

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

Third Year

First Semester 2024

Class Test II

Principles of Programming Language (Set II)

Time- Fifty Minutes

Full Marks-30

1. Discuss call-by-need concept. 2
2. (a) Write a program in Prolog (declaratively) to generate and print n Fibonacci numbers.  
(b) Show how it works for an input query.  
(c) How to improve the efficiency of such a construct? 5+3+2=10
3. Write a Prolog program to calculate the summation of n natural numbers. Show the DFS tree that gets generated for any input >2. 7

```
int arr[]={1,2,3};  
if(arr[3]==0)  
    return arr[2]/arr[3];  
else  
    return 0;
```

4. Represent above construct in lambda calculus. Derive any predicates, constructs and data types that you need. No need to define Church numerals, pair, 'true', 'false' and if\_then\_else construct. 8
5. Discuss the necessity for introducing types in lambda calculus. 3

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

Third Year

First Semester 2024

Class Test II

Principles of Programming Language (Set IV)

Time- Fifty Minutes

Full Marks-30

1. a. Given the following Prolog clauses:

```
ancestor(X, Y) :- ancestor(Z, Y), parent(X, Z).  
ancestor(X, X) :- !.  
parent(amy, bob).
```

Show the search tree to be generated for the query `ancestor(X, bob)`. Discuss the role of cut here. 8

2. Write a code in Prolog to implement (i) maximum of 3 numbers, (ii) generating a list by replicating a number  $n$ ,  $x$  times, (iii) constructing a list or merging two lists (show an example of each), (iv) finding the length of a list.  $(3*4)=12$

3. `sum=0; i=1;`

```
while(i<10) {  
    sum+=i;  
}
```

Represent above construct in lambda calculus. You can assume that *Church numerals, predecessor, addition, less than* predicates are in place. Justify your answer. 7

4. Identify the axioms from the following clauses.

```
natural(0).
```

```
natural(-1).
```

```
natural(X) :- natural(successor(X)).
```

3

**BACHELOR OF COMPUTER SCIENCE AND ENGINEERING**

**Third Year**

**First Semester 2024**

**Class Test II**

**Principles of Programming Language (Set I)**

**Time- Fifty Minutes**

**Full Marks-30**

1. a. Write a program in Prolog to compute gcd according to Euclid's algorithm. Compare it with the following method:

`gcd(U,V,W) :- not(V=0), R is U mod V, !, gcd(V,R,W).`      6

2. Write a procedural code for insertion sort. Write a Prolog program for insertion sort. Differentiate the two approaches. Show the computation steps of the Prolog program for the list [3,2,1].      3+6+2+3=14

3. Write the formulation for (i) tail recursive summation of n Church numerals using lambda calculus. You can assume the notations for Church numerals, and if-then-else. Show steps for an example. (ii) Does call by name support lazy evaluation? Discuss.      6+4

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

Third Year

First Semester 2024

Class Test II

Principles of Programming Language (Set III)

Time- Fifty Minutes

Full Marks-30

1. a. Given the following Prolog clauses:

```
ancestor(X,Y) :- parent(X, Z) , !, ancestor(Z, Y).  
ancestor(X, X) :- !.  
parent(amy, bob).
```

Show the search tree to be generated for the query `ancestor(X, bob)`. Discuss the role of cut here.

b. How are these clauses modified so that all the solutions could be found by Prolog. 8+2

2. `i=1;sum=0;`  
`while(i<10) {`  
    `sum+=i; }`  
`Average=sum/10;`

Represent above construct in lambda calculus. You can assume that *Church numerals, addition, boolean predicates are in place*. Show the steps for an example. 10

3. a. Write a program in Prolog that prints the reverse of a list. Explain its working with an example.  
b. Why is call-by-value favored by language designers when call-by-name is more terminating? 6+4