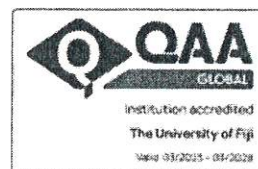


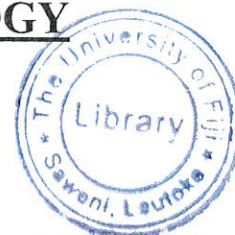


The University of Fiji
(An Entity of Arya Pratinidhi Sabha of Fiji)



SCHOOL OF SCIENCE AND TECHNOLOGY
Semester 1, 2025

FINAL EXAMINATION PAPER



ITC215: DATA STRUCTURES AND ALGORITHMS

Important Instructions to Students:

1. Read the questions, instructions carefully and answer the questions.
2. You are allowed an **extra 10 minutes reading time** (during this time, you are not allowed to write with pen) and **180 minutes (3 Hours) to complete this paper.**
3. Silent, non-programmable calculators are allowed, but not supplied.
4. Write all your answers in the answer booklet provided. If you need extra sheets, please ask from the supervisor. Write your name, course code and student identification number in the answer booklet and in the attendance sheet.
5. This paper contains two (2) sections A and B. All questions are compulsory to answer.
6. Write all your answers legibly. The marker has the right to award zero for answers that are not legible.
7. This is a **Closed Book Examination**. The University views plagiarism, cheating and dishonest practice as serious offences and if found, offenders are penalized.

Examination Paper Content (Format):

<u>Section</u>	<u>Question</u>	<u>Content</u>	<u>Marks</u>	<u>Suggested Time</u>
A	1-7	Short Answer Questions	90	162 minutes
B	1	Long Answer Questions	10	18 minutes
		TOTAL	100	180 Minutes

*Registered with the Fiji Higher Education Commission as a University under the Higher Education Act 2008.
Registration Certificate Number (RGN0020/11)*

All questions within this section are compulsory.

Question 1: Java Primer [13 Marks]

- A. Write a Java program that calculates the area of a rectangle. Allow the users to enter the values for Width and Height, and show the calculated area as an output. Your program structure should be in this context: **[7 Marks]**

Enter Width: 10

Enter Height: 10

The total Area is: 100

- B. Write a Java program that consists of classes and methods that will help display the following output: **[6 Marks]**

Enter Speed: 200

The car is going as fast as it can!

Max Speed: 200

Question 2: Arrays [5 Marks]

- A. Study the code below and explain line by line how the code will execute:

```
import java.util.*;
class OnedimensionalScanner
{
    public static void main(String args[])
    {
        int len;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Array length : ");
        len=sc.nextInt();
        int a[]=new int[len];//declaration
        System.out.print("Enter" + len + "Element to Store in Array :\n");
        for(int i=0; i<len; i++)
        {
            a[i] = sc.nextInt();
        }
        System.out.print("Elements in Array are :\n");
        for(int i=0; i<len; i++)
        {
            System.out.print(a[i] + " ");
        }
    }
}
```


Question 3: Algorithm Analysis and Recursion [17 Marks]

- A. How do you typically test the runtime of an algorithm? **[1 Mark]**
- B. To assess the efficiency of an algorithm, it's crucial to first understand the primitive operations involved. Discuss the primitive operations considered when analyzing the performance of complex algorithms. **[2 Marks]**
- C. List the seven functions used for algorithm analysis, along with their respective formulas. **[3.5 Marks]**
- D. Discuss the five rules of logarithms. **[2.5 Marks]**
- E. What does Recursion mean in Java? **[1 Mark]**
- F. Write a recursive function that prints the numbers 1...n in descending order: **[2 Marks]**

```
public void descending (int n)
{
}
}
```

- G. Study the code below. Identify what the output would be and explain your thought process. **[5 Marks]**

```
public class RecursionExample {
    public static void main(String[] args) {
        int number = 10;
        int result = factorial(number);
        System.out.println("Factorial of " + number + " is: " + result);
    }
    // Recursive factorial method
    public static int factorial(int n) {
        // Base case: when n is 0 or 1, return 1
        if (n == 0 || n == 1) {
            return 1;
        } else {
            // Recursive case: n * factorial of (n-1)
            return n * factorial(n - 1);
        }
    }
}
```


