



UNIVERSITI TEKNOLOGI MARA
FINAL EXAMINATION

COURSE	:	FUNDAMENTALS OF DATA STRUCTURES
COURSE CODE	:	CSC248
EXAMINATION	:	DECEMBER 2019
TIME	:	3 HOURS

INSTRUCTIONS TO CANDIDATES

1. This question paper consists of two (2) parts :
PART A (10 Questions)
PART B (7 Questions)
2. Answer ALL questions in the Answer Booklet. Start each answer on a new page.
3. Do not bring any material into the examination room unless permission is given by the invigilator.
4. Please check to make sure that this examination pack consists of :
 - i) the Question Paper
 - ii) an Answer Booklet – provided by the Faculty
5. Answer ALL questions in English.

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO

This examination paper consists of 10 printed pages

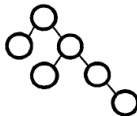
PART A

1. Which of the following is **NOT** a primitive data type in Java?
 - a) int
 - b) char
 - c) boolean
 - d) String
2. The _____ is an organization of data of the same or different types, usually in computer memory, for better algorithm efficiency during the program execution.
 - a) repetition structure
 - b) data structure
 - c) selection structure
 - d) sequential structure
3. Which of the following is **NOT** a method for the `ArrayList` class?
 - a) `add(int index)`
 - b) `get(int index)`
 - c) `remove(int index)`
 - d) `delete(Object o)`
4. What is the output of the following code segment?

```
ArrayList a = new ArrayList(10);  
System.out.println(a.size());
```

 - a) 0
 - b) 1
 - c) 10
 - d) 100
5. A `LinkedList` data structure is constructed by a group of list nodes. A list node contains of a data and a/an _____.
 - a) address of the next `LinkedList`
 - b) element
 - c) reference to the next node
 - d) string

6. What will happen to a LinkedList when the only list node in the data structure is removed?
- a) The size of the LinkedList become null.
 - b) The head will be assigned with null value.
 - c) The tail and the head of the LinkedList will contain different values.
 - d) The reference variable that represent the LinkedList will become null.
7. Which of the following scenario does **NOT** represent a Queue?
- a) A pile of clothes on the bed waiting to be folded.
 - b) A row of students waiting to buy food at the canteen.
 - c) Cars in line waiting for the traffic lights.
 - d) Processes waiting to be executed by the operating system in a computer.
8. What is the advantage of using a stack over a queue?
- a) The implementation of a stack is easier than a queue.
 - b) A stack can store more element than a queue.
 - c) The time to add a new item into a stack is faster than a queue.
 - d) The time to remove the last item stored in the stack is faster than a queue.
9. A tree is a hierarchical data structure. A tree node with no children is known as _____.
- a) branch
 - b) edge
 - c) leaf
 - d) root
10. The height of the tree in the following figure is _____.



- a) one
- b) two
- c) three
- d) four

(20 MARKS)

PART B**QUESTION 1**

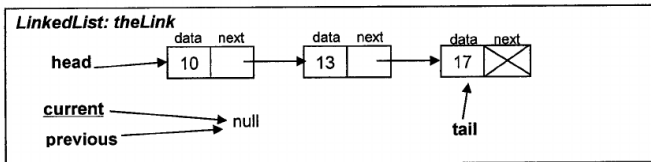
Write a Java program that could perform the following:

- a) Using `Integer` as a generic class, declare an object named `markList` from the `java.util.ArrayList` class. (2 marks)
- b) Using a loop, prompt the user to enter **FOUR (4)** integer numbers and store them into the `markList` data structure. (2.5 marks)
- c) Display the second integer number in the `markList` data structure. (2.5 marks)
- d) Change the value of the last integer number in the `markList` data structure to 100. (3 marks)
- e) Remove the first element in the `markList` data structure. (2 marks)

(12 Marks)

QUESTION 2

Given the following diagram (note that *head*, *tail*, *current* and *previous* are the reference variables):



- a) Redraw the **theLink** list above after the following program segment is executed.

```

node = new Node <String> ("D");
node.next = head;
head = node;
node = new Node <String> ("S");
node.next = head.next;
head.next = node;

```

(3 marks)

- b) Based on the **theLink** list above, write a program segment to remove the current tail, and assign the second last node as the **theLink** tail.

(4 marks)

- c) Refer to the following program fragment, if *head* is a reference to the first node and *xxx(head)* is being called, as shown in the program below, what is/are output produced by the method?

```

public static void main(String[] args) {
    //... some other program declarations and statements
    head = tail = current = null;
    //... some other program statements
    current = head;
    while (current != null) {
        xxx(current); //method call
        current = current.next;
    }
}

static void xxx(Node current) {
    if(current.next == null) return;
    System.out.print(current.next.data + " ");
}

```

(3 marks)

QUESTION 3

Given the following `Book` class and `BookshopLinkedList` class:

```
public class Book {
    private String title;
    private String author;
    private int pubYear;

    //normal constructor
    //accessors: getTitle(); getAuthor(); getPubYear();
    //printer: toString();
}

public class Node {
    Book data;
    Node next;

    //constructor
}

public class BookshopLinkedList{
    private Node first, last, current;
    public BookshopList() {...}           //Default constructor
    public void addFirst(Book b) {...}
    public Book getFirst() {...}
    public Book getNext() {...}
}
```

Assume that many `Book` objects has been created and added to the `bList1`, an instance of the `BookshopLinkedList` class.

