Part A: Word definitions [20 marks]

Match up the term on the left with the phrase on the right that best describes the term. An example answer has been provided for you.

A	abstract class		a collection of data values together with a set of well-specified operations on that data
B	quicksort		a collection of elements (or nodes) that are connected, one to another, such that each element references the next in the collection, and any element can act as the list head
C	encapsulation		in Java; a program component that declares methods required of a class
D	foreach loop		a collection of nodes whose last node references the first node
E	abstract data type		an iteration construct especially useful when iterating through all of the elements in a collection
F	backtracking		a property of a list where the removal/retrieval operations access the most recently inserted item
G	interface		the usual method normally used in Java when determining the sort order of two objects
H	divide-and-conquer		an algorithm that requires the collection of data to fit entirely in the computer's main memory
I	node		a technique for hiding the inner details of some data type's implementation
J	internal sort		a class without instances that forms the basis of other classes that descend (i.e., are subclassed) from it
K	push		a set of zero or more nodes, partitioned into a root node and two possible empty sets, that are also the same kind of thing as this term
L	LIFO	**P**	**business tycoon from the State of New Jersey who has a dramatic hairpiece**
M	recursion		a process that solves a problem by breaking it into smaller sub-problems
N	circular linked list		a sorting algorithm that partitions an array's elements around a pivot to generate two smaller sorting problems
O	postorder		a problem-solving technique that retraces steps in order to visit other possible solutions
P	**Donald Trump**		a problem solving technique that decomposes a problem into smaller versions of the same problem
Q	complete binary tree		an element in a ref-based implementation of a linked-list, stack, or tree
R	binary tree		a stack operation
S	linked list		indicates the relative position taken by an operator relative to operands in an expression
T	box trace		may be implemented using a single array storing node item values (i.e., no need for left and right links)
U	compareTo()		one possible way of visualizing the recursive calls made by a method call

Part B (Multiple Choice): 50 marks

All 25 questions in this part have equal weight (i.e., two marks). ***For each question in this part, clearly circle the letter corresponding to your answer.***

1. A reference variable declared as a data field within a class has the default value _____.

   a) `0`
   b) `1`
   c) `null`
   d) `empty`
   e) None of the above.

2. In describing the linked lists as covered in lectures, the last node of a linked list ______.
   a) has the value `null`
   b) has a `next` reference whose value is `null`
   c) has a `next` reference which references the first node of the list
   d) cannot store any data
   e) None of the above

3. Which of the following statements deletes the first node of a linked list that has 10 nodes?

   a) `head.next = curr.next;`
   b) `prev.next = curr.next;`
   c) `head = head.next;`
   d) `head = null;`
   e) None of the above.

4. A _______ can be used to facilitate adding nodes to the end of a linked list.

   a) head record
   b) dummy head node
   c) `tail` reference
   d) dummy head node
   e) None of the above

5. These values are pushed onto the stack in the order given:

   6, 2, 7, 13, 5, 4

   Which number will be the first to be removed from the stack?

   a) 6
   b) 2
   c) 5
   d) 13
   e) None of the above

6. The item that is removed first from a stack is called the ________ of the stack

   a) front
   b) top
   c) base
   d) prime
   e) None of the above.

7. The last-in, first-out (LIFO) property is found in the ADT ______.

   a) list
   b) binary tree
   c) binary search tree
   d) array
   e) None of the above

8. The ______ method of the ADT stack retrieves and then removes the item at the top of a non-empty stack.

   a) createStack
   b) push
   c) pop
   d) peek
   e) None of the above

9. Which of the following methods of the ADT stack accepts a parameter?

   a) `push`
   b) `pop`
   c) `createStack`
   d) `peek`
   e) None of the above.

10. Which one of the following strings contains balanced braces?

   a) `ab{cde{fg}hi{jkl}`
   b) `ab{cde{fghi}j}kl}`
   c) `{abc{de}{fg}hij}kl`
   d) `{ab{cde{fgh}ijkl}`
   e) None of the above.

11. A `StackInterface` class provides the specifications for ___________.

   a) only the array-based implementation of a stack
   b) only the reference-based implementation of a stack
   c) only an implementation of a stack that uses the ADT list
   d) all the implementations of a stack
   e) None of the above.

12. What is the postfix form of the infix expression `(a + b) * (c / d)`?

   a) `a b + c * d /`
   b) `a b * c / d +`
   c) `a + b * c / d`
   d) `a b + c d * /`
   e) None of the above.

13. Assuming a linked list of n nodes, the code fragment

```
Node curr = head;
while (curr != null) {
     System.out.println(curr.item);
     curr = curr.next;
}
```

requires _______ assignments.

a) n
b) n - 1
c) n + 1
d) 1
e) None of the above.

14. The solution to the Towers of Hanoi problem with n disks requires $2^n - 1$ moves. If each move requires the same time *m*, the solution requires _____ time units.

a) $2^n - 1$
b) $(2^n - 1) + m$
c) $(2^n - 1) - m$
d) $(2^n - 1) / m$
e) None of the above.

15. With reference to the course lectures, which of the following is recommended for comparing the time efficiency two algorithms?

a) growth rates of the two algorithms
b) implementations of the two algorithms
c) test data used to test programs which implement the two algorithms
d) computers on which programs which implement the two algorithms are run
e) None of the above.

16. Which of the following growth-rate functions grows the fastest?

a) $O(n)$
b) $O(n^2)$
c) $O(1)$
d) $O(\log_2 n)$
e) None of the above.

17. In the best case, a sequential search is ______.

a) O(n)
b) O(1)
c) O($\log_2 n$)
d) O(n^2)
e) None of the above.

18. The complete traversal of a binary tree with *n* nodes is ______.

a) O(n)
b) O(1)
c) O(n^2)
d) O(2^n)
e) None of the above.

19. In an array-based representation of a complete binary tree, which of the following represents the left child of `tree[i]`?

a) `tree[i+2]`
b) `tree[i-2]`
c) `tree[i/2]`
d) `tree[i*i+1]`
e) None of the above.

20. A full binary tree with height 4 has ______ nodes.

a) 15
b) 16
c) 31
d) 32
e) None of the above.

21. Given the following array

```
4, 15, 8, 3, 28, 21
```

which of the following represents the array after the second swap of the selection sort?

a) `4, 3, 8, 15, 21, 28`
b) `4, 15, 8, 3, 21, 28`
c) `3, 4, 8, 15, 21, 28`
d) `21, 4, 3, 8, 15, 28`
e) None of the above.

22. The ___________ compares adjacent items and exchanges them if they are out of order.

a) selection sort
b) binary search
c) bubble sort
d) quicksort
e) None of the above.

23. A node that is directly below node n in a tree is called a(n) _______ of node n.

a) root
b) leaf
c) offspring
d) parent
e) None of the above.

24. The maximum number of comparisons for a retrieval operation in a binary search tree is the _________.

   a) length of the tree
   b) height of the tree
   c) number of leaves in the tree
   d) number of nodes in the tree
   e) None of the above.

25. The city of Victoria, B.C. ________.

   a) is famous in the winter for its excellent alpine ski runs at Beacon Hill Park
   b) has absolutely no deer anywhere nearby
   c) was initially the capital of Canada because it was named for Queen Victoria Beckham
   d) is a pretty town on the Atlantic Ocean
   e) None of the above.

Part C (short answers): 20 marks

(a) Assuming we do not use recursion, why is a loop necessary to find an arbitrary node in a linked list?

(b) What is the restriction placed on the push operation for an array-based implementation of a stack?

(c) How does the quicksort algorithm partition an array?

(d) Write a *for*-loop that prints the numbers from 100 to 1. You *must* start with the code provided below *and are not permitted to add code before it.*

```
for (int i = 0;
```

Part D (Stacks): 35 marks

Assume we have some reference-based implementation of a stack ADT for storing Strings called `StringStackRefBased`. Implemented operations are `isEmpty`, `pop`, `push`, `peek` and `popAll`.

