

79
/100

- Closed book, closed note, closed mouth, open eyes!
- Write your own answer legibly. Illegible answers will result in deduction.
- Write answers only on the numbered page. Backside is only for scratch purpose and won't be graded.
- For every question, you must show all your work, including intermediate steps if there's any. Otherwise, you will get some deductions.
- Weight: 30 %
- Before starting, write your name and id below:
Exam number: 1
Class number with section number (circle one): CSC220-001
Term: Winter, 2018
Instructor name: Dr. Manki Min

Student name: *Micah Warren*Student ID: *102-48-375*

1 Indicate whether the statement is true or false (mark as T or F) (2 point each).

- Ans: T 1. A missing semicolon at the end of a Java statement is one of the most common syntax errors.
- Ans: F 2. A .java file (Java source code file) can have more than one public classes defined inside.
- Ans: F 3. The "big-O" notation of an algorithm describes an exact bound of performance.
- Ans: T 4. Stacks are a LIFO data structure.
- Ans: T 5. Queues are a FIFO data structure.

3 Write the letter of the choice that best completes the statement or answers the question (2 points each).

- Ans: C 6. ____ is not used in Java for selection.
 a. if b. else if
 c. elif d. switch
- Ans: A 7. ____ is the keyword used in Java to mean the object is available to all.
 a. public b. open
 c. private d. protected
- Ans: C 8. ____ is the fastest growing complexity.
 a. $O(\log_2(n))$ b. $O(\log_3(n))$
 c. $O(n^{10^{10}})$ d. $O(2^n)$
- Ans: A 9. ____-linked lists maintain two links.
 a. Bidirectionally b. Doubly
 c. Dually d. Circular
- Ans: A 10. Stacks are not used for ____.
 a. user memory management by OS b. ~~reversing the word~~
 c. linked lists d. ~~postfix-expression-evaluation~~

7 Write the word or phrase that best completes each statement or answers the question (4 points each).

11. In Java, a(n) ____ class is a class that cannot be instantiated.

Ans: private

12. ADT stands for ____.

Ans: _____

13. ____-linked lists can have a reverse traversal algorithm with $O(1)$ complexity.

Ans: Bidirectionally

14. LIFO stands for ____.

Ans: Last in First out

15. FIFO stands for ____.

Ans: First in First out

16. In the space below, mark the time analysis, $T(n)$, for each loop and identify the overall complexity, $O()$ (10 points).

```
for (i=0; i<2*n; i++)  
{  
    for (j=1; j<=n; j*=4)  
    {  
        ...  
    }  
}
```

 $2n$ $n/4$

$$2n \cdot \frac{1}{4}$$

$$\frac{1}{2} n^2$$

$$O(n^2)$$

17. Write a Java code snippet that declares an integer type two dimensional array A (with 10 rows and 5 columns) and assigns 10 to the second row and the first column of A (10 points).

```
public class test {  
    public static void main(String args) {  
        int[10] A = new int[10][5];  
        A[1][0] = 10  
    }  
}
```

18. Consider the following array of integers, arr:

0 1 2 3 4 5 6 7 8 9 10

2	5	6	7	10	13	23	32	33	57	98
---	---	---	---	----	----	----	----	----	----	----

Conduct a binary search on this array for the value 5 (10 points). For each iteration of the while loop, show the current value of first, last, mid, and arr[mid] in the table below. Also show the value returned below. You may use the provided code below to assist you.

Iteration	first	last	mid	arr[mid]
0	0	10	5	13
1	0	4	2	6
2	0	2	1	5

Value returned: ~~-1~~ ?

```

public int BinarySearch(int[] list, int item)
{
    int first = 0;
    int last = LIST LENGTH - 1;
    int mid;
    while (first <= last)
    {
        mid = (first + last) / 2;
        if (list[mid] == item)
            return mid;
        if (list[mid] > item)
            last = mid - 1;
        else
            first = mid + 1;
    }
    return -1;
}

```

19. Write the pseudocode of removing the head node of a linked list with two links (prev and next) (10 points). Consider every possible case.

head node Link = head
number of items -= 1

20. Write the definition of the "Dequeue" method in the space between ...'s of the following Java code (10 points).

// Node<T> is defined with getData/setData, getLink/setLink methods.
 // Queue<T> is defined with Dequeue, Peek, Size, IsFull, IsEmpty, Equals methods.

```
public class Queue<T>
{
    public static final int MAX_SIZE = 50;

    private Node<T> head;
    private Node<T> tail;
    private int num_items;
```

```
... int Dequeue() {
    int value;
    head.getData() = value;
    head = head.getLink();
    num_items -= 1;
    return value;
}
```

```
...
}
```

21. Convert the following infix expression into a postfix equivalent using two queues (for infix and postfix) and a stack (for operators) and evaluate the postfix equivalent using a stack (10 points). Explain each step.
 $3 \times ((4 + 5) - 3) / (1 + 2)$

Handwritten work showing the conversion of the infix expression $3 \times ((4 + 5) - 3) / (1 + 2)$ to postfix and its evaluation.

Conversion to Postfix:

- Initial stack: 3
- Read $($: Push to stack. Stack: 3
- Read 4 : Push to stack. Stack: $3, 4$
- Read $+$: Push to stack. Stack: $3, 4, +$
- Read 5 : Push to stack. Stack: $3, 4, +, 5$
- Read $)$: Pop 5 , then $+$. Postfix: $4 5 +$. Stack: 3
- Read $-$: Push to stack. Stack: $3, -$
- Read 3 : Push to stack. Stack: $3, -, 3$
- Read $)$: Pop 3 , then $-$. Postfix: $4 5 + 3 -$. Stack: 3
- Read $*$: Push to stack. Stack: $3, *$
- Read 1 : Push to stack. Stack: $3, *, 1$
- Read $+$: Push to stack. Stack: $3, *, 1, +$
- Read 2 : Push to stack. Stack: $3, *, 1, +, 2$
- Read $)$: Pop 2 , then $+$, then 1 . Postfix: $4 5 + 3 - 1 2 +$. Stack: $3, *$
- Read $/$: Push to stack. Stack: $3, *, /$
- Read 3 : Push to stack. Stack: $3, *, /, 3$
- Read $)$: Pop 3 , then $/$. Final postfix: $4 5 + 3 - 1 2 + 3 /$

Evaluation of Postfix:

- Stack: 4
- Stack: $4, 5$
- Stack: $4, 5, +$
- Stack: $4, 5, +, 3$
- Stack: $4, 5, +, 3, -$
- Stack: $4, 5, +, 3, -, 1$
- Stack: $4, 5, +, 3, -, 1, 2$
- Stack: $4, 5, +, 3, -, 1, 2, +$
- Stack: $4, 5, +, 3, -, 1, 2, +, 3$
- Stack: $4, 5, +, 3, -, 1, 2, +, 3, /$
- Final result: 6

Handwritten notes include "postfix?" and "6" at the bottom.

22. Evaluate the following infix expression which is assumed to be stored in "infix" queue, using only one stack without first converting it into a postfix equivalent (**BONUS 10 points**). Explain each step.
- $$3 \times ((4 + 5) - 3) / (1 + 2)$$

6