

**FINAL**  
Examination Paper

(COVER PAGE)

Session : April 2016

Programme : Diploma In Information And Communication Technology (DICTN)

Course : **ICT2102: Introduction To Data Structure**

Date of Examination : 28 July, 2016 (Thursday)

Time : 11:00am – 1:00pm Reading Time : Nil

Duration : 2 Hours

**Special Instructions :**

Answer any **FOUR (4)** questions.

Materials permitted : Standard Calculator

Materials provided : Nil

Examiner(s) : **Ang Chee Huei** and **Koo Lee Chun**

Moderator : Siti Hawa Mohamed Said

*This paper consists of 6 printed pages, including the cover page*

DIPLOMA IN INFORMATION AND COMMUNICATIONS TECHNOLOGY (DICTN)  
 ICT2102: INTRODUCTION TO DATA STRUCTURE  
 FINAL EXAMINATION: APRIL 2016 SESSION

**Instruction:** This paper consists of **SIX (6)** questions. Answer any **FOUR (4)** questions in the answer booklet provided. All questions carry equal marks.

**QUESTION 1**

(a) Fill in the blanks :

- (i) \_\_\_\_\_ means that a function can be used without taking into account how the function is implemented.
- (ii) A variable that is used to store the address of another variable is called a \_\_\_\_\_.
- (iii) When a linked-list based stack is empty, the value of the top pointer is \_\_\_\_\_.
- (iv) Which data structure has the “First in, First out” feature? Answer: \_\_\_\_\_.
- (v) A \_\_\_\_\_ is a constructor without arguments.
- (vi) Traversing the nodes of a tree in \_\_\_\_\_ visits the parent after the children.
- (vii) In a \_\_\_\_\_ the value stored in any parent node is never greater than the value stored in any right child of the parent.
- (viii) \_\_\_\_\_ refer to new types of data, defined by the programmer, where the implementation details are hidden from the user, and access to the data is provided through an interface to a specific set of operations.

(8 marks)

(b) Using diagrams, show the detailed workings of an insertion sort in ascending order on the following array.

5	3	1	9	8	2	4	7
---	---	---	---	---	---	---	---

(7 marks)

(c) Write in pseudocode or otherwise, the algorithm for bubble sort. How does it perform when an array becomes sorted at an early stage?

(10 marks)

**QUESTION 2**

- (a) Show the results of the following sequence of events, by drawing the state of the data structure:

`insert(4), insert(8), insert(1), insert(6), remove(), remove()`

where `insert` and `remove` are the operations that correspond to the basic operations in :

- (i) stack (6 marks)
  - (ii) queue (6 marks)
- (b) Consider a linked list :
- (i) Describe an algorithm in plain English, to remove the last item from the list. (4 marks)
  - (ii) Write a function `removeLast ()` that implements your algorithm in (i). You can use without need to define all the attributes and functions that are members of the `LinkedList` class discussed in lectures. (9 marks)

**QUESTION 3**

- (a) What is a binary search tree? Outline the advantage of it. (5 marks)

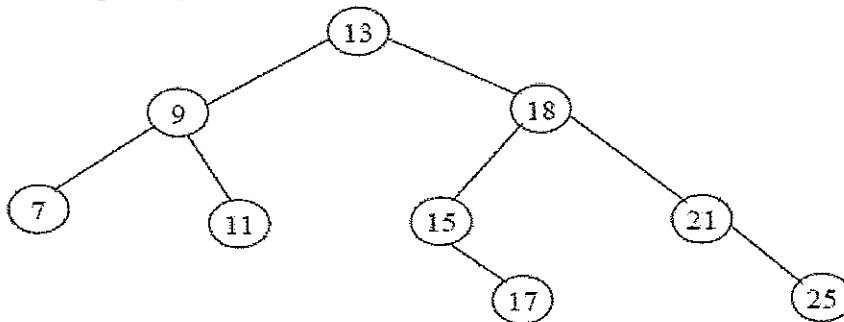
- (b) Show how binary search works when searching for 20 in the following array:

2	3	5	8	10	12	13	15	18	20	23
---	---	---	---	----	----	----	----	----	----	----

(6 marks)

- (c) Write an algorithm for binary search on a sorted array in pseudocode or C++. (8 marks)

- (d) Show the effect of inserting key 14 and removing key 18 (in that order) into the following binary search tree.



(6 marks)

**QUESTION 4**

- (a) Assume the ADT Stack consists of following attributes:

```

struct StackNode
{
    int item;
    StackNode *next;
};
StackNode *topPtr;

```

Write the C++ implementation of the following functions in the ADT Stack :

- (i) function
- `push()`
- that push an item into the stack. Use the prototype below :

```
bool Stack::push (double newItem)
```

(5 marks)

- (ii) function
- `displayStack()`
- that display the stack content from top to bottom.

Use the prototype below:

```
void Stack::displayStack ()
```

(5 marks)

- (b) Assume
- `s1`
- is a stack consists of 1,4,2,8,9 where 9 is the top of the stack. Write a program to reverse the content of
- `s1`
- . For example, after reversing, the stack shall contain 9, 8, 2, 4, 1 where 1 is the top of the stack. You program shall display appropriate message to handle empty stack. You are limited to use only the operations provided in ADT Stack :
- `push()`
- ,
- `pop()`
- ,
- `isEmpty()`
- ,
- `getSize()`
- ;

(5 marks)

- (c) Show the complete working steps (including stack contents) to evaluate the postfix expression below.

4	3	6	3	*	12	-	*	+	=
---	---	---	---	---	----	---	---	---	---

(10 marks)

## QUESTION 5

- (a) Consider the C++ code fragment below. Write down the output.

```

char str[] = "Exam Paper";
int *p;
char *c;
int index = 5;
c = str;
cout << c << endl;
p = &index;
cout << *p << endl;
c = c + *p;
cout << *c << endl;

```

(6 marks)

- (b) The
- Node*
- class below can be used to implement a linked list.

```

class Node {
private:
    int data;
    Node *next;
public:
    Node();
    Node(int x, Node *ptr);
    int getData();           //return the data
    Node *getNext();        //return next Node's pointer
    void setData(int x);    //set data of the Node to x
    void setNext(Node *ptr); //set next pointer to Node ptr
};

```

- (i) Write the implementation of the default constructor that will create a
- Node*
- object with the value of
- data*
- set to 0 and the
- next*
- pointer point to nothing. (4 marks)

- (ii) Write the implementation of the constructor
- `Node(int x, Node *ptr);`
- (3 marks)

- (iii) Write the implementation of all other member functions :

```

int getData();
Node *getNext();
void setData(int x);
void setNext(Node *ptr);

```

(12 marks)

## QUESTION 6

(a) Define the *queue* data structure and give a real-life example of *queue*. (5 marks)

(b) The class below is an ADT for a static array based queue:

```
const int MAX=100;
class Queue{
public:
    Queue(length=0; head=0; tail=0);
    void enqueue (char x);
    char dequeue();
    int getHead();
    int getTail();
    int getSize();
    bool isEmpty();
private:
    char qArray[MAX]
    int length;
    int head;
    int tail;
};
```

(i) Provide the full implementation in C++ of the member function `enqueue (char x)`, which insert the data  $x$  into the queue. Note that you cannot insert the data if the queue is full. (8 marks)

(ii) Provide the full implementation in C++ of the member function `dequeue ()`, which remove a data  $x$  from the queue. Note that you cannot remove the data if the queue is empty. (5 marks)

(c) Write the selection sort algorithm (function) for an array of size  $n$ . (7 marks)