© Vocational Training Council

IVE Information Technology

Information & Communications Technology

Programme Board

Instructions:

(a) This paper has a total of TWELVE pages including the covering page.

(b) This paper contains TWO Sections.

(c) Section A is WORTH 40 marks and Section B is WORTH 60 marks.

(d) Section A contains THREE questions. Answer ALL questions in Section A.

(e) Section B contains THREE questions. Answer ALL questions in Section B.

Note: The result of this assessment will not be counted if you do not meet the minimum attendance requirement (if any) governed by the general academic regulations of your programme/course unless approval of the campus principal has been granted.

HIGHER DIPLOMA IN

SOFTWARE ENGINEERING

(IT114105)

MODULE TITLE:

DATA STRUCTURES & ALGORITHMS: CONCEPTS AND IMPLEMENTATION

MODULE CODE: **ITP4510**

SEMESTER TWO MAIN EXAMINATION

16th May, 2017

1:30 PM TO 3:30 PM (2 hours)

© Vocational Training Council

This paper contains TWO sections.

Section A (40 marks)

This section contains 3 questions.

Answer ALL questions.

A1 Consider the following program segments:

```
interface Moveable {
  // With a public abstract method called move with two
  // integer parameters called direction and distance,
  // the return type of move is void.
       ...
}

public abstract class Animal implements Moveable {
  // two private variables:
  // name (type String)and weight(type int)
       ...

  // a public abstract method call talk with no
  // parameter, the return type of talk is void.
       ...
}

class Car implements Movable {
    ...
}

class Bird extends Animal {
    ...
}

class Fish extends Animal {
    ...
}

class Tiger extends Animal {
    ...
}
```

© Vocational Training Council

(a) List the method(s) that MUST be written in class "Car" and class "Bird". [3 marks]

(b) Write the "Moveable" interface in JAVA code. [2 marks]

(c) Write the "Animal" abstract class in JAVA code. [4 marks]

(d) Three objects are created as follow:

```
Car    c = new Car();
Bird   b = new Bird();
Tiger  t = new Tiger();
```

Write ONE JAVA statement to create an array which can store the above 3 objects. [3 marks]

(e) Can the following code be compiled? Explain your answer. [2 marks]

```
Fish f = new Animal();
```

© Vocational Training Council

A2 Consider the following Java program.

```
class Human {
    public String toString() {
        return "hi";
    }
}

class CusExceptionA extends RuntimeException { }

class CusExceptionB extends RuntimeException { }

public class QA2 {
    public static void main(String[] args) {

        try {
            int[] p = {17, 34, 51};
            boolean b1 = true, b2 = false;
            Human[] h = new Human[10];

                /*  missing segment */

        }
        catch (IndexOutOfBoundsException e) {
            System.out.println("A");
        }
        catch (NullPointerException e) {
            System.out.println("B");
        }
        catch (NumberFormatException e) {
            System.out.println("C");
        }
        catch (CusExceptionA e) {
            System.out.println("D");
        }
        catch (Exception e) {
            System.out.println("E");
        }
        finally {
            System.out.println("F");
        }
    }
}
```

© Vocational Training Council

(a) Write the output of the program if the following segment of code is plugged into the missing segment. [4 marks]

```
System.out.println(p[1]);
System.out.println(p[2]);
System.out.println(p[1 + 2]);
```

(b) Write the output of the program if the following segment of code is plugged into the missing segment. [3 marks]

```
if (b1 == b2 && b2 == false)
    System.out.println("OK");
else
    if (b2 == true || b2)
        throw new CusExceptionA();
    else
        throw new CusExceptionB();
```

(c) Write the output of the program if the following segment of code is plugged into the missing segment. [3 marks]

```
System.out.println(h.length);
System.out.println(h[9].toString());
```

© Vocational Training Council

A3 Consider the following ListNode and LinkedList classes,

```
class ListNode {
    Object data;
    ListNode next;
    ListNode(Object o) { data = o; next = null; }
} // class ListNode

class EmptyListException extends RuntimeException {
    public EmptyListException () {
      super("List is empty"); }
} // class EmptyListException

class LinkedList {
    private ListNode head;
    private ListNode tail;
    public LinkedList() { head = tail = null; }
    public boolean isEmpty() { return head == null; }
    public void addToHead(Object item){...}
    public void addToTail(Object item){...}
    public Object removeFromHead()throws ... {...}
    public Object removeFromTail()throws EmptyListException
    {
                /*  missing segment */
    }
```

(a) Write the missing segment for the method **`removeFromTail()`**. [6 marks]

(b) If a stack call LinkedStack is written by inheriting the above LinkedList class, write the following methods [assume the top of stack is at the head of the linked list]: [7 marks]

- `public void push(Object item){...}`
- `public Object pop( )throws EmptyListException{...}`
- `public Object top( )throws EmptyListException{...}`

(c) The following series of operations are performed on an empty stack one after one, list the contents of the stack after **EACH** operation has completed. [3 marks]

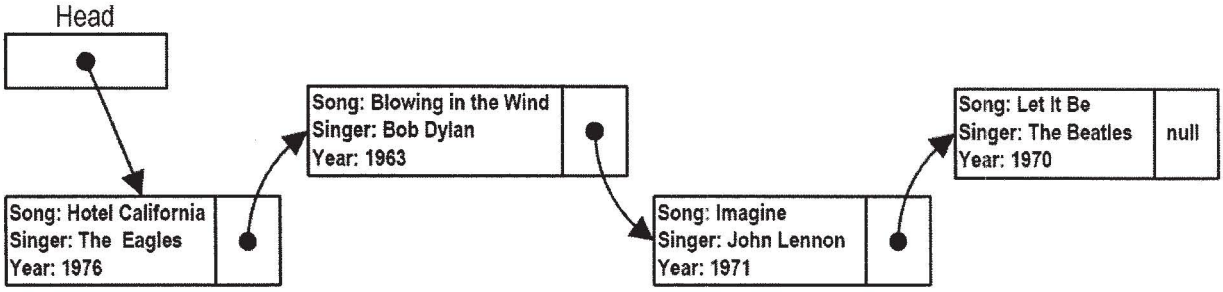
1. Push 2
2. Top
3. Pop
4. Push 23
5. Push 8

© Vocational Training Council

Section B (60 marks)

This section contains 3 questions. Each question carries 20 marks.
Answer ALL questions.

B1 (a) Given the following diagram showing a singly linked list with four Song nodes. The linked list is used in a music player program and the nodes are inserted according to the order the songs are played.



(i) Describe the steps insert a Song named "Grease" at the end of the list. [5 marks]

(ii) Complete the method **size( )** of the LinkedList class. The method returns the number of nodes in the list. [3 marks]

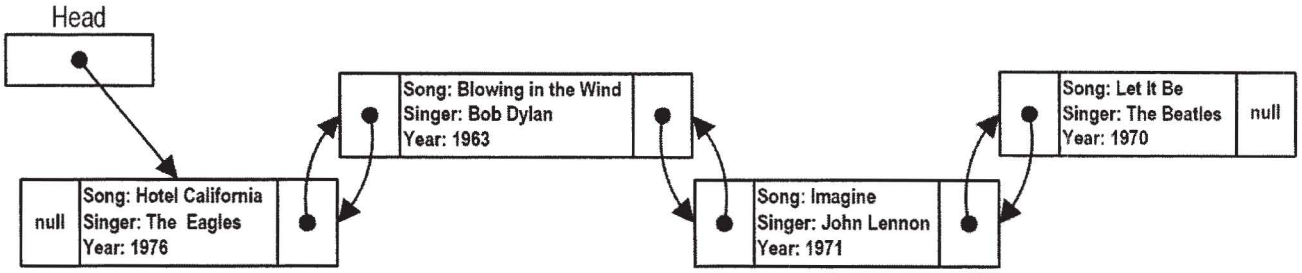
```
public class ListNode {
    String song;  String singer;  int year;
    public ListNode next;
    public ListNode(String song, String singer, int year) {…}
}

public class LinkedList {
    private ListNode head;

    public LinkedList() { head = null; }
    public boolean isEmpty() {…}
    public void addToHead(String song, String singer, int year)
        {…}

    public int size() {
        /*** TO BE COMPLETED in B1 part a(ii) ***/
    }
}
```

*** Question B1 continues in next page ***

© Vocational Training Council

*** Question B1 continues from previous page ***

(b) Given the following illustration of doubly linked list implementation for the music player.



(i) Complete the method **addToHead(...)** of DoublyList class. The method add a new node to the head of the doubly linked list. [4 marks]

(ii) Besides music player, give one application for doubly linked list. [2 marks]

```
public class DoublyNode {
  String song;  String singer;  int year;
  public DoublyNode prev;  public DoublyNode next;
  public DoublyNode(String song, String singer, int year) {
     this.song=song; this.singer=singer; this.year=year;
  }
}

public class DoublyList {
  private DoublyNode head;
  public DoublyList() { head = null; }
  public boolean isEmpty() { return (head == null); }

  public void addToHead(String song, String singer, int year){
     DoublyNode node = new DoublyNode(song, singer, year);
     if (isEmpty()) {
        /*** TO BE COMPLETED in B1 part b(i) ***/
     } else {
        /*** TO BE COMPLETED in B1 part b(i) ***/
     }
     head = node;
    }

  public void removeFromHead() throws EmptyListException {
     if (isEmpty()) throw new EmptyListException();
       head = head.next;
       if (head != null)
        head.prev = null;
  }
}
```