Question 4: Stacks, Queues, and Dequeues [14 Marks]

- A. Suppose an initially empty stack S has performed a total of 40 push operations, 15 top operations, and 12 pop operations, 5 of which returned null to indicate an empty stack. What is the current size of S? **[1 Mark]**
- B. Suppose an initially empty queue Q has performed a total of 32 enqueue operations, 10 first operations, and 15 dequeue operations, 5 of which returned null to indicate an empty queue. What is the current size of Q? **[1 Mark]**
- C. For the following Stack S of Integers, fill out the return value and the stack contents. **[4 Marks]**

Method	Return Value	Stack Contents
push(3)		
push(9)		
push(7)		
pop()		
push(2)		
top()		
size()		
pop()		
pop()		
isEmpty()		
push(6)		
push(4)		
top()		
pop()		
size()		

- D. For the following Queue, determine the return value and contents of Q. **[4 Marks]**

Method	Return Value	First $\leftarrow$ Q $\leftarrow$ Last
enqueue(5)		
enqueue(8)		
dequeue()		
enqueue(2)		
enqueue(7)		
size()		
first()		
dequeue()		
dequeue()		
isEmpty()		
enqueue(4)		
enqueue(6)		
first()		
enqueue(9)		
dequeue()		

E. For the following Deque, determine the return value and the output content for D. [4 Marks]

Method	Return Value	D
addFirst(7)		
addLast(5)		
addFirst(2)		
first()		
last()		
removeLast()		
first()		
removeFirst()		
isEmpty()		
size()		
addLast(8)		
addFirst(1)		
last()		
removeLast()		
addLast(4)		

Question 5: List ADT and Positional List [14 Marks]

A. Differentiate between an Array, Dynamic Arrays, and ArrayList with appropriate examples. [3 Marks]

B. Refer to the below code and explain what the output will be: [3 Marks]

```
List<Integer> num = new ArrayList<Integer>();
num.add(0, 45);
num.add(1, 12);
num.add(2, 89);
System.out.println(num);
```

```
List<Integer> val = new ArrayList<Integer>();
val.add(33);
val.add(74);
val.add(28);
val.add(91);
val.add(60);
val.add(17);
```

```
num.addAll(1, val);
System.out.println(num);
```

C. For the following List, determine the return value and the output for List content. [4 Marks]

Method	Return Value	List Contents
add(0, "B")		
add(1, "D")		
add(2, "G")		

add(3, "L")		
get(2)		
remove(0)		
remove(2)		
add(2, "T")		
set(1, "Z")		
set(1, "Z")		
remove(1)		
get(1)		
isEmpty()		
add(2, "X")		
set(2, "Y")		

D. For the following Positional List, determine the return value and the output for List content.
[4 Marks]

Method	Return Value	List Contents
addLast(6)		
first()		
addAfter(p, 3)		
after(p)		
addBefore(q, 1)		
r.getElement()		
before(q)		
addFirst(5)		
remove(last())		
set(p, 9)		
remove(q)		
size()		
isEmpty()		
addFirst(7)		
addLast(4)		

Question 6: Priority Queues and Maps [8 Marks]

A. For the following Priority Queue, determine the return value and the output for the Queue.
[4 Marks]

Method	Return Value	Queue
insert(30, "L")		
insert(10, "M")		
insert(7, "B")		
min()		
removeMin()		
insert(42, "T")		
insert(2, "D")		
size()		
removeMin()		

insert(6, "K")		
insert(51, "R")		
insert(4, "S")		
removeMin()		
removeMin()		
isEmpty()		

