CS 14 – Midterm
100 points possible

For all questions, lg N or log N is the log base 2 of N.
For all questions, assume all ADT implementations and algorithms are good implementations.

Do NOT make any stray marks on your answer sheet. You may use pen or pencil, but if you want to change an answer, you MUST erase completely. Marks that are "X"ed out will be counted as filled. Your answer sheet will be scanned and graded automatically.

True/False questions. Please mark A for True and B for False on the answer sheet provided: (1.5 points each – 18 points total) *(Average 30 seconds per question)*

1. A friend function is considered a class member function

2. A derived class can redefine an inherited member function of its base class

3. Late binding is a good example of polymorphism

4. If an exception is thrown but not caught, the program will ignore the exception and continue with execution.

5. A stack reverses the order in which items occur.

6. A queue preserves the order in which items occur.

For the following two questions, suppose $f(N) = O(h(N))$ and $g(N) = O(h(N))$.
7. Is $f(N) + g(N) = O(h(N))$?

8. Is $f(N) * g(N) = O(h(N))$?

---

9. If x and y are real numbers such that $0 < y < x$ then n^x is $O(n^y)$ but n^y is not $O(n^x)$

10. f(N) = N has a faster growth rate than g(N) = lg N

11. The size () operation for a singly linked list can never run in O(1) time because every node in the list must be traversed to count how many nodes are in the list.

12. If class X publically inherits class Y, the following code is legal:

```
X x;
Y y;
X* xPtr= &y;
```

Multiple choice questions. Please mark the answer on the answer sheet provided: (2 pts each) – 46 points total *(Average 30 second per question)*

13. A derived class inherits private members from the base class and

A. the derived class has unrestricted access to the private members of the base class
B. the derived class can read but not modify the private members of the base class
C. the derived class can modify but not read the private members of the base class
D. the derived class cannot directly access the private members of the base class
E. None of the above
F. All of the above

14. In the template prefix

template <class T>

what kind of variable is the parameter T?

A. T must be a class
B. T must not be a class
C. T can be only types built into the C++ language
D. T can be any type, whether built into the C++ language or defined by the programmer
E. Both B and C are true
F. Both A and D are true

15. For a large array and large list holding the same type and number of objects, inserting a new object ***at a known*** location into the middle of a linked list compared with insertion in an array is

A. More efficient
B. Less efficient
C. About the same
D. Dependent on the size of the lists
E. Impossible to determine with the information provided.

16. Suppose that you have a sorted array of 500 integers and suppose also that you want to use binary search to find a specific value. What is the maximum number of array elements that you actually have to look at until you find the value that you are looking for (or until you can tell for sure that the value you're looking for isn't in the array)?

A. 8
B. 9
C. 10
D. 500
E. Using binary search in this case is a bad idea.

17. Which of the following is a limitation/are limitations of a pointer-based implementation of the ADT List? Mark one answer only.

A. The list will have a fixed maximum size determined at compile time.
B. Element retrieval is not done in constant time.
C. Insertions into the list have a worst-case runtime complexity of $O(n^2)$.
D. Both A and B.
E. Both B and C.

18. The operation enqueue() affects

A. the front of the queue.
B. the back of the queue.
C. both the front and the back of the queue.
D. either the front or the back of the queue, but not both at the same time, depending on the user's needs.

19. Retrieving the minimum value in a linked list of integers

A. always has a worst-case runtime complexity of O(1).
B. always has a best-case runtime complexity of O(1).
C. always has an average-case runtime complexity of O(1).
D. may never have a worst-case runtime complexity of O(1).
E. None of the above.

20. Which of the options below could be used to indicate the end of a linked list?

A. The data stored in the last node is an end-of-list flag of some kind.
B. The member variable next of the last node is set to NULL.
C. An external pointer pointing to the last node is used.
D. All of the above methods.
E. None of the above methods.

21. The runtime complexity of an algorithm

A. depends on the programming language being used.
B. depends on how the data accessed by the algorithm is organized.
C. depends on the speed of the computer on which the algorithm is being run.
D. All of the above.
E. None of the above.