*** Question B1 continues in next page ***

© Vocational Training Council

***** Question B1 continues from previous page *****

(c) Given the following LinkedList class:

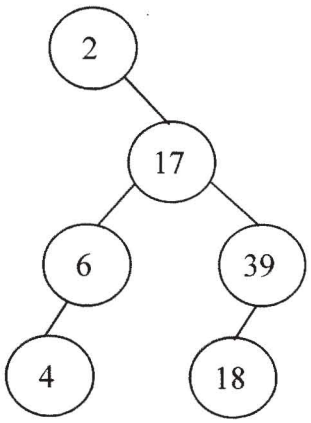
```
public class LinkedList {
  private ListNode head, tail;
  public LinkedList() {...}
  public boolean isEmpty() {...}
  public void addToHead(Object item) {...}
  public void addToTail(Object item) {...}
  public Object removeFromHead() throws EmptyListException {...}
  public Object removeFromTail() throws EmptyListException {...}
}
```

Complete the following missing segments (A) to (E) in your answer book by using the above LinkedList class together with object-oriented inheritance skill. [6 marks]

```
public class ListQueue extends LinkedList {
    public ListQueue() { ...missing segment (A)...; }

    public boolean isEmpty() { ...missing segment (B)... }

    public void enqueue(Object item)
         { ...missing segment (C)... }

    public Object dequeue() { ...missing segment (D)... }

    public Object front() {
        // return first element without removing it
        // return null object if the queue is empty
        ...missing segment (E)...
    }
}
```

© Vocational Training Council

B2 Given a Binary Search Tree which is inserted in the order of [2, 17, 39, 6 , 4, 18].



(a) List the traversal result of the above Binary Search Tree. [6 marks]

(i) Preorder traversal
(ii) Inorder traversal
(iii) Postorder traversal

(b) List the resulted array for above Binary Search Tree that implemented with array structure. [4 marks]

(c) Compare the searching performance of Binary Search Tree and Sequential Search? [2 marks]

(d) Build an AVL Tree using same item set in B2 part (a). Draw the result of the insertion for the AVL Tree **step by step**. [5 marks]

(e) Complete the following table about search performance for different data structures with same data set [2, 17, 39, 6 , 4, 18]. [3 marks]

	Searching Step needed for Best Case	Searching Step needed for Worst Case
Binary Search Tree		
AVL Tree		
Sequential Search		

© Vocational Training Council

B3 (a) Given the following sequence of numbers:

28, 18, 24, 16, 51, 30, 22, 11

(i) Perform a **Bubble Sort** on the above sequence of numbers. Show the result of **each pass** of the sorting. The first pass is shown below for your reference (number(s) in [ ] is/are sorted) :

28, 18, 24, 16, 51, 30, 22, 11
18, 24, 16, 28, 30, 22, 11, [51] [3 marks]

(ii) What is the time complexity (in Big-O notation) for an **Insertion Sort** on data in random order? [1 mark]

(iii) Perform a **Merge Sort** on the above sequence of numbers. Show your steps in partitioning and merging the list clearly. [4 marks]

(iv) Copy and complete the following tables on your answer book to show how the **Quick Sort** algorithm (using the first number as the pivot) partitions the above sequence of number into two sub-lists. The first step is shown below for your reference:

index	**0 (pivot)**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
data	28	18	24	16	51	30	22	11
Step 1	28	18	24	16	51	30	22	11
Step 2								
Step 3								
Step 4								
Step 5								
Step 6								
Step 7								
Step 8 (swap pivot)								

storeIndex
1
2

[4 marks]

(v) What is the worst case time complexity of **Quick Sort** with the first number as the pivot and give **ONE** example of data of such case. [2 marks]

***** Question B3 continues in next page *****

© Vocational Training Council

***** Question B3 continues from previous page *****

(vi) Both **Merge Sort** and **Quick Sort** have average time complexity O(n log n). However **Quick Sort** is considered to be better than **Merge Sort**. Why?
[1 mark]

(b) What are the time complexities (in Big-O notation) of the following algorithms/methods?

(i)
```
int sum = 0;
 for (int i=2; i<=n; i+=2)
   sum += i;
```
[1 mark]

(ii)
```
int sum = 0;
for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        sum = sum + i*j;
```
[2 marks]

(iii)
```
public int f(int n) {
   if (n<=1)
       return 1;
   else
       return n + f(n/2);
}
```
[2 marks]

******* END OF PAPER *******