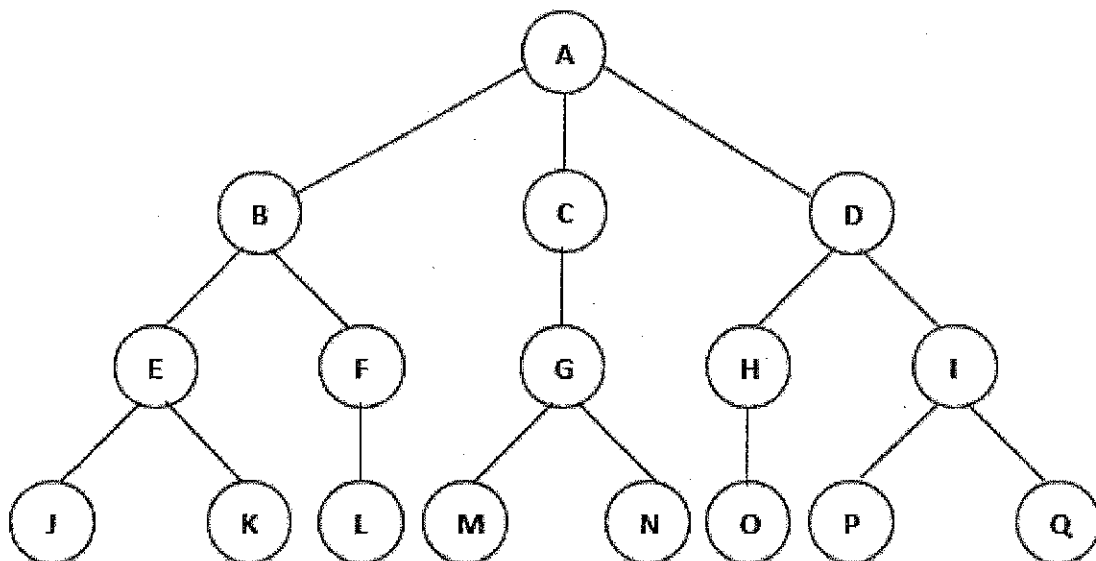
B. For the following Map, determine the return value and the output for Map. **[4 Marks]**

Method	Return Value	Map
put(4, "X")		
put(9, "Y")		
put(1, "Z")		
put(4, "W")		
get(9)		
get(4)		
size()		
remove(9)		
remove(1)		
get(1)		
remove(1)		
isEmpty()		
entrySet()		
values()		
keySet()		

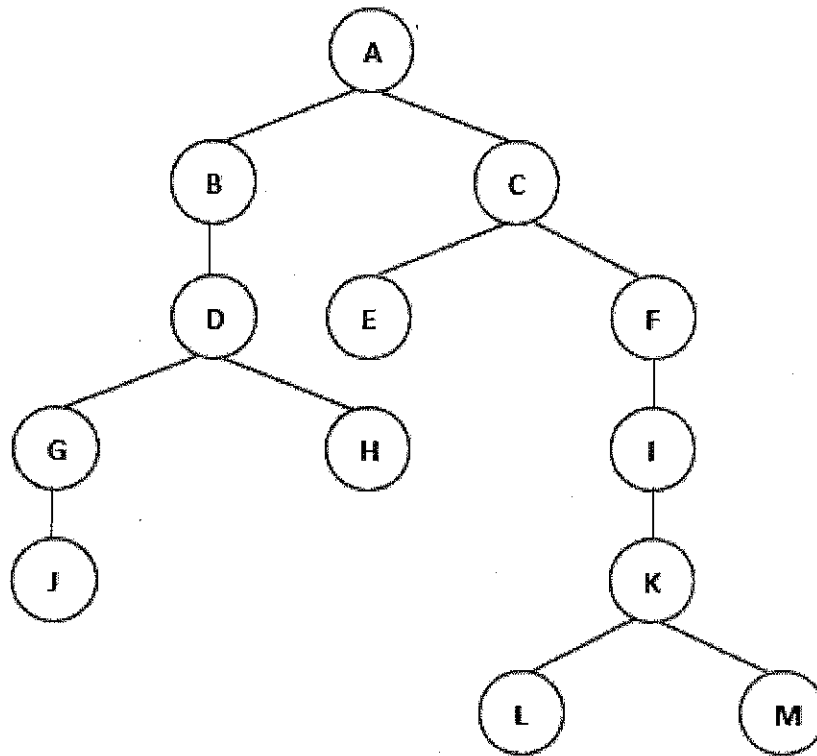
Question 7: Trees [19 Marks]

