BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
HYDERABAD CAMPUS
FIRST SEMESTER 2019 – 2020
PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301) – COMPREHENSIVE EXAM

Date:10/12/2019 **Weightage:35%[70 Marks]**
Duration: 180 mins **Type: Closed Book**

Please note: 1. All parts of the questions have to be answered consecutively.
2. Your answers should be brief.
3. No doubts would be clarified in case anything is missing make your assumption and answer.

Q1. Syntax and semantics of programming languages

A. You are asked to design a new programming language for robot's movement. Using this new language, a programmer can write a program to move the robot from its current location to the other using moves like Left, Right, Forward and Backward.
For this language to be designed consider the following Context Free Grammar G = (V, T, P, S) where V={Program,MMoves,SMove}, T={begin,end,left,right,forward,backward};S={Program}.
Answer questions i to v
i. For the sample program given below and give the set of production rules P. **[3 M]**
ii. Using your solution in question i draw the parse tree for the sample program **[2 M]**

Sample program

```
begin
    left
    right
    backward
end
```

iii. Given the robot's initial positon in a Cartesian plane is (0,0) and the definition of other moves in the Cartesian plane are as follows **[3 M]**
Left: x=x-1, y=y
Right: x=x+1, y=y
Back: x=x, y=y-1
Forward: x=x, y=y+1
Write synthesized attribute grammar to compute the total distance (assume Manhattan distance) travelled by the robot at the root node.
iv. Write inherited attribute grammar to get the final location of the robot at the root node. **[4 M]**
v. Using your solution in question **iii** and iv draw a single decorated parse tree for the sample program given in question **ii**. **[5 M]**

B. Compute the precondition for the following three code segments. **[1+2= 3M]**

{ ?? }
x = x+2;
x = x+1
{ x = 7 }
	{ (??}
	if (y < 0)
	    x := x+1
	else
	    x := y

Q2. Data Types

A. Given the following declarations answer questions **i-iii.** **[3 M]**

type alink = pointer to cell;
subtype blink = alink;
p, q : pointer to cell;
r : alink;
s : blink;
t : pointer to cell;
u : alink;
	i. According to structural equivalence which variables will be treated equal.
	ii. According to Strict name equivalence which variables will be equal.
	iii. According to Loose equivalence which variables will be equal.
	[Note there may be multiple equivalences in each question according the given declaration just show all equal variables in brackets]

Q3. Memory Management

A. The following code uses a macro to define INCR function with an argument x. Answer questions **i-iii**

```c
#include <stdio.h>
#define INCR (x) ++x
int main() {
   char *ptr = "PPLCOMPRE";
   int x = 10;
   printf("%s  ", INCR(ptr));
   printf("%d", INCR(x));
   return 0;
}
```

i. What will be the output? **[2+2+1 = 5 M]**
ii. What problem does the INCR macro pose?
iii. Which phase of the compiler fails to detect it?

B. Given the following C like code snippet (where nesting of functions is allowed) and the function call sequence P-> R-> Q-> T-> S-> Q-> T-> S. Answer questions **i - iii.** **[3+1.5+1.5=6 M]**

```
P() {     //P function definition
    Q(){     //Q function definition
         S(){  //S function definition
              Q();   //function call to Q
         }//end S
         T(){   //T function definition
              S();     //function call to S
         }//end T
         T();   //function call to T
    }// end Q
    R(){    //R function definition
         Q();    //function call to Q
    }//end R
    R(); //function call to R
}//end P
```

i. If the language follows static linkage for resolving variable references show the sequence of activation records including the static links in the stack.
ii. Assume that the activation record consists of only local variables, parameters and return address. At what stage of the compilation process these values are available?
iii. Is it possible to estimate local variables, parameters and return address at the compile time? Justify why or why not.

Q4. Functional Programming

A. Prove i & ii using the appropriate λ-calculus encodings given below **[2+3= 5 M]**

i. or false true = true **ii.** if false then x else y = y	λ-calculus encodings or = λx. λy. ((x true) y) true = λx.λy.x false = λx.λy.y if a then b else c = a b c

B. Beta reduce the following lambda expression using Normal and applicative order. Do they produce the same result why or why not? **[Note you need not expand the encoding if, true and false assume their usual meanings]** **[5 M]**
if (less 3 4) (plus 5 5) (divide 1 0)

C. Write a program to compute the sum of all elements of a list in racket using tail recursion. **[3 M]**

Q5. Logic Programming

A. For the following two statements in Prolog where the LHS is the query and RHS is the predicate in the knowledge base, show the variable bindings if the following queries succeed. **[3 M]**
f(X,a,h,g(Y)) = f(h(Z), Z, h(W)).
p(X,g(b,f(a)),L).= p(a,g(Z,f(X)),[X|Z]).

B. Given the following set of clauses, answer questions **i-ii** **[1+3=4 M]**

travel(X) :- on_vacation(X), has_money(X). on_vacation(mary). on_vacation(peter). has_money(peter).	**i.** List all answers generated for ?- on_vacation(X). **ii.** Draw the Prolog search tree for travel(X).

C. Write a (recursive) prolog program to check weather a given list is in sorted order. **[4M]**

Q6. Heap Management

A. Figure A shows the snap shot of heap memory, answer questions i and ii.

i. Apply mark and scan algorithm for garbage collection.
[Note: Show both mark and scan steps separately] **[4 M]**

ii. What is the problem with Mark and Scan algorithm and how copy garbage collector overcomes this problem? **[2 M]**

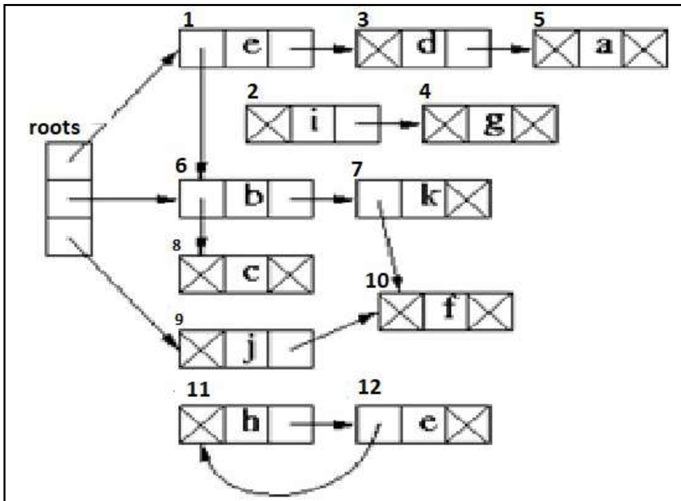

Figure A

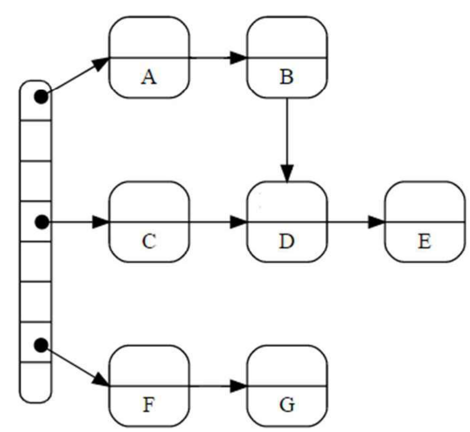

Figure B

B. Refer Figure B, show how the reference count algorithm works if delete(A) is encountered in the code. **[2 M]**

Q7. Comparison of all programming paradigms **[4 M]**

Write a functional program (using λ-calculus) and logic program (using prolog) which performs the same task as this imperative program given below **[Note: assume that sqrt predefined function exist]**

```
int func (int x,int y){
        return sqrt(a*a+b*b);
}
```

*************************** *That's all folks* ****************************