

FINAL
Examination Paper

(COVER PAGE)

Session : January 2017

Programme : Diploma In Information And Communication Technology (DICTN)

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

Date of Examination : 07 March, 2017 (Tuesday)

Time : 2:00pm – 4:00pm Reading Time : Nil

Duration : 2 Hours

Special Instructions :

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

Materials permitted : Non-programmable calculator

Materials provided : Nil

Examiner(s) : **Shee Fui Chie**

Moderator : Siti Hawa Mohamed Said

This paper consists of 7 printed pages, including the cover page

DIPLOMA IN INFORMATION AND COMMUNICATIONS TECHNOLOGY PROGRAMME
(DICTN)
ICT2102: INTRODUCTION TO DATA STRUCTURE
RESIT EXAMINATION: JANUARY 2017 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) Write a function `readAttendance()` in C++ that would read data into the 20 records of the following structure. The overtime is assigned according to the attendance. If the morning attendance is more than 20 and the afternoon attendance is greater than or equal to 25, then overtime is RM100 else RM30.

```
struct employee {  
    string name;  
    string ID;  
    int morningAttendance;  
    int afternoonAttendance;  
    float overtime;  
} frontEnd[20];
```

(10 marks)

- (b) Write a program to read the rainfall of Kuala Lumpur into a dynamic array called *rainfall*. Display the rainfall in the array that is less than or equal to the average rainfall. Free memory of *rainfall* before the program terminates. (10 marks)
- (c) Explain the terms Abstraction and Abstract Data Type (ADT). (5 marks)

(Total: 25 marks)

Question 2

(a) Assume that you are given an ADT that stores the of subjects offer in a semester.

- (i) Identify **FIVE (5)** operations that can be performed on the ADT. (5 marks)
- (ii) Provide **FOUR (4)** types of data structure to store the data in the list (4 marks)

(b) Explain the purpose of the following:

- (i) Default constructors
- (ii) Initialization constructors
- (iii) Copy constructors
- (iv) Destructor

(8 marks)

(c) The following Abstract Data Type (ADT) called *Attendance* is used to store a set of student names using dynamic array.

```
class Attendance
{
public:
    Attendance( int noStudents ); //constructor
    ~ Attendance( ); //destructor
    void editNoStudent(int newNoStudent );//change attendance
status
private:
    string *names; // points to the dynamic array
    int size; // size of the array
};
```

(i) Implement the constructor and create the array (*names*) dynamically based on the value of *noStudents*. Assign the *size* to *noStudents*. If the *noStudent* is less than 1, set the *size* to 1.

(6 marks)

(ii) Implement the destructor to deallocate the memory space from the dynamic array.

(2 marks)

(Total: 25 marks)

Question 3

- (a) Array is a collection of elements having same data type with common name. Linked list is an ordered collection of elements which are connected by links/pointers. Use diagrams to represent the following scores in array and linked list with appropriate links (state the data and link):

20, 45, 78, 92, -100

(6 marks)

- (b) Implement a linked list to store the records of the residents in a community. The community needs to record the resident's information: house number, street, and maintenance fees.

- (i) Declare a linked list class *communityClass* with the following:

- First private member *communityNode* is a structure contains the members for the resident's information and a link that points to the next item.
- Second private member *NODEPTR* is a pointer of *communityNode*.
- Declare third private member *head* with *NODEPTR*.
- First public member is the default constructor.
- Second public member is an associate operation *addNode()* that inserts a new record of resident's information to the beginning of the linked list.

(8 marks)

- (ii) Define the associate operation *addNode()*.

(7 marks)

- (iii) Declare a variable *BTcommunity* with the type of *communityClass*. Write the instruction to use the variable *BTcommunity* to access the operation *addNode()* with the house number 20, street is "Lorong Batu Nilam 2c", maintenance fees as 200.00.

(4 marks)

(Total: 25 marks)

Question 4

(a) An ADT stack is a data structure used to store a collection of objects. Describe **FOUR** (4) operations perform by the stack. (8 marks)

(b) Consider the following sequence of operations of stack:

```
push("ICT1101")
push("ICT1103")
push("ICT2102")
pop()
push("ICT2107B")
push("INT2100")
pop()
pop()
push("ICT2100")
```

Use a diagram to illustrate the state of the stack after each of the nine operations. (9 marks)

(c) The banking queue system facilitates the queuing process to customers who are waiting for their turn to be served. Consider the following structure and function:

```
struct BankQueueNode {
    int number;
    BankQueueNode *next;
};

Bool Queue::dequeueNumber (int & data){
    ...
}
```

Write a function member dequeueNumber () that removes the number in the queue when the counter is free to serve the bank customer. (8 marks)

(Total: 25 marks)

Question 5

(a) Transform the following infix expression into postfix expression:

- (i) $a+(b+c)*d$ (3 marks)
- (ii) $4 + 6 / 2 * 5 - 3$ (3 marks)

(b) Define the following term with respect to trees:

- (i) Root (2 marks)
- (ii) Parent and child (2 marks)
- (iii) Leaf node (2 marks)
- (iv) Level of the tree (2 marks)
- (v) Height of the tree (2 marks)

(c) Provide the implementation of the insertion sort function below.

```
void insertionSort (int a[], int n)
{
    ....
}
```

(9 marks)

(Total: 25 marks)

Question 6

(a) Explain the process of pre-order and in-order traversal for binary tree. (6 marks)

(b) Draw a binary search tree for the following set of data: 56, 34, 40, 60, 62, 11, 78, 17, 2. (9 marks)

- (c) Given the following declaration of a binary search tree. Provide the implementation the function *bool search (int data)* below.

```
class BST {  
private:  
    struct TreeNode {  
        int item;  
        TreeNode *leftChild;  
        TreeNode *rightChild;  
    };  
    TreeNode *root; //pointer to the root node  
    //...  
}  
public :  
    BST ();  
    bool search (int data);  
    //...  
};
```

(10 marks)

(Total: 25 marks)

