

**FINAL EXAMINATION****DIPLOMA IN COMPUTER SCIENCE****COURSE : DATA STRUCTURES****COURSE CODE : SWC3684/SWC3344****DURATION : 2 HOURS AND 30 MINUTES****INSTRUCTIONS TO CANDIDATES :**

1. This question paper consists of **FOUR (4)** questions.
2. Answer ALL questions in the Answer Booklet provided.
3. Please check to make sure that this examination pack consists of:
  - i. The Question Paper
  - ii. An Answer Booklet
4. Do not bring any material into the examination hall. An electronic calculator is allowed.
5. Please write your answer using permanent ink.

**MYKAD/ PASSPORT NO. :** \_\_\_\_\_**ID. NO. :** \_\_\_\_\_**LECTURER :** \_\_\_\_\_**SECTION :** \_\_\_\_\_

---

**DO NOT OPEN THIS QUESTION PAPER UNTIL YOU ARE TOLD TO DO SO**

---

*This question paper consists of 6 printed pages including the front page*

## QUESTION 1

- a. Define what an algorithm is. (2 marks)
  - b. List **SIX (6)** properties that a valid algorithm must possess. (6 marks)
  - c. State the **TWO (2)** critical members of a class in Java. (2 marks)
  - d. Name the **TWO (2)** primary resource constraints used to measure the efficiency of a data structure solution. (2 marks)
  - e. Identify **FOUR (4)** advantages of using the Java Collection Framework (JCF). (8 marks)
- (Total : 20 marks)

## QUESTION 2

- a. Define the term Linked List. (2 marks)
- b. State the **TWO (2)** structural components contained within every node of a basic linked list. (2 marks)
- c. List **TWO (2)** types of linked lists used in data structures. (2 marks)
- d. You are given a partial program with a `LinkedList` named `patients` of type `String`. Write the Java statements for each of the following tasks:

```
import java.util.LinkedList;

public class EmergencyRoomSystem {
    public static void main(String[] args) {

        LinkedList<String> patients = new LinkedList<>();

        // Write your code here

    }
}
```

- i. Add **FOUR (4)** patient names into the `patients` linked list. (2 marks)
  - ii. Print the original contents of the `patients` linked list. (2 marks)
  - iii. Retrieve and display the last patient currently waiting at the end of the list using an appropriate method. (2 marks)
  - iv. Discharge the patient from the very front of the list. (1 mark)
  - v. Check the total count of patients remaining in the list using an appropriate method and display the result. (2 marks)
  - vi. Print the final state of the `patients` linked list after all operations are completed. (2 marks)
- e. Based on **QUESTION 2.d.vi**, illustrate a linked list representation diagram showing the content of the `patients` linked list after the program has completed execution. (3 marks)
- (Total : 20 marks)

**QUESTION 3**

- a. Identify the specific operations used to perform each of the following tasks on a stack:
- i. Inserting a new data item onto the top of the stack. (1 mark)
  - ii. Removing an entry from the top of the stack. (1 mark)
  - iii. Inspecting the top entry of the stack without removing it. (1 mark)

- b. Consider the following mathematical infix expression:

$$(14 - 4) * (2 + 18 / 3)$$

- i. Convert the given infix expression into its equivalent prefix expression. Clearly show all your intermediate working steps.  
(6 marks)
- ii. Based on your answer in **QUESTION 3.b.i.**, find the final numerical result. Show the step-by-step evaluation process using a tracking stack.  
(6 marks)

- c. Give the output of the given Java program below:

```
import java.util.*;

public class ITSupportTicketSystem {
    public static void main(String[] args) {

        Queue<String> ticketQueue = new LinkedList<>();

        ticketQueue.offer("Ticket #101");
        System.out.println("After offering Ticket #101: " +
ticketQueue);

        ticketQueue.add("Ticket #102");
        System.out.println("After adding Ticket #102: " +
ticketQueue);

        String nextTicket = ticketQueue.element();
        System.out.println("Next ticket to be processed: " +
nextTicket);

        String resolvedTicket1 = ticketQueue.remove();
        System.out.println("After first removal: " +
ticketQueue);
        System.out.println("Resolved ticket: " +
resolvedTicket1);

        ticketQueue.add("Ticket #103");
        ticketQueue.add("Ticket #104");
        ticketQueue.add("Ticket #105");
        System.out.println("After adding tickets: " +
ticketQueue);

        String resolvedTicket2 = ticketQueue.remove();
        System.out.println("After second removal: " +
ticketQueue);
    }
}
```

```
String frontTicket = ticketQueue.peek();
System.out.println("Ticket at the front: " +
frontTicket);

String resolvedTicket3 = ticketQueue.poll();
System.out.println("After polling: " + ticketQueue);
System.out.println("Resolved ticket: " +
resolvedTicket3);

System.out.println("Final tickets in the queue: " +
ticketQueue);
}
```

(15 marks)

(Total : 30 marks)

**QUESTION 4**

- a. Consider the following mathematical infix expression:

$$(A + B) * (C - D / E)$$

- i. Illustrate the corresponding expression tree for the infix expression provided above.  
(5 marks)
  - ii. Based on your answer from **QUESTION 4.a.i.**, give its postorder traversal sequence.  
(2 marks)
- b. A collection of integer data elements is received in the following chronological sequence:

55 30 72 40 20 65 80 35 99

- i. Illustrate the final layout of the Binary Search Tree (BST).  
(9 marks)
- ii. Based on your answer from **QUESTION 4.b.i.**, find its preorder traversal sequence.  
(2 marks)
- iii. Based on your answer from **QUESTION 4.b.i.**, delete the value 30. Illustrate the final BST.  
(2 marks)

- c. State **TWO (2)** types of sorting algorithms that can be utilized to arrange data in a specific order.

(2 marks)

- d. Illustrate the selection sort algorithm for the following numbers:

**48 281 74 30 87 99 85 62 56**

(8 marks)

(Total : 30 marks)

**(TOTAL :100 MARKS)**

**END OF QUESTION PAPER**