

CS 2501: DSA1 Weekly Quizzes

Name _____





You **MUST** write your e-mail ID on **EACH** quiz page that you choose to complete. Please put your name on the top of each page you complete as well.

You may complete up to two pages of this quiz booklet. If you complete more than that, we will only grade two of the pages (which two is up to our discretion).

There are 7 pages to this quiz booklet. Once the time starts, please make sure you have all the pages.

This quiz is **CLOSED** text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

*A crash reduces
Your expensive computer
To a simple stone.*

	a b c d e f g h i j k l m n o p q r s t u v w x y z	<div style="background-color: #cccccc; width: 100px; height: 100px;"></div>	Do not write in this area					
	a b c d e f g h i j k l m n o p q r s t u v w x y z		Department	Course	Exam	Page	Version	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	0 1 2 3 4 5 6 7 8 9		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z	<div style="background-color: #cccccc; width: 100px; height: 100px;"></div>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
	a b c d e f g h i j k l m n o p q r s t u v w x y z		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Module 1: Basic Java 1

Your TA will select one of the following short answer questions for you to answer:

- What is the difference between a float and a double (looking for high level description here)?
- Java is strongly typed. What does this mean? Give an example to illustrate your point.
- Describe one difference between primitive and Object types in Java.
- Briefly, what is the Java API and why is it useful?
- What is casting? Provide an example to illustrate your point.

1. [1 points] Answer the question that was selected.

2. [1 points] For this question, write a simple java program that reads in two variables and does one of the following. Your TAs will randomly select one of these options.

- Reads in two Strings from the keyboard and prints them to the console.
- Reads in two integers from the keyboard and prints their sum to the console.
- Reads in two doubles from the keyboard, converts them to integers, and prints the sum of the integers to the console.

```
//IMPORTS AND SUCH HERE
public class QuizQuestion{
    public static void main(String[] args){
        //YOUR CODE STARTS HERE:
```

```
    }
}
```

Module 2: Basic Java 2

Your TA will select one of the following short answer questions for you to answer:

- If an array is passed to a method and its contents altered, will the actual parameter be changed as well? Why or why not?
- True or False: An if statement can contain an expression that evaluates to an integer because Java knows how to treat integers in this scenario. Explain your answer.
- In Java, what happens if you do not include curly braces with your if-statements?
- Briefly describe the difference between how primitives and references are stored in memory.
- Write a short code snippