Birla Institute of Technology & Science-Pilani, Hyderabad Campus
First Semester 2018-2019
Principles of Programming languages (CS F301)

Due Date for Assignment 1: 23-10-2018

Assignment-1(Syntax of programming languages)   Max marks:20

In this assignment you are expected to write the syntax of your chosen mini programming language. You can find the list of around 256 different programming languages on the following link.

Link: https://dzone.com/articles/big-list-256-programming

Every group will have a unique language.

NOTE: Register your team (max 3) and language name with Mr. Jabez in H-109during the two slots mentioned below. Avoid last minute rush to get your favorite language since the language choice will be first come first serve.

Slots:

Tuesday (16/10/2018) 8am to 11am
Wednesday (17/10/2018) 10am to 1pm

Your grammar has to include the following

- sequential statements - **2 marks**

- a loop (for, while, do...while, for each etc.,) - **4 marks**

- a conditional statement (if, switch etc.,) - **4 marks**

- a function (function declaration, function definition and function call) - **4 marks**

- **Arithmetic , logical and relational operators - 2+2+2 marks**

**Use operators of different precedence (eg. * and +)

Partial Syntax for C is given below for your reference:

```
<program> → <header><function><main>
<main> → main()<stmnts>
<stmnts> → <stmnt>;<stmnts> | <stmnt>
<stmnt> → <condStmnt>| <LoopStmnt> | <DeclStmnt>| <AssignStmnt> |<function_call>
```

<condStmnt> → <if_stmt> | <switch_stmnt>

<LoopStmnt> → <For_stmnt> | <while_stmt>

<AssignStmnt> → id = <arithmetic_exp>

<DeclStmnt> → <Type> <VarList>;

<VarList> → <VarList>,id | id;

<arithmetic_exp> → <arithmetic_exp> + <var> | <var>

Assignment 2: Type Checking   Maximum Marks : 20

Goal and motivation:
This assignment aims at testing your understanding on type equivalence and type checking (Name, Internal Name and Structural).

Problem Statement:
Part1:
In this assignment you are expected to write code in C/C++/Java to implement the rules of Name, Internal Name and Structural Equivalence using the methods discussed in the class. Refer to slides on Type systems.

1. Your code should include the rules for name equivalence checking. The brief rules are mentioned below. Refer to the lecture notes for detailed rules.

    •   Types must have the exact same name to be equivalent

2. Your code should include the rules for Internal name equivalence checking. The brief rules are mentioned below. Refer to the lecture notes for detailed rules.

    •   If the program interpreter gives the same internal name to two different variables, then they share the same type

3. Your code should include all the five rules for structural equivalence and the table entry manner to check for structural equivalence. Here is a short description of the five rules. Refer to the lecture notes for a detailed explanation.

    A. Same built in data types are structurally equivalent.
    B. Pointers to structurally equivalent types are structurally equivalent.
    C. Determining struct structural equivalence
        a. Two structures
        b. st1 { x_1: W_1, x_2: W_2, ..., x_k: W_k }
        c. st2 { y_1: Q_1, y_2: Q_2, ..., y_k: Q_k }
        d. st1 and st2 are structurally equivalent iff
            i. W_1 structurally equivalent to Q_1
            ii. W_2 structurally equivalent to Q_2
            iii. ...
            iv. W_k structurally equivalent to Q_k

    D. Determining array structural equivalence
       a. Two Arrays
       b. T1 = array range1 of t_1
       c. T2 = array range2 of t_2
       d. T1 and T2 are structurally equivalent iff:
          i. range1 and range2 have (1) the same number of dimensions and (2) the same number of entries in each dimension
          ii. t_1 and t_2 are structurally equivalent
    E. Determining function structural equivalence
       a. Two functions
       b. T1 = function of (t_1, t_2, t_3, ..., t_k) returns t
       c. T2 = function of (v_1, v_2, v_3, ..., v_k) returns v
       d. T1 and T2 are structurally equivalent iff:
          i. For all i from 1 to k, t_i is structurally equivalent to v_i
          ii. t is structurally equivalent to v

Use the structural equivalence algorithm as discussed in class (By constructing tables till T(N)= T(N-1)) to determine which two types are structurally equivalent.

Part2:
For the language you have chosen in Assignment 1, conduct a literature review of when and where name and structural equivalence is used. In the absence of such rules for name and structural equivalence formulate your own rules and submit the report.

For eg: In a language called DisneyLand (assuming this is the language I took in assignment-1),
The rules are:
1. Name equivalence is used for basic data types.
2. Structural equivalence is used for structures.
3. Internal name equivalence is used for arrays.

So, you are expected to write a code which checks all of the three rules for my language using the name and structural equivance rules in Part1.
Eg:
A code snippet in DisneyLand language is as follows:
Testcase1:
int a,b;

Testcase 2:
struct foo{
int h,
string i;
};
struct cat{
int d;

```
string f;
};
```

Testcase3:
int array[0-4] q,r;

Step by step explanation of how your code should work:
This would go to code 2 which would say,
In my language basic datatypes use name equivalence, so check if a and b are name equivalent.
This would go to code 1 which would say,
a,b are name equivalent.

Similarly do for test case 2 and 3 where code in part 2 specifies what should be checked for a particular data type and code in part 1 specifies how it should be checked. I have specified sample testcases and your test cases would vary based on your language and its syntax.

Deliverables:
1. Code for part1
2. Code for part2
3. Rules for part 2
4. At least 3 test cases and their outputs
5. readme.txt file specifying the execution process of the codes and input and output folders

Submissions: The assignment has to be submitted before the due date (11.59 p.m. on 23/10/2018) to email id:- ppl.twenty18@gmail.com
The subject of the email submission should be "**Assignment-1 First Sem 2018-19**".