22. Suppose that you are considering running two algorithms under the same conditions and on the same computer, on problems of size n. Furthermore, suppose that Algorithm A has a worst-case runtime complexity proportional to *log n* while algorithm B has a worst-case runtime complexity proportional to *n*. For large enough values of n,

A. Algorithm A is always faster than algorithm B, in the worst-case scenario.
B. Algorithm A is always slower than algorithm B, in the worst-case scenario.
C. Algorithm A may be faster or slower than algorithm B, in the worst-case scenario, depending on those pesky constants which we typically ignore when we use O-notation.
D. We cannot conclude anything about this situation when n is large; we should be looking at the case when n is small.

23. Suppose I gave you a homework assignment in which you have to implement a hierarchical file system. Which data structure would be most appropriate for this assignment?

A. A graph.
B. A pointer-based list.
C. An array-based stack.
D. A pointer-based queue.
E. A tree.
F. Any of the above are equally appropriate

For the next 6 questions, use the following answer choices:

A.	O(1)	B.	O(N)
C.	$O(N^2)$	D.	O(N log N)
E.	O(log N)	F.	None of the above

Give the worst-case Big-Oh running time of the following.

24. Deleting the last element of a singly linked list with a tail pointer.

25. Linear search on a sorted array.

26. Binary search on a sorted array

27. Printing the elements of an unsorted list in sorted order. (Do not sort the elements or use any auxiliary data structure).

28. insertAfter for a linked list implemented sequence.

29. removeAtRank for an array implemented sequence?

30. Enqueue into a circular array implemented queue

31. Printing the elements in an array

32. Push onto a linked list implemented stack

33.
```
void Ex(int n)
    int a;
    for ( int x = 0; x < n; x+=4 ) a = x;
```

34.
```
void Ex(int n)
    int a;
    for ( int x = 0; x < n*n; ++x )
        for ( int y = 0; y < x+x; ++y ) a = x;
```

35.
```
void Ex(int n)
    int a;
    for (int x = 0; x < n; x++ )
        for (int y = n; y > 1; y /= 2) a = x;

    for (int x = 0; x < 3*n; x++ ) a = x;
```

(Plan to spend no more than 4 minutes on this question)
36. (8 pts) Briefly define the following:

a) Object oriented programming

b) Dynamic/late binding

c) Virtual function

d) Polymorphism

(Plan to spend no more than 3 minutes on this question)
37. (5 pts) Assume that a templatized linked list based stack and node class have been written for you. The stack class has one node pointer to the top of the stack and the node class stores the item and a node pointer to the next node in the class. The stack class is a friend of the node class. Write code for the ***templatized*** push member function of the stack class. The push function will take as a parameter the item to push onto the stack. (Do not write code for the stack or node class, just the push function). Template related syntax will be graded.

(Plan to spend no more than 5 minutes on this question)
38. (6 pts) Given the following snippet of code: (There are no syntax errors in this code)

```
int* x = new int(5);
int y = *x;
int* z = x;
(*x)++;
int* a = 0;
```

Describe what the following statements will produce (either what will be printed or what will happen). If a memory address is printed, be as specific as you can by telling me what variable/value the memory address is associated with.

a) cout << *x << endl;

b) cout << y << endl;

c) cout << *z << endl;

d) cout << a << endl;

e) cout << x << endl;

f) cout << z << endl;

g) cout << &a << endl;

h) cout << &x << endl;

i) cout << &z << endl;

j) cout << *a << endl;

(Plan to spend no more than 7 minutes on this question)

39. (7 pts) Show how to implement a stack using two queues. Describe the push and pop functions. You may assume the Queue has the functions Enqueue, Dequeue, and isEmpty. You do not have to actually write code, you may just describe the method and use pictures, but be sure that you are clear. **Give the Big-Oh running time of the push and pop operations.** *(Source – modification of homework question)*

(You should now have approximately 13 minutes to work on this question)

40. (10 pts) Write a function that will delete from a linked list of integers the node that contains the largest integer. ***You must accomplish this with 1 pass through the list***. Remember to use good programming style. **Give the Big-Oh running of your function.** You may assume the following classes exist:

```
class List {
private:
        Node* head;
public:
        void deleteLargest();
};
```

```
class Node {
friend class List;
private:
        int item;
        Node* next;
};
```