**44-542 Object Oriented Programming Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exam 03 Part 1 (40 points) KEY** *please print*

**Multiple choice (2 points each).**  Write the letter corresponding to the BEST correct answer.

***Select only ONE answer for each question. If you select more than one answer, the entire question will be counted as wrong.***

1. Which of the following is true of complete binary trees?
   1. a complete binary tree is always a full tree
   2. a complete binary tree may not be a two-tree
   3. a complete binary tree must have nodes that are comparable
   4. a complete binary tree is a heap
2. The underlying structure used in the Java **Stack** class is a(n)
   1. **ArrayList**
   2. **LinkedList**
   3. **TreeSet**
   4. **Vector**
3. Which of the following is true of heaps?
   1. a heap is a complete binary tree
   2. in a heap, the contents of the nodes must be comparable
   3. a heap is a full tree
   4. all of the above are true
   5. a) and b) are true
4. Which of the following is true of the peek operation for stacks?
   1. peek returns the top element on the stack
   2. peek removes the top element on the stack
   3. both of the above are true
   4. none of the above are true
5. The root of a tree representing a heap is called the \_\_\_\_\_ of the heap
   1. top
   2. first element
   3. leader
   4. bottom
6. The root interface in the Java Collections Framework is the \_\_\_\_\_ interface.
   1. **Collection**
   2. **Collections**
   3. **Root**
   4. **System**
7. In Java, a **LinkedList** object can be traversed only in the forward direction.
   1. true
   2. false
8. If the last level of a heap is full, you cannot add any more elements to the heap.
   1. true
   2. false
9. Java provides the abstract class \_\_\_\_\_ that implements the **Collection** interface.
   1. **AbsCollection**
   2. **AbstractCollection**
   3. **Collections**
   4. **ArrayList**
10. When using \_\_\_\_\_ to resolve collisions in hash tables, each entry in the hash table is a pointer to a list of items, all of which hash to the same entry in the hash table.
    1. Robin Hood hashing
    2. open addressing with linear probing
    3. hopscotch hashing
    4. cuckoo hashing
    5. chaining
11. Which of the following Java classes allow duplicate elements?
    1. **ArrayList**
    2. **HashSet**
    3. **LinkedList**
    4. all of the above
    5. a)and c) only
12. Linked lists do not allow for direct access.
    1. true
    2. false
13. Which of the following is true of stacks?
    1. all insertions and removals take place at the top of the stack
    2. a stack is a FIFO structure
    3. both of the above are true
    4. none of the above are true
14. Suppose we want to store the following integers in a hash table of size 10, using the hash function h(x) = x mod 10:

**22 32 42 67 68 69**

If these numbers are inserted into the hash table in the order given here, then 32 will be placed in the location indexed by

* 1. 0
  2. 1
  3. 2
  4. 3
  5. 4

1. When using a **TreeSet**, the **add** and **remove** methods are \_\_\_\_\_.
   1. O(1)
   2. O(log(n))
   3. O(n)
   4. O(nlog(n))
2. Which of the following is true of **Map**s in Java?
   1. duplicate keys are not allowed
   2. the **Map** interface extends the **Collection** interface
   3. there are no classes in Java that implement the **Map** interface
   4. use the **add** method to add key-value pairs to a **Map** object.
   5. both a) and d) are true
3. In a tree, the level of the root node is
   1. 0
   2. 1
   3. the same as the height of the tree
4. The height of a full tree is \_\_\_\_\_, where **n** is the number of nodes in the tree.
   1. **log2(n)**
   2. **floor(log2(n))**
   3. **ceil(log2(n))**
   4. **n2**
5. Searching for elements stored in a hash table is an \_\_\_\_\_ operation.
   1. O(1)
   2. O(log(n))
   3. O(n)
   4. O(nlog(n))
6. What rule of thumb is used in regard to the size of the hash table.
   1. The hash table should be 75% of the size of the data set.
   2. The hash table should be the same size as the data set.
   3. The hash table should be 150% of the size of the data set.
   4. The hash table should be twice the size (200%) of the data set.