B.E. COMPUTER SCIENCE & ENGINEERING 3rd YEAR 1st SEMESTER EXAM- 2023

Principles of Programming Languages

Time: 3 hours

Full Marks: 100

Group A (Total Marks: 30) [CO1]
Answer any TWO questions

1.	Traders execute transactions. The two class structures are as follows. A collection of transactions is given.	3+2+3 +3+4= 15

Trader

private final String name;
private final String city;

public Trader(String n, String c);

public String getName();

public String getCity();

public String toString();

Transactions

private final Trader trader;

private final int year;

private final int value;

public Transaction(Trader trader, int year, int value);

public Trader getTrader();

public int getYear();

public int getValue();

public String toString();

You're asked by your manager to find answers to the following queries.
(a) Group traders from different cities who performed high-value transactions.
(b) Find the transaction with the highest value for each year.
(c) Find the average transaction value for each year.
(d) What is the minimum transaction made by any trader from Mumbai?
(e) From a list of words, extract the list of words that ends with a number and then print the duplicates among them.

2.	*Artist class denotes an* individual or group who creates music having the following fields.	4+3+6 +2=15

- *name*: The name of the artist (individual or group name)
- *members*: A set of other artists who comprise this group - this field might be empty
- *origin*: The primary location of origin of the group (e.g., "Kolkata").

(a) Convert the following code sample from using external iteration to internal iteration:

```
int totalMembers = 0;
for (Artist artist : artists) {
if(artist.getOrigin()=="Kolkata") {
    Stream<Artist> members = artist.getMembers();
    totalMembers += members.count();}}
```

(b)Find the bands with most members using lambda expressions and/or streams API.
(c) max(), average() and count() using only reduce and lambda expressions. Can it handle empty list?
(d) Is this lambda expression side-effect free or does it mutate state?

$$x \rightarrow x+1$$

3.	Given a text file, answer (a) and (b) (a) Identify and list the distinct letters; (b) Group it's words into three categories depending on word length-2-letter words, 3-letter words and more than 3 letter words. (c) Create a collection of *n* Tribonacci numbers using java streams API. The number series looks like 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81,...	4+6+5 =15

Group B (Total Marks: 20) [CO2]
Answer any ONE question from this group.

4.	(a) Write code in Prolog to implement (i) maximum of 3 numbers, (ii) generating a list by replicating a number *n*, *x* times, (iii) finding the last element of a list, (iv) prepending an element to a list.
	
	(b) Write a program in Prolog that takes a list as input and sorts the list. Show how it works.
	
	(c) Given the following Prolog clauses:
	`ancestor(X, X).`
	`ancestor(X,Y) :- ancestor(Z, Y), parent(X, Z).`
	`parent(amy, bob).`
	Explain Prolog's response to the query `ancestor(amy, Y)` using a search tree of subgoals.
		(3x3+1)
		+6+4=
		20

5.	(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(amy,X)`. Discuss the role of cut here.
	(b) Write Prolog code to compute factorial of a number with and without using accumulator. Compare the two approaches.
	
	(c) Explain the difference in Prolog between the following two definitions of the sibling relationship. Which definition is better? Give reasons.
	
	`sibling1(X,Y) :- not(X=Y), parent(Z,X), parent(Z,Y).`
	
	`sibling2(X,Y) :- parent(Z,X), parent(Z,Y), not(X=Y).`
	
	(d) Write Prolog clauses to print the numbers that are not part of the Fibonacci series.
		7+6+4+
		3=20

Group C (Total Marks: 30) [CO3]
Answer any TWO questions.

6.	(a) State two theorems of Church-Rosser.
	
	 (b) `if(roll_no%2!=0)`
	         `return roll_no;`
	   `else`
	         `return "Even Number";`
	 Represent above construct in lambda calculus. Derive any predicates, constructs and data types that you need. Do not use Y-combinator.
	
	 (c) Define pair in Lambda calculus. How would you select *one element* from a *pair* using Lambda calculus? Justify.
		4+8+3= 15

7.	(a) Derive the Boolean operator OR in lambda calculus and validate its truth table. No need to define '*true*', '*false*'.
	 (b) `int arr[]={1,2,3};`
	     `if(arr[2]==0)`
	         `return arr[2]+1;`
	     `else`
	         `return arr[2];`
	 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.
	 (c) Use applicative order reduction to reduce the lambda expression.
	 $(\lambda x.y)((\lambda z.zz)(\lambda w.w))$
		4+9+2= 15

8.	(a) Reduce the following lambda expression using both normal order and applicative order reduction. $(\lambda x . x\,x\,x)(\lambda x . x\,x\,x)$
	
	 (b) `while(i<10) {`
	         `product*=a[i];`
	         `i++;`
	   `}`
	   `Average=product/10;`
	
	 Represent above construct in lambda calculus. Here, a[i] represents the collection of natural numbers. You can assume that *Church numerals*, *predecessor, and multiplication* predicates are in place. Justify your answer.
		4+11= 15

Group D (Total Marks: 10) [CO4]
Answer any ONE question.

9.	(a) What is the advantage of following value semantics while designing parallel programming? (b) Which of these lambda expressions are valid `Function<Long,Long>` implementations? (i) `x -> x + 1;` (ii) `(x, y) -> x + 1;` (iii) `x -> x == 1;` (c) Define functional interface. Is `java.io.Closeable` a functional interface?	4+3+3 = 10

10.	(a) Discuss the applicability of call by name and call by value semantics of Lambda calculus to modern programming languages. Give suitable examples. (b) What is the implication of call by need construct? State one advantage and one disadvantage of it. (c) Which semantics do higher order functions follow? Give reasons.	4+4+2 =10

Group E (Total Marks: 10) [CO5 and CO6]
Answer any ONE question.

11.	(a) Compare between dynamic dispatch and double dispatch. Give an example. (b)Discuss the bottlenecks of imperative programming paradigm. Name one paradigm where these can be overcome. Explain how this is done.	5+5= 10
12.	(a) Define regularity and the aspects of it. (b) Discuss structured abstractions as implemented by programming languages. You may use code snippets.	6+4=10