

**FINAL**  
Examination Paper

(COVER PAGE)

Session : August 2015

Programme : Diploma In Information And Communication Technology (DICTN)

Course : ICT2102: Introduction To Data Structure

Date of Examination : December 9, 2015

Time : 2.00pm – 4.00pm Reading Time : Nil

Duration : 2 Hours

**Special Instructions** :

Answer any FOUR (4) questions.

Materials permitted : Standard Calculator

Materials provided : Nil

Examiner(s) : Mr. Koo Lee Chun and Ms. Shee Fui Chie

Moderator : Mr. Ang Chee Huei

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

DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY (DICTN)  
ICT2102 : INTRODUCTION TO DATA STRUCTURE  
FINAL EXAMINATION: AUGUST 2015 SESSION

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) Trace the output for the following code segments :

(i)

```
int a;  
int b;  
int* p;  
int* q;  
a = 3;  
p = &a;  
q = p;  
b = 4;  
cout << *p << endl;  
    << *q << endl;  
    << a << endl;  
    << b;
```

(4 marks)

(ii)

```
int array[]={2,4,6,8};  
int *arptr = array;  
int value = *arptr;  
cout << value << endl;  
value = *arptr++;  
cout << value << endl;  
value = *++arptr;  
cout << value << endl;  
value = *arptr + 3;  
cout << value << endl;  
value = *(arptr + 1);  
cout << value << endl;
```

(5 marks)

- (b) Given the following structure definition:

```
struct GAME{
    string gamename;
    string participant[30];
    int score[30];
    string status[30];
} cybercafe[10];
```

Write a function *inputGame()* in C++ that would read 10 records into cybercafe. The status is assigned according to the score. If the score is greater than 1000, assigned 'Premium' to status else assigned "Regular".

(12 marks)

- (c) Use the C++ dynamic allocation operators to:

- (i) Allocate array of variable *yptr* in integer with 30 elements
- (ii) Free memory of *yptr*.

(4 marks)

### Question 2

- (a) Define **priceStruct** by using struct for the pricing schema (custType, price) for a Theme Park. (3 marks)
- (b) Define a linked list node, **priceNode**, using struct that consists of structure defined in Question 2(a) : (3 marks)
- (c) Assume S is a linked list that consists of a list of priceNode as following :

Customer type	Price (RM)
Child	8.00
Adult	10.00
Family (A) 2 adults 2 children	20.00
Family (B) 2 adults 4 children	30.00

Assume the ADT list provide the following methods:

`bool insert (int position, priceStruct newItem);`

`//insert a price Schema in the position specified by position`

```
bool delete(int position)
//remove a priceSchema specified by the position from the loinked list

bool update(int position, priceStruct item);
//insert a price Schema in the position specified by position
```

Based on the structure defined in (a), write a program code segment in C++ to perform the following action in sequence:

- Insert Item with customer type=Senior Citizen, Price= 5.00 into the node after adult price scheme.
- Remove The Family package (B) from the linked list.
- Update The Family package (A) with new price : RM 30.00
- Insert a new Family package (B) 2 Adults 1 child, with the price of RM 22.00 after family package (A)

(7 marks)

(d) Implement the below new function into the ADT linked-list above :

```
void insertHead(priceStruct p) // to insert a price Scheme at the beginning of
linked list.
```

(6 marks)

(e) Discuss **THREE (3)** distinct advantages of List implementation using linked list compare to Array.

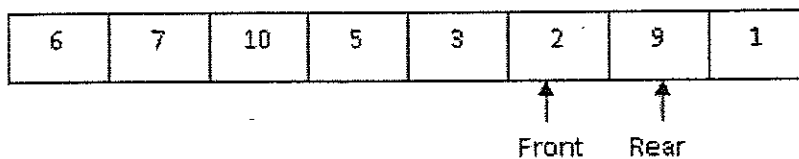
(6 marks)

**Question 3**

(a) Describe Queue data structure. Provide an example to support your answer.

(4 marks)

(b) Suppose you have a queue Q which implementing a circular array sized 8 at the state as following:



Draw the diagrams to represent EACH state changes in for Q above (including Front and Rear pointer) when execute EACH of the following statement. Assume statements are executed continuously.

- (i) Q.enqueue(10)
- (ii) Q.enqueue(7)
- (iii) Q.dequeue()
- (iv) Q.enqueue(5)
- (v) Q.enqueue(9)
- (vi) Q.dequeue(4)

(6 marks)

- (c) Given the following class declaration, write the complete enqueue() and dequeue() functions to execute the program using a linked list as a linear queue. enqueue() is a function to add elements into the linked list and dequeue() is a function to delete an element from the linked list. There should be additional functions to initialize the queue and to check for empty queue.

```
class Queue
{
public:
    Queue();
    bool isEmpty();
    bool enqueue(int newItem)
    bool dequeue(int &data);
private:
    QueueNode *frontPtr; // pointer to front of Queue
    QueueNode *backPtr;  // pointer to back of Queue
    int length;          // the length of the queue
};
```

(15 marks)

#### Question 4

- (a) Show the results of the following sequence of events, by drawing the state of the data structure in a linked-list and array-based stack respectively:

push(14), push(19), push(20), pop(), push(12)

(5 marks)

- (b) Consider a stack of integers cStack. Using only the stack methods isEmpty(), pop(), pop(int x), and push(int y) write code (in C++) which pops all the elements off the cStack and display the sum of all these elements.

(5 marks)

- (c) Describe an algorithm that use stack to help in recognizing the balance of parenthesis in a statement.

(5 marks)

- (d) Convert the below expression into postfix. Describe each processing steps when evaluating the postfix expression in a stack. :

$$30-10/20*4$$

(10 marks)

**Question 5**

- (a) Construct a binary tree by inserting the value in the following order :

20, 15, 18, 40, 45, 23, 56, 70

(8 marks)

- (b) Based on Question 5 (a) answer, list the result after traverse the tree using :

(i) Preorder

(ii) In-order

(iii) Post-order

(12 marks)

- (c) Based on the binary tree constructed in Question 5(a), show the effect of inserting key 52, follow by removing key 40.

(5 marks)

**Question 6**

- (a) Define Abstract Data Type. Provide two examples of useful data structure for an ADT.

(4 marks)

- (b) In the course ICT1234, the coursework assessment consists of 3 components as shown in below Table:

Component	Description	Contribution to the coursework mark
Test	A 100-mark midterm test	50%
Online Quiz	30 objective questions of 1 mark each	30%
Assignment	An individual assignment which carries a total of 40 marks	20%

Write in C++, a class named Coursework which should contain the following data fields: student ID, and marks scored for each of the coursework components.

The class should have the following features:

- A no-arg constructor
- A parameterized constructor
- Mutators for all the data members except student ID
- Accessor for all data members
- A method -named `getCourseworkMark` that will compute the final coursework mark using the formula below:  
$$\text{coursework mark} = \text{test mark} \times 0.5 + \text{online quiz mark} + \text{assignment mark} \times 0.5$$

(11 marks)

(c) Consider the following list of words:  
yellow, orange, pink, red, blue, black, white

- (i) Sort the above list alphabetically using selection sort. Show the details working of the passes.
- (ii) Sort the above list alphabetically using a bubble sort. Show the details working of the passes.

(10 marks)

**~THE END~**

*ICT2102(F)August2015*

