

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
HYDERABAD CAMPUS
FIRST SEMESTER 2018 – 2019
PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301)
COMPREHENSIVE EXAM

Date: 03.01.2018 Weightage: 35% [70M] Duration: 3 Hrs. Type: Closed Book

Please note: 1. All parts of the questions have to be answered consecutively.

2. Your Answers should be brief.

Q1. Features of a Programming Languages **[2+2+2+3=9M]**

A. In a language named Cool the variables are just strings of a's, b's, and c's. As programmer if you are asked to choose between C and Cool language which one would you prefer and why?

B. Under what circumstances does the below C-code snippet affect reliability of C-language?

```
int i;  
double d;  
...  
i = d;
```

C. "Pointers must be able to point to any type of variable or data structure." Which criteria of the readability feature does this statement effect and why?

D. You are asked to add a new feature in C-Language that checks for array-out-of bounds. What is the effect of this feature on compile time and run time?

Q2. Syntax

A. How many tokens would the following C-code generate?

[2+3+5+3=13M]

```
float fun(char *s) {  
    /* Find a zero */  
    if(!strcmp(s, '\0'))  
        return 0; }
```

B. Given the following regular expression $((xz)|(yz))^*$ answer questions i and ii.

- i. Describe in one sentence the strings accepted by the above regular expression.
- ii. Give an equivalent regular expression that accepts same strings as the above regular expression.

C. Give BNF grammar to accept structure definitions that are valid as per C language. For example

```
struct id {  
    member definition;  
    member definition;  
    ...  
    member definition;  
} id;
```

Note: You may assume that required terminals and nonterminals are already defined. For example, you need not define <TYPE> or id etc.

D. Using the BNF grammar you have given in question C draw a parse tree for the following input string

```
struct Books {  
    char author[50];  
    int book_id;  
} book;
```

Q3. Types and type checking

A. Given the following function declaration `void sum (int myarray[3][]).` What type of ordering does the language use to store the array elements in the memory? **[2 M]**

B. For the following declaration of a discriminated union draw the compile-time descriptor.

type Node (Tag : Boolean) is

[4 M]

record

case Tag is

when True => Count : Integer;

when False => Sum : Float;

end case;

end record;

C. Give an example of fixed heap dynamic array. What is fixed these type arrays?

[2 M]

Q4. Run time memory model

<p>Consider the following C-program and answer questions A-C</p> <pre> 1. int count; 2. char *cat(char *x, int i){ 3. static int r = 4; 4. x[3] = 'h'; 5. char *result = (char *)malloc(20); 6. sprintf(result, "%s x %d", x, i); 7. return result; 8. } 9. int main(){ 10. char *s = "my cat"; 11. int z = 12; 12. cat(s, z); 13. }</pre>	<p>A. Show the memory snapshot when the statement number 7 is executing.</p> <p>B. What is the scope of the variables s, x and r?</p> <p>C. What is the lifetime of the variables count, result and z?</p> <p>Note: For questions B and C you may just write in terms of the line numbers.</p> <p>[4+1.5+1.5 =7M]</p>
---	--

Q5. Procedure and parameter passing

[3+3+3=9M]

A. Compare and contrast static and dynamic scoping. Explain how identifier bindings are resolved at compile-time or run-time.

B. Give two reasons why very few programming languages prefer dynamic scoping.

C. Consider the following C like code where nesting of functions is allowed and IN OUT refers to Call by Value-Result and REF refers to call by reference. What values would be printed?

```

void outer (void) {
    int a = 5, b = 7;
    void inner (IN OUT int c; REF int d) {
        printf ("a: %d b: %d c: %d d: %d\n", a,b,c,d);
        a = 0; b = 9; c = 4; d = 6;
        printf ("a: %d b: %d c: %d d: %d\n", a,b,c,d);
    }
    inner(a,b);
    printf ("a: %d b: %d\n",a,b);
}
```

Q6. Garbage collection

A. A real-time program is one where program carries out various actions that must occur within a certain amount of time. For example, if a car's braking system were controlled by software, then if the software didn't do its job in time, you might crash.

[2+2=4M]

i. Why is an implementation that uses mark-and-sweep or stop-and-copy garbage collection not a good idea in a real-time program?

ii. Which garbage collection algorithm is suitable for this application? Justify your answer.

B. Given the following heap configuration as shown in Figure-1

[3 M]

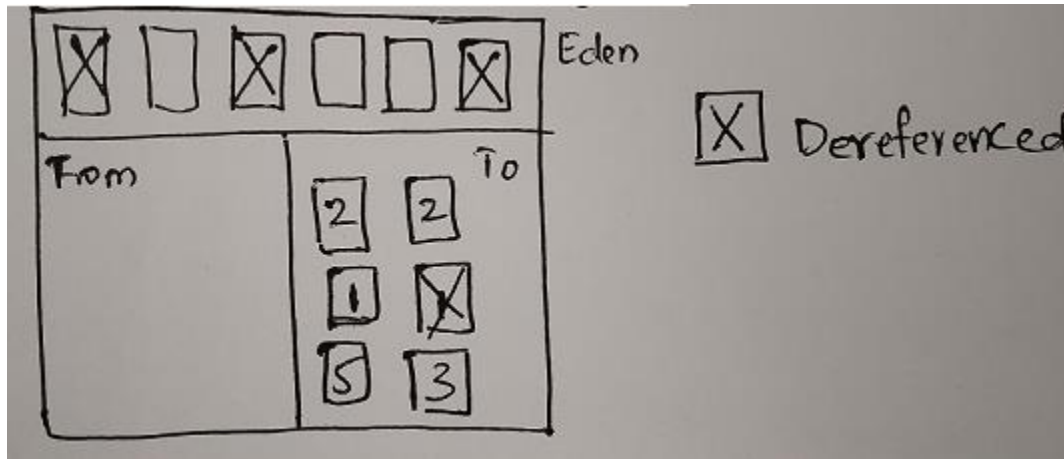


Figure-1

Show the heap configuration after one minor Garbage Collection of generational garbage collector.

Q7. Functional Programming

[4+5+4+4=17M]

A. Given the following BNF for lambda expressions

$\langle \text{expr} \rangle ::= \langle \text{constant} \rangle \mid \langle \text{variable} \rangle \mid (\langle \text{expr} \rangle \langle \text{expr} \rangle) \mid (\lambda \langle \text{variable} \rangle . \langle \text{expr} \rangle)$

Draw a parse tree for the input string $((\lambda x. (\text{add } x \ x)) \ 5)$

B. Perform β -reduction for the lambda expression $(\lambda x. (\text{add } x \ x)) (\text{add } 2 \ 3)$ using applicative(call by value) and normal order(call by name).

C. Write a scheme program to find the maximum of three numbers.

Note: Scheme has relational operator as in C-Language and its logical operators are and, or, not.

D. **Table-1** lists few concepts which exist in an imperative paradigm. Write the equivalent concept in functional paradigm.

Imperative Paradigm	Functional Paradigm
Assignment statement	
Function call	
Loop	
Global Variable	
Local Variable	
If Statement	
Function	
Arrays	

Table-1

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