Now suppose we have an instance of the stack named `aStack` on which some items have been pushed, and another empty stack named `tempStack`:

```
StringStackRefBased aStack = new StringStackRefBased();
/* ... Operations pushing values onto aStack ... */
StringStackRefBased tempStack = new StringStackRefBased();
```

Show how you can perform each of the following tasks by using only the operations of the `StringStackRefBased` class. Your answers must be written in Java and need not be written as a whole method.

(a) Print the contents of `aStack` in reverse order; that is, print the top item last but with the contents of `aStack` left unchanged after the output is complete.

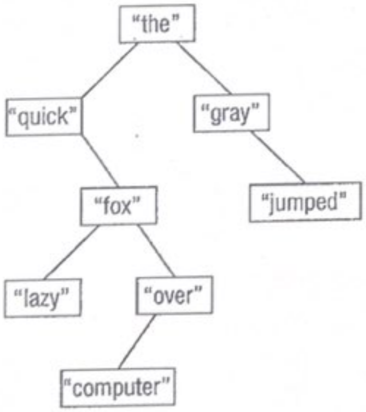
(b) Count the number of items in aStack but with the contents of aStack the same before and after the counting.

(c) Delete from aStack every occurrence of the string specified in the variable deleteString, but leaving the order of the remaining items unchanged.

Part E (Trees): 45 marks

(a) Draw the *binary search tree* resulting from the insertion of these values in the order given: 3, 1, 4, 59, 26, 535, 58, 9, 79, 32, 34

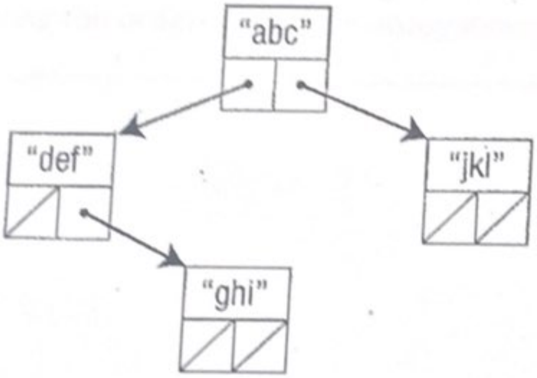
(b) Give the post-order traversal of the node values in this tree.



(c) Consider the following class intended to represent nodes in some binary tree. Line numbers have been added for your reference.

```
1 public class TreeNode {
2     String   item;
3     TreeNode left, right;
4
5     public TreeNode(String item) {
6         this.item = item;
7         this.left = this.right = null;
8     }
9 }
```

Write the Java statements needed to construct the binary tree shown below, including statements to create each node. Assume package-private access to the fields in TreeNode, and **do not** call BinaryTree methods in your answer. The reference to the tree's root must be stored in a variable named `root`. (**Do not** create a method to contain your answer, and **do not** add methods to the class.)



(d) If you ever delete an item from a binary search tree and then insert it back into the tree, will you ever change the shape of the tree? Explain with an example.

Part F (Efficiency): 30 marks

(a) Consider the following Java method *f()*. Do not be concerned with *f's* purpose.

```
 public static void f(int[] theArray) {
    int i, j;
    int key;
    if (theArray.length < 2) {
        return;
    }
    for (int j = 1; j < theArray.length; j++) {
        key = theArray[j];
        for (i = j-1; (i >= 0) && (theArray[i] < key); i--) {
            theArray[i+1] = theArray[i];
    }
    theArray[i+1] = key;

    return;
}
```

How many comparisons involving theArray does *f* perform? How many array-related moves in the worst case? What is the time cost of this code in big O notation? Show all work. Some marks will be given for the quality of your answer.

(b) Why is the worst-case time efficiency for *mergesort* better than the worst-case efficiency for *quicksort*? Explain your answer.

END OF EXAM

This page is for the sole use of examination evaluators.

art A	/20
art B	/50
art C	/20
art D	/35
art E	/45
art F	/30
otal	**/200**