A. For the following trees, perform In-Order, Pre-Order, and Post-Order traversals. **[6 Marks]**

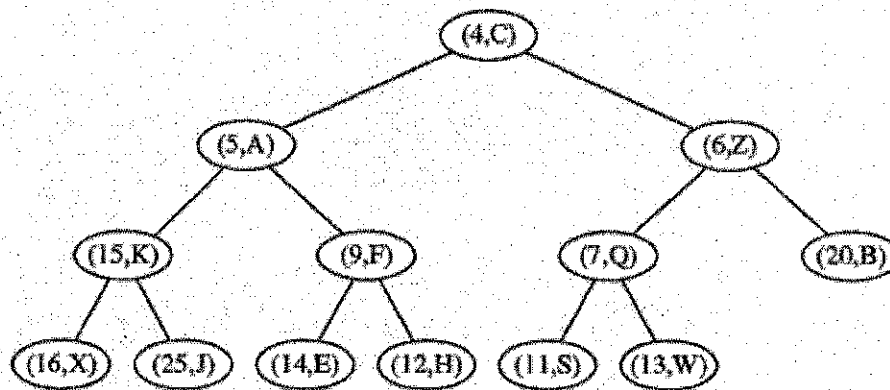
I.



II.



B. For the following Tree, perform Heap Insertion and Heap Deletion. [4 Marks]



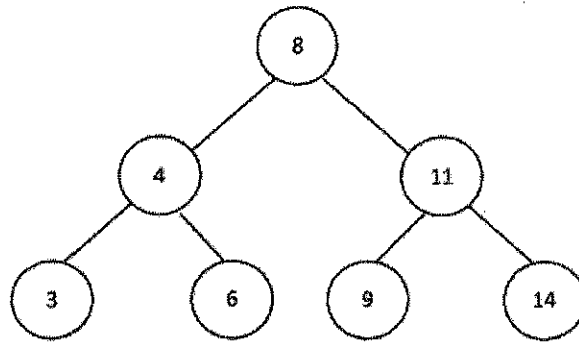
I. Insert (8, M)

II. Delete (25, J)

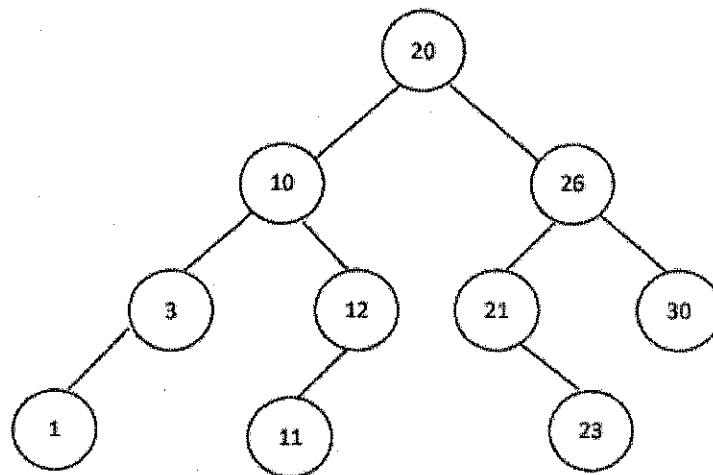
C. Write Java codes to create a tree structure, constructor, and its respective nodes from the below illustration. Include codes for InOrder, PreOrder, and PostOrder Traversals to present outputs.

[9 Marks]

I.



II.



SECTION B: LONG ANSWER QUESTIONS

[10 MARKS]

There is only 1 question within this section.

Objective:

Design and implement a console-based Java application that calculates the area of different shapes using separate methods for each calculation. The program should prompt the user to select a shape, take the necessary inputs, and display the computed area.

Main Program:

Display a menu allowing the user to select one of the following shapes:

- Circle
- Rectangle
- Triangle
- Square

Based on the user's selection, call the respective method to perform the area calculation.

Methods for Calculation:

Implement separate methods for each shape's area calculation:

- calculateCircleArea() → Computes the area of a circle using $A = \pi r^2$
- calculateRectangleArea() → Computes the area of a rectangle using length × width.
- calculateTriangleArea() → Computes the area of a triangle using (base × height) / 2.
- calculateSquareArea() → Computes the area of a square using length².

Each method should prompt for input, calculate the area, and display the result.

User Input Handling:

Use a Scanner object to take user input for the necessary shape dimensions.

If the user enters an invalid choice, display an error message and exit.

```
Select a shape to calculate its area:
1. Circle
2. Rectangle
3. Triangle
4. Square
Enter your choice: 2

Enter the length: 8
Enter the width: 5

The area of the rectangle is: 40.0
```

You are required to code for the entire application.

THE END