- a) Write a complete definition of the `addFirst()` method. This method will take a `Book` object as parameter and store it into a new node in front of the Linked List.
(3 marks)
- b) Count and display the total number of `Book` objects in the `bList1` that are published in the year 2019.
(3 marks)
- c) Search and display the title of the first book (smallest value of `pubYear`) written by "James Gosling" in the `bList1`.
(4 marks)

(10 Marks)

↓ Document continues below

Discover more from:

Fundamentals Of Data Structures

CSC248

163 documents



Go to course



Chapter 1 Sorting & Searching Coding

4

Fundamentals Of Data Structures

100% (3)



Lab Test Question - -----

4

Fundamentals Of Data Structures

100% (2)



Lab exercise week 8 (Queue) Oct 2020

3

Fundamentals Of Data Structures

100% (2)



CSC248 Project Group Proposal

3

Fundamentals Of Data Structures

100% (1)



Test1 CSC248 Oct2018 Suggested Answer

10

Fundamentals Of Data Structures

100% (1)



Chap 1 Bubble Sort

18

Fundamentals Of Data Structures

100% (1)

QUESTION 4

A Queue data structures is a collection of items in which only the earliest added item may be accessed. Assume the following process has no syntax and can be executed successfully until the end of the program execution.

```
class Queue {
    public Queue() {...}           //default constructor
    public enqueue(Object obj) {...} //insert new value to the
Queue
    public Object dequeue() {...}   //remove value from the Queue
    public boolean isEmpty() {...}
    public String toString() {...}
}

public class testQ {
    public static void main (String[] args) {
        Queue q1 = new Queue();
        Queue q2 = new Queue();
        Queue q3 = new Queue();

        for (int n=0; n<10; n++) {
            if (n % 2 == 1)
                q1.enqueue(n);
            else
                q2.enqueue(n + 3);
        }

        //Line AA

        //Line BB. Add code statements here.
        System.out.println (q3.toString());
    }
}
```

- a) Draw diagrams of q1 and q2 to represent the content in these data structures at **Line AA**. Please label the front and the rear of the data structures. (4 marks)
- b) Write the missing code statements at **Line BB**. The code must retrieve and add the values from q1 and q2 accordingly, then store the answers into q3. (6 marks)

(10 Marks)

QUESTION 5

A Stack is one of the basic data structures where insertion and deletion of items takes place at one end called top of the stack. Assume the following code has no syntax error and can be executed correctly.

```
class Stack {
    public Stack() {...}           //default constructor
    public push(Object obj) {...}  //insert new value to the Stack
    public Object pop() {...}      //remove value from the Stack
    public boolean isEmpty() {...}
    public String toString() {...}
}

public class testStack {
    public static void main (String[] args) {
        Stack stk1 = new Stack();
        Stack stk2 = new Stack();
        Stack temp = new Stack();

        int count = 0;
        for (int n=100; n<110; n++) {
            stk1.push(n + count);
            count = count + 1;
            if (count >= 3)
                count = 1;
        }

        //Line XX

        stk2.push('a'); stk2.push('b'); stk2.push('c');
        stk2.push('d'); stk2.push('e'); stk2.push('f');

        //Line YY. Add code statements here.

        System.out.println (stk2.toString());
    }
}
```

- Draw a diagram of `stk1` to represent the content in these data structures at **Line XX**. Label the data structures properly. (4 marks)
- Write the missing code statements at **Line YY**. The output of the program from the statement `System.out.println(stk2.toString());` has to be: (8 Marks)

[f, e, d, c, b, a]

(4 marks)

(8 Marks)

QUESTION 6

a) Convert the following expression into postfix notation.

- i) $A + B - C * D$
- ii) $E * (F - G * H)$

(3 marks)

b) Convert the following expression into prefix notation.

- i) $J + K * L - M$
- ii) $P * Q / R \$ T$

(3 marks)

c) Given the following mathematical expression, solve the equation and show all the steps by using stack configuration. You must convert the expression into postfix notation first.

$$(1 + 2) * 4 / 3$$

(4 marks)

(10 Marks)

QUESTION 7

a) Given an arithmetic expression as follows:

$$P + Q - (R + S) * (T / U)$$

- i) Draw the expression tree of the given arithmetic expression.

(4 marks)

- ii) What is the level of the node '*' in question (i)?

(1 mark)

- iii) List the external nodes of the tree in question (i).

(1 mark)

- iv) What is/are the ancestors of the tree node 'R'?

(1 mark)

- v) Write the sequence of the node visited when we perform the **preorder** and **postorder** traversal on the tree drawn in Question 7 (a) (i) above.

(3 marks)

- b) Given the ADTs of Appliances, TreeNode, and BSTAppliances:

```

public class Appliances {
    private String name; //fan, microwave, blender
    private int watt;     //power consumption: 1000, 1500, 1600
    //constructors
    //accessors, mutators
    //printer methods
}

public class TreeNode {
    Appliances data;
    TreeNode left, right;
    /** Definition of the other methods */
}

public class BSTAppliances {
    TreeNode root;
    //constructors
    //accessors, mutators
    public int countHiPowerApp() {...}
    public void showWatt(String name) {...}
    //printer methods
}

```

- i) Draw the Binary Search Tree (BST) diagram that stores the data given in the following table. Arrange the node in the BST based on the power consumption.

Appliance Name	Power Consumption (watt)
Air Conditioner	3250
Ceiling Fan	120
Gaming Desktop Computer	500
Microwave	1000
Plasma TV	200

(2 marks)

- ii) Write the definition of the `countHiPowerApp()` method and its recursive method to count and return the number of appliances that has power consumption that is more than 1000 watt.

(4 marks)

- iii) Write the definition of the `showWatt(String name)` method and its recursive method to display the power consumption of the appliances that match with the name specified in the parameter.

(4 marks)

(20 Marks)

END OF QUESTION PAPER