	<p style="text-align: center;">MIDTERM EXAMINATION SEMESTER FALL 2003 CS301-DATA STRUCTURE</p>	<p>Total Marks:86 Duration: 60min</p>
<p>Instructions</p> <p>Please read the following instructions carefully before attempting any question:</p> <ol style="list-style-type: none">1. The duration of this examination is 60 Mins.2. This examination is closed book, closed notes, closed neighbors; any one found cheating will get no grade.3. Unless stated otherwise, all questions carry a single mark.4. Do not ask any questions about the contents of this examination from anyone.<ol style="list-style-type: none">a. If you think that there is something wrong with any of the questions, attempt it to the best of your understanding.b. If you believe that some essential piece of information is missing, make an appropriate assumption and use it to solve the problem.5. Most, but not all, of the examination consists of multiple-choice questions. Choose only one choice as your answer.<ol style="list-style-type: none">a. If you believe that two (or more) of the choices are the correct ones for a particular question, choose the best one.b. On the other hand, if you believe that all of the choices provided for a particular question are the wrong ones, select the one that appears to you as being the least wrong.7. You are allowed to use any development environment like Dev C++ etc.		

Question No: 1

Marks: 2

Is it possible for a member function of a class to activate another member function of the same class?

- a. No.
- b. Yes, but only public member functions.
- c. Yes, but only private member functions.
- d. Yes, both public and private member functions can be activated within another member function.

Question No: 2

Marks: 2

Consider this class definition:

```
class quiz
{
public:
    quiz( );
    int f( );
    int g( ) const;
private:
    double score;
};
```

Which functions can carry out an assignment `score=1.0;` to the private member variable `score`?

- a. Both f and g can carry out the assignment.
- b. f can carry out the assignment, but not g.
- c. g can carry out the assignment, but not f.
- d. Neither f nor g can carry out the assignment

Question No: 3

Marks: 2

In C++, when allocating an array of objects, what constructor is used to initialize all of the objects in the array?

- a. The automatic copy constructor.
- b. The constructor specified at the declaration.
- c. The default constructor.
- d. None of the above.

Question No: 4

Marks: 2

The list abstract data type (ADT) is used to work with ordered or unordered sequence of items such as numbers or strings. What of the following implementation of list ADT is best to answer questions such as "What is the item at position n?"

- a. Lists implemented with an array.
- b. Doubly-linked lists.
- c. Singly-linked lists.
- d. Doubly-linked or singly-linked lists are equally best

Question No: 5

Marks: 5

Consider this function declaration:

```
void quiz(int i)
{
    if (i > 1)
    {
        quiz(i / 2);
        quiz(i / 2);
    }
    cout << "*";
}
```

How many asterisks are printed by the function call *quiz(5)*?

- a. 3
- b. 4
- c. 7
- d. 8
- e. Some other number

Question No: 6

Marks: 2

Suppose T is a binary tree with 14 nodes. What is the minimum possible depth of T?

- a. 0
- b. 3
- c. 4
- d. 5

Question No: 7

Marks: 2

“Entries in a stack are Ordered”. What is the meaning of this statement?

- a. A collection of stacks can be sorted.
- b. Stack entries may be compared with the < operation.
- c. The entries must be stored in a linked list.
- d. There is a first entry, a second entry, and so on.

Question No: 8

Marks: 2

Which of the following applications may use a stack?

- a. A parentheses balancing program.
- b. Keeping track of local variables at run time.
- c. In-order traversal of a binary tree.
- d. All of the above.

Question No: 9

Marks: 2

When the compiler compiles your program, how is a recursive call treated differently than a non-recursive function call?

- a. Parameters are all treated as reference arguments
- b. Parameters are all treated as value arguments
- c. There is no duplication of local variables
- d. None of the above

Question No: 10

Marks: 2

What is the maximum depth of recursive calls a function may make?

- a. 1
- b. 2
- c. n (where n is the argument)
- d. There is no fixed maximum

Question No: 11

Marks: 2

In which location do dynamic variables reside?

- a. The code segment.
- b. The data segment.
- c. The heap.
- d. The run-time stack.

Question No: 12

Marks: 2

Select the one FALSE statement about binary trees:

- a. Every binary tree has at least one node.
- b. Every non-empty tree has exactly one root node.
- c. Every node has at most two children.
- d. Every non-root node has exactly one parent.

Question No: 13

Marks: 6

The nodes of a binary tree have data 1, 2, 3, 4. The in-order traversal of the tree yields 2,1,4,3. The postorder traversal is 2, 4, 3, 1. The root of the tree is at level 0.

