with the hardcopy of your MP documentation. Keep your evaluation **confidential**.

MINIMUM SPECIFICATIONS

1. Implement an interpreter/compilers for the designed language. For parsers, you can choose between LR or LL parsers as discussed in class.
2. Implement a debugger that will check for syntax errors and display debugging information such as line number and details on errors
3. Create an interface for coding, debugging and running the programs

DESIGN REQUIREMENTS:

1. Provide set of reserved and key words and their uses
2. Define rules for naming identifiers (variables, subprograms and etc.)
3. Variable scoping and binding will be your preference
4. Support at least integers and floating points as primitive data types - may also include some other data type
5. Statements such as assignment statement with basic arithmetic expressions should be implemented - follow the PMDAS precedence
6. Provide way or method for user input and outputting data
7. Implement control structures for sequence, selection and iteration - should also include means of blocking compound statements and nesting possibilities.
8. Assume that boolean expressions are limited to single relational expression that supports coercion
9. Provide a way on implementing subprograms - passing parameters rule will be your preference.
10. Provide way of including comments in your program

DELIVERABLE:

1. **Language Design & Functionalities Report**
    a. **Introduction**
        i. Language Name
        ii. Paradigm
        iii. Inspiration
    b. **Grammar Definition** using BNF/EBNF/Attribute Grammar/Semantics of the ff:
        i. Identifiers - valid identifiers for variables and constant
        ii. Data Types
        iii. Expressions and Assignment Statements
        iv. Statement Level Control Structures
        v. Subprograms
    c. **Lexical and Syntax Analysis-** LR or LL parsing
    d. **Names, Binding, and Scoping-**
        i. Are your names case sensitive?
        ii. What are the reserve words in your language? How about keywords?
        iii. What will be the name form?
        iv. Binding Type of your programming language
        v. Lifetime and scoping of your identifiers
        vi. How to represent blocks
    e. **Data Types**
        i. Primitive data types (numeric, boolean, character types) available
        ii. Will your language support strings? if yes would it be primitive or special kind? static or dynamic length? What operations are available?
        iii. 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.
        iv. Type checking - allows coercion and typecasting
    f. **Expression and Assignment Statements**
        i. Arithmetic Expressions - operators, operands, groupings, unary/binary/ternary, infix/prefix
        ii. Operator overloading
        iii. Type conversions - narrowing and widening; coercions and type casting
        iv. Relational and Boolean expressions Assignment Statements - simple/multiple assignments, conditional targets, assignment operators
    g. **Statement-Level Control Structures**
        i. 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)
        ii. Iterative Statements - counter controlled loops; logically controlled loops; user-located loop control; nested iterations

h. **Subprograms**
    i. Subprogram definition (header, parameters, return types)
    ii. Subprogram calls and returns
    iii. Global and local variables - scoping and allocation
    iv. Location of subprogram definitions
    v. Parameter passing methods
    vi. Parameters format and types
    vii. Overloaded or Generic subprograms availability
    viii. Allows recursions?

2. **Language/User Manual-**
   Step-by-step instruction on how to use your compiler/interpreter of your programming language. Include also instructions on how to use your interfaces for coding, debugging and running.
3. **Sample Programs**
   Provide 3 sample programs that tests the capability of the implemented compiler/interpreter.
    1. The first sample program should be a program that outputs all leap years within a range of years inputted by the user.
    2. The second sample program will be your preference but make sure it can show the capability of your programming language.
    3. The third sample program will be a program to test your debugger. Pepper your test files with comments
4. **Evaluation**
   Peer Evaluation - Each member of the group will be evaluated (self and peer) based on contributions. Given a total of 150 points, allocate the adequate amount of points per groupmate (including yourself) based from the contributions rendered to the group. Justify your score. Use the attached evaluation sheet.

   Ex: Riza - 200, Rae - 10, Pineda - 0

DEDUCTIONS

1. Late submission of MP will be penalized with a deduction of **10 points** per day.
2. A maximum of **1 week** will be given for late MPs. Afterwards, the late MP will merit a grade of **0**.
3. Follow all instructions. There will be **5 points deduction** for every instruction not followed.
4. Even if the MP will merit a grade of zero, students are required to submit the MP.
5. If your source code does not work due to errors or some other reasons, the maximum grade that you will get under Implementation is 30 points. There will be minimal partial points given to programs that do not work.

DEPARTMENT OF COMPUTER SCIENCE
College of Engineering
University of the Philippines – Diliman

CS 150 Machine Problem Evaluation Sheet

Justly allocate a total of 150 points among the members of your group.

Name	Score	Justification
1.		
2.		
3.		
4.		
5.		
6.		

_____

Signature above printed name

Date Signed: