BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
PILANI CAMPUS
FIRST SEMESTER 2017 – 2018
PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301)
COMPREHENSIVE EXAMINATION

Date: 07.12.2017 **Weightage: 30 % (60 M)**

Duration: 2 hours **Type: Closed Book**

Note: Answer all parts of the question together.

Answers must be brief. **Total no. of pages: 4**

Q1. Fill in the blanks: **[10 * 1 = 10 marks]**

a) An activation tree is a ____________ (static/ dynamic) computation.
b) A task may be ______________ (implicitly / explicitly) started while a subprogram is ______________ (implicitly / explicitly) started.
c) LISP uses ____________ (static / dynamic) scoping while Scheme uses ______________ (static/ dynamic) scoping.
d) EQ? works for ______________ (symbolic/ numeric / both symbolic and numeric) atoms while EQV? works for __________________ (symbolic / numeric / both symbolic and numeric).
e) Using Semaphores to provide Synchronization creates an unsafe environment. In Cooperation Synchronization, leaving the wait (emptyspots) statement out of the _____________ (producer / consumer) task would cause a buffer overflow.
f) The difference between the static_depth of the subprogram containing the reference and the static depth of the subprogram containing the declaration of that non-local variable is given by the _______________ or the _____________.
g) The four categories of variables are _____________, _________, ____________ and ____________.
h) A member that is not accessible in a subclass because of private derivation can be declared to be visible there using ________________________________.
i) For resolution to work, the propositions must be in _________________ form.
j) The difference between ML and Haskell are _________________ and ______________________.

Q2. Answer the following questions: **[2 + 4 + 1 + 2 + 2 + 3 + 3 + 1 + 2 = 20 marks]**

a) What are the design issues involved in allocation and de-allocation of objects in an object oriented language such as C++?
b) Write the return semantics for a subprogram with stack dynamic local variables considering that the environment pointer (EP) points to the base address of the ARI of the calling program at the beginning of the call.
c) Why the dynamic binding of C++ is faster than Smalltalk?
d) Write the code for the append function in ML. Append function takes two lists as parameters and returns the result as a list containing the second list appended to the first list.
e) Consider 2 tasks A and B trying to access a shared variable ‘result’. A wants to divide ‘result’ by 3 and B wants to add 3 to ‘result’. Assume that the initial value of ‘result’ is 9. Give the complete scenario in which inconsistency in the value of ‘result’ would arise without competition synchronization.
f) Write the code to discuss how cooperation synchronization can be achieved using semaphores for the producer consumer problem. Assume that there are 2 semaphores used ‘fullspots’ and ‘emptyspots’ to indicate the no. of filled and empty locations in the shared buffer respectively.
g) Draw the trace and the control flow model for the following goal “functor (A, X), functor (D, X)” given the database as:

functor(A,B).	functor(A,C).	functor(D,E).	functor(D,C).

h) Consider a Prolog statement: Result is Result+1. What is the error in this statement?
i) What are the intrinsic limitations of Prolog? Explain with the help of the example of sorting a list. Also, write the program in Prolog for the same.

Q3. Answer the following: **[6 + 2 + 2 + 2 = 12 marks]**

a) Below is given a prolog program to rotate a list 'n' places to the left. For your understanding, the meaning of the functors used is given in the comments below. Fill in the missing code:

% rotate(L1,N,L2) :- the list L2 is obtained from the list L1 by rotating the elements of L1 N places to the left.
% Examples: rotate([a,b,c,d,e,f,g,h],3,[d,e,f,g,h,a,b,c]), rotate([a,b,c,d,e,f,g,h],-2,[g,h,a,b,c,d,e,f]),
% (list,integer,list) (+,+,?)
% split(L,N,L1,L2) :- Split a list into two parts, the list L1 contains the first N elements of the list L, the list L2
% contains the remaining elements,
% (list,integer,list,list) (?,+,?,?)

1. rotate(list,integer,__________).
2. length(list,integer).
3. rotate_left(list,integer,___________).
4. append(list,list,_________).
5. split(list,integer,______,_______).
6. rotate(L1,N,L2) :- N >= 0, length(L1,NL1), N1=_________________, rotate_left(L1,N1,L2).
7. rotate(L1,N,L2) :- N < 0, length(L1,NL1), N1=_________________, rotate_left(L1,N1,L2).
8. rotate_left(L,0, L).
9. rotate_left(L1, N, L2) :- N > 0, split(L1,N,S1,S2), __________________.
10. append([], Listb, Listb).
11. append(_______________,______________,______________):- append(List1,List2,List3).
12. split(L,0,______, _______).
13. split(_______, N, ________,_____) :- N > 0, N1=________________, split(Xs,N1,Ys,Zs).
14. length([],0).
15. length([_|L],N) :- _______________, _________________.

b) Write the following English conditional statements as Prolog headed Horn clauses:
   (i) If Ron feels hungry, then he eats quickly.
   (ii) If he eats quickly, he gets heartburn.
   (iii) If he gets heartburn, he takes medicine.
   (iv) Ron feels hungry.

c) Consider the following set of clauses given in the first column of the table below in a prolog program. List all answers generated for the following queries from (i) to (vi):

father(bob,julie).	(i) father(bob,julie).
husband(bob,Alice).	(ii) eats(bob,oranges).
eats(bob,pizza).	(iii) husband(X,Alice).
eats(julie,pizza).	
eats(Alice,pizza).	(iv) eats(X,pizza).
eats(julie,oranges).	(v) eats(X,pizza), eats (X,oranges).
bought(bob,pizza,Alice).	(vi) tired(Alice).
tired(julie).	

d) What is the need of backtracking in prolog? How we can control the backtracking? Why do we need such control?

Q4. Answer the following: **[6 + 2 + 2 = 10 marks]**

a) Consider the C like code snippet given below:

```
list_update(int a, int b, int c, int d)
{
    int ref_list [ ] = {1,2,3,4,5,6,7,8,9};
    a--;
    b = a+b;
    c = ref_list[a]+1;
    d= ref_list[b]+2;
}
void main( )
{
    int list[ ] = {7,9,8,3,5,2,6,4,1};
    int i =3, j =5;
    list_update (i, j, list[i], list[j]);
    print("\n %d %d", i,j);
    for (z = 0 ; z < 9 ; z++) { print("%d", list[z]); }
}
```

Assuming that the index of the array starts with '0', what will be the output of the above program in the following cases:

(i) All parameters are passed by reference.
(ii) Pass i and list[j] by value, and pass j, list[i] as value-result.
(iii) All parameters are passed by result (assuming that the values are copied back in left-to-right order and binding is done at return time). **Additionally assume that variable 'a' and 'b' are initialized to '1', '2' respectively for this case only.**
(iv) All parameters are passed by name.

b) Answer the following about object oriented languages:

1	In C++, if objects are allocated from the heap, then object slicing will not occur.	True/ False
2	Java does not support protected derivation of C++.	True/ False
3	Java messages are statically bound to methods.	True/ False
4	In Java, all objects are stored as stack dynamic objects.	True/ False
5	Java supports multiple inheritance.	True/ False
6	A pure virtual class in C++ should have atleast one pure virtual function.	True/ False
7	In C++, a protected variable of a base class when inherited in protected mode in a derived class A will be accessible in the 2nd derived class derived from the class A.	True/ False
8	In C++, methods that are statically bound are stored as a pointer in the class instance record.	True/ False

c) What will be the output of the given C like program in case dynamic scoping is used?

```
P()
    {
    int x, y;
    x = 5; y = 7;
    R()
        {
            int z;
            z = x+y;
            printf("%d %d %d\n", x,y,z);
        }
    Q()
        {
            int y,z;
            y = 1;  z = 11;
            z = x+y+z;
            R();
            printf("%d %d %d\n", x,y,z);
        }
    Q();
    R();
}
main()
{  P();
}
```

Q5. Answer the following: **[3 + 1 + 4 = 8 marks]**

a) Given a function f_1: $X_1 \rightarrow Y_1$ and the relation between the arguments is $X_3 <: X_1 <: X_2$ and the relation between the return types is $Y_3 <: Y_1 <: Y_2$, check if the following functions are a subtype of 'f_1' or not. Write your answer as a (Yes/ No).

(i) $f_2 : X_2 \rightarrow Y_1$
(ii) $f_3 : X_2 \rightarrow Y_3$
(iii) $f_4 : X_1 \rightarrow Y_2$
(iv) $f_5 : X_1 \rightarrow Y_3$
(v) $f_6 : X_3 \rightarrow Y_3$
(vi) $f_7 : X_3 \rightarrow Y_1$

b) Find the loop invariant P for the following piece of code written in C like language for calculating the sum of the series starting from 'm' uptil 'n' and storing the sum of 'm....n' in z.

```
int z= 0;
int k= m;
// invariant P
while (k <= n)
{
  z= z + k;
  k= k + 1;
}
```

c) Write the BNF and EBNF grammar for java class header. The rules for the java class header are given below:

1. Starts with keyword 'class' followed by the class name.
2. The class can have access modifier as public, abstract or final which will be preceding the keyword "class".
3. The name of the base class contains one uppercase alphabet from (A|B|C|…|Z)
4. A class can extend one base class or it may not extend a class. The keyword "extends" is used to extend a base class.
5. The name of the derived class contains one uppercase alphabet from (A|B|C|…|Z)
6. A class can optionally implement an interface using the keyword "implements".
7. The interface name starts with "I" and followed by one lowercase alphabet from (a|b|…|z).
8. If a class extends a base class and implements an interface too, then the class will extend the base class first followed by implementing the interface.
9. A class can implement any no. of interfaces. In this case, multiple interface names are separated by commas.

Examples:
class B
abstract class B
class A extends B
public class A extends B
class A extends B implements Ib
class A extends B implements Ib, Iz
class A implements Ib

END

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
PILANI CAMPUS
FIRST SEMESTER 2017 – 2018
PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301)
COMPREHENSIVE EXAMINATION

Date: 07.12.2017 **Weightage: 15 % (30 M)**

Duration: 1 hour **Type: Open Book**

Note: Answer all parts of the question together.

Answers must be brief. **Total no. of pages: 4**

Q1. Consider following C code snippet and answer the questions that follow: [**6 marks]**

```
int **ptr = NULL, N = 5, i;
ptr = (int**) malloc(sizeof(int*) * N);
for(i = 1; i < N; i++)
        ptr[i] = (int*)malloc(sizeof(int)* (N-i));
int * q = (int*)malloc(sizeof(int)* N);
ptr[0] = q;
for(i = 0; i < N; i++)
        if(i % 2 == 0)
                free(ptr[i]);
ptr[2] = ptr[3];
ptr[4] = ptr[3];
ptr[1] = ptr[3];
=========== location 1 ==========
free(ptr);
=========== location 2 ==========
```

Assume that int type takes **8 bytes** of memory while pointer type takes **4 bytes**

	Leaked Memory in Bytes	List of Dangling Pointers
at location 1		
at location 2		

Number of tombstones at location 1, if **Tombstone based approach** for detecting dangling pointers is used	
Number of keys at location 1, if **Lock and Key based approach** for detecting dangling pointers is used	

Q2. Consider following C code snippet and answer the questions that follow: **[1 + 1 + 4 = 6 marks]**

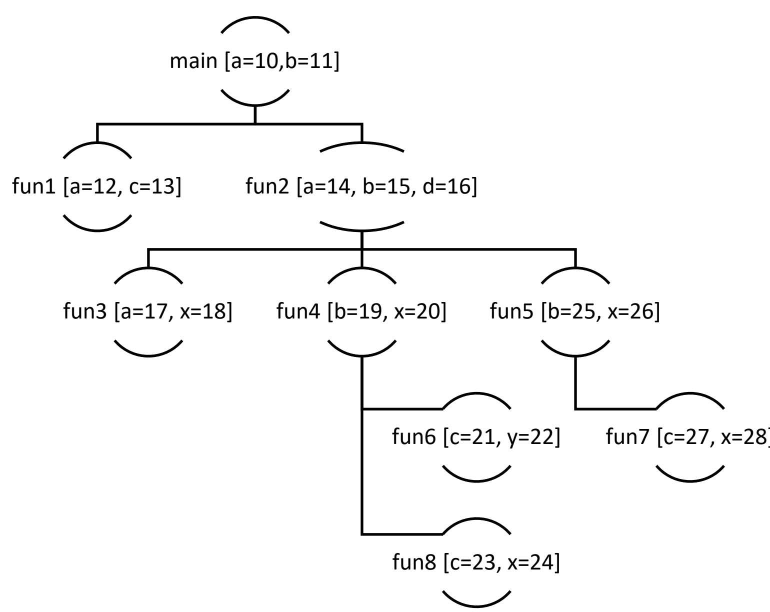
```
int **ptr = NULL, N = 5, i, *x, *y;
ptr = (int**) malloc(sizeof(int*) * N);
for(i = 1; i < N; i++)
        ptr[i] = (int*)malloc(sizeof(int)* (N-i));
x = (int*)malloc(sizeof(int)* N);
ptr[0] = x;
y = (int*)malloc(sizeof(int)* N);
ptr[1] = y;
y = (int*)malloc(sizeof(int));
ptr[0] = y;
y = (int*)malloc(sizeof(int));
ptr[2] = y;
```

‘int’ type takes **8 bytes** of memory while pointer type takes **4 bytes.** Copying collection Garbage collection mechanism is used and total available heap size is 400 bytes, equally split between “From Space [locations 0 to 199]” and “To Space [locations 200 to 399]”. In “From Space” memory allocation for a new block starts at next consecutive lowest available address which is multiple of 20. Assume that garbage collector is called as soon as memory allocation fails for the first time. Answer the questions below:

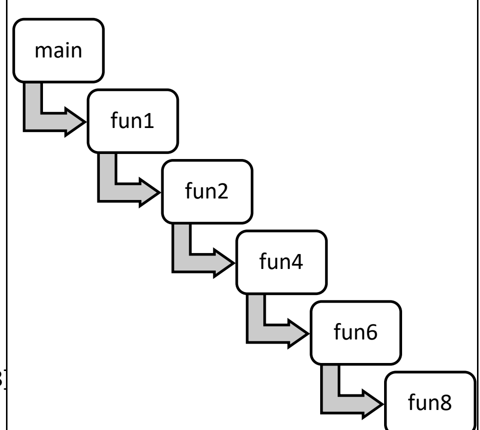
a) Line number in which the garbage collector is called for the first time? _____________.
b) Size of the largest free contiguous memory available after the process of garbage collection is over and the previously failed allocation has succeeded? ______________________.
c) Fill in the table below for the situation after the garbage collection is finished and the previously failed allocation has succeeded . Marks will be awarded only if memory locations for both the spaces are correct for a pointer:

Start address of memory pointed to by this pointer	**Before Garbage Collection**	**After Garbage Collection**
ptr		
ptr[0]		
ptr[1]		
ptr[2]		

Q3. Consider a hypothetical language which allows nested function definition. A code has been written in the language to implement following “function definition hierarchy”. The variables written in square brackets represent the local variables defined in the given function block along with their initialized value i.e. main has two local variables “a = 10” and “b = 11” and fun7 has local variables “c = 27” and “x = 28”. **[4 + 4 = 8 Marks]**



Function definition hierarchy

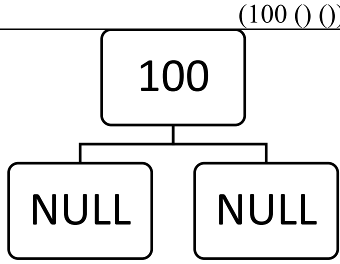
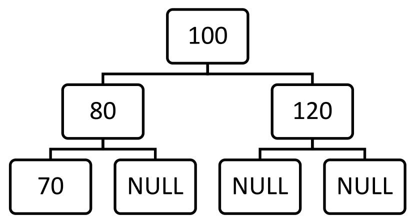


Function call sequence

With respect to above Function Definition Hierarchy and Function Call Hierarchy, fill in the values of cells in following tables if STATIC and DYNAMIC is scoping used. Mark “X” if a variable is not visible in a scope. Marks will be awarded only if all values of a particular scope are correct.

Value of variables in different function scopes if STATIC Scoping is used						Value of variables in different function scopes if DYNAMIC Scoping is used					
	Variables						Variables				
Scope	a	b	c	x	y	Scope	a	b	c	x	y
fun1						fun1					
fun4						fun4					
fun6						fun6					
fun8						fun8					

Q4. A binary search tree of integers can be represented in Scheme such that each node contains an integer value followed by left subtree followed by right subtree. For example: **[1 + 1 + 1 + 1 + 2 + 2 + 2 = 10 marks]**

(100 () ())	(100 (80 (70 () ()) ()) (120 () ()))
	

Given the following code in scheme to implement a binary search tree, fill in the blanks. You can use “left_subtree” and “right_subtree” functions to answer “insert” and “height” functions:

Output:

"Insert 100 -> (100 () ())"
"Insert 80 -> (100 (80 () ()) ())"
"Insert 70 -> (100 (80 (70 () ()) ()) ())"
"Insert 120 -> (100 (80 (70 () ()) ()) (120 () ()))"
"Insert 110 -> (100 (80 (70 () ()) ()) (120 (110 () ()) ()))"
"Insert 130 -> (100 (80 (70 () ()) ()) (120 (110 () ()) (130 () ())))"
"Is BST Empty -> #f"
"Height of BST -> 2"

(Fill in your answers in the code given on next page).

```
(define create_newnode (λ (node_value left_subtree right_subtree)

              ( ________________________________________________ )
))
(define node_value (λ (tree) (car tree)))

(define left_subtree (λ (tree) ( ____________________________________ )  ))

(define right_subtree (λ (tree) ( ___________________________________ )  ))

(define height_difference (λ (left_subtree right_subtree)

           ( _________________________________________________ )  ))

(define initialize (λ () '()))

(define insert
  (λ (tree value)
    (cond
      ((isempty? tree)
            (create_newnode value (initialize) (initialize)))
      ((< value (node_value tree))
            (create_newnode (  _____________________________________ )

                              ( ______________________________________ )

                              (  ______________________________________ ) )
      ((> value (node_value tree))

            (create_newnode (  _____________________________________ )

                              ( ______________________________________ )

                              (  ______________________________________ ) )

      (else tree))))
(define isempty? (λ (tree) (null? tree)))
(define height
  (λ (tree)
    (cond
          ((isempty? tree) -1)
          (else (cond
                    ( ___________________________________________________________

                    ___________________________________________________________ )
                    (else (+ 1 (height (right_subtree tree))))))))

(define sa (initialize))
(define sa1 (insert sa 100))
(format "Insert 100 -> ~s" sa1)
(define sa2 (insert sa1 80))
(format "Insert 80 -> ~s" sa2)
(define sa3 (insert sa2 70))
(format "Insert 70 -> ~s" sa3)
(define sa4 (insert sa3 120))
(format "Insert 120 -> ~s" sa4)
(define sa5 (insert sa4 110))
(format "Insert 110 -> ~s" sa5)
(define bst (insert sa5 130))
(format "Insert 130 -> ~s" bst)
(format "Is BST Empty -> ~s" (isempty? bst))
(format "Height of BST -> ~s" (height bst))
```