Q1: In this binary tree, which value is at the root? (1 Pt)

(A) 1 (B) 2 (C) 3 (D) 4 (E) none

Q2: Which value is in the left child of the root? (1 Pt)

(A) 1 (B) 2 (C) 3 (D) 4 (E) none

Q3: Which value is in the right child of the root? (1 Pt)

(A) 1 (B) 2 (C) 3 (D) 4 (E) none

Q4: Which value is in a node at level 2 and is the left child of a node at level 1? (1.5 Pt)

(A) 1 (B) 2 (C) 3 (D) 4 (E) none

Q5: Which value is in a node at level 2 and is the right child of a node at level 1? (1.5 Pt)

(A) 1 (B) 2 (C) 3 (D) 4 (E) none

Question No: 14

Marks: 5

I am going to execute this code with THREE pushes and ONE pop:

```
Stack s;  
s.push(1);  
s.push(2);  
s.push(3);  
cout << s.pop( );
```

Suppose that the stack *s* is represented by a *fixed-sized array*. Draw the state of the private member variables “data” and “top” of “s” after the above code:

	0	1	2	3	4	5	6	7	8	9				
data												top:		

Answer/Solution

0	1	2	3	4	5	6	7	8	9			
1	2										Top	1

Question No: 15

Marks: 10

Complete the body of this function. Use a Queue of characters to store the input line as it is being read.

```
int counter( )  
// Precondition:  
// There is a line of input waiting to be read from cin.  
// Postcondition:  
// A line of input has been read from cin, up to but not  
// including the newline character. The return value of  
// the function is the number of times that the LAST  
// character of the line appeared somewhere in this line.  
  
// EXAMPLE  
// Input: ABBXDXXZX The value returned by counter would  
// be 4 for this input since there are 4 X's in  
// the input line.  
{  
int answer = 0;  
Queue q;
```

Answer/Solution

```

int counter()
{
    char a[100];
    int i=0;
    int answer=0;
    Queue q;

    cin.getline(a,98,'\n');
    for(i=0;i<strlen(a);i++)
    {
        q.enqueue(a[i]);
    }

    i--;
    while(!q.isEmpty())
    {
        if(a[i]==q.dequeue())
        {
            answer++;
        }
    }
    return answer;
}

```

Question No: 16

Marks: 5

I am going to execute this code with **THREE** inserts and **ONE** remove:

```

Queue s;
s.insert(1);
s.insert(2);
s.insert(3);
cout << s.remove();

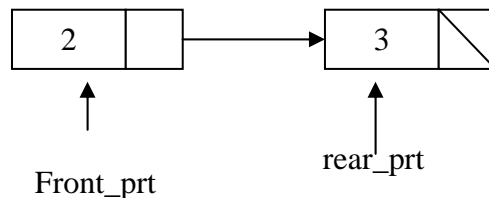
```

Suppose that s is represented by a singly linked list. Draw the linked list and the state of the private member variables of s after the above code:

front_ptr

rear_ptr

Answer/Solution



Question No: 17

Marks: 10

A list is said to be *sorted* if the elements are in (say) increasing order, so {1, 2, 3, 3, 4, 4} is sorted whereas {1, 2, 3, 4, 3, 1} is not. A partial declaration for a singly-linked integer list class that keeps data in sorted order is as follows:

```
class Node {
public:
    int data;
    Node* next;
    Node(int d, Node *n)
    {
        data = d; next = n;
    }
};
class intList
{
public:
    // ...
    // Remove any duplicate elements from the sorted list
    void removeDuplicates();
private:
    Node* head; // points to the first Node in the list
    // ...
};
```

Give an implementation of the function `removeDuplicates` which removes any duplicate elements from the sorted list, so, for example {1, 2, 3, 3, 4, 4} would be reduced to {1, 2, 3, 4}.

Answer/Solution

```
// there a number of ways of doing this. Here is one.
void intList::removeDuplicates()
{
    Node* cur = head;
    Node *temp;
    Node* next = head != NULL? head->next : NULL;
    while( next != NULL )
    {
        // delete next node if it has the same data as current node
        if( cur->data == next->data )
        {
            temp=next;
            next = next->next; // cur stays where it is
            cur->next=next;
            delete temp;;
        }
        else
        {
            cur = next; // move to next pair
            next = cur->next;
        }
    }
}
```

Trace the running of the infix to postfix conversion algorithm on the infix expression
A - (B + C) / D

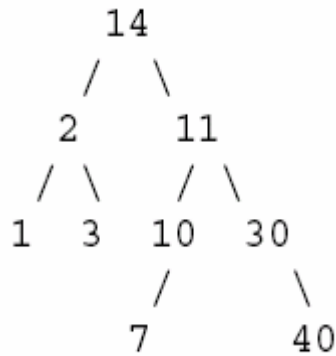
Answer/Solution

Symbol	Postfix	Stack
A	A	
-	A	-
(A	- (
B	AB	- (
+	AB	- (+
C	ABC	- (+
)	ABC+	- (
	ABC+	-
/	ABC+	- /
D	ABC+D	- /
	ABC+D/-	

Question No: 19

Marks: 13

Here is a small binary tree:



- A. What are all the leaves? (2pts)
- C. What are the ancestors of the node 10? (2pts)
- D. What are the descendants of the node 30? (2pts)
- E. Is the tree a binary search tree (BST) (true/false)? (2pts)
- F. Print the tree when visited in post-order manner? (5pts)

Answer/Solution

- A) Leaves of the tree = 1,3,7,40
- B) Ancestors of the node 10 = 11,14
- C) Descendants of the node 30 = 40
- D) Is the tree a binary search tree (BST) (true/false) False
- E) Post Order Traversal = 1,3,2,7,10,40,30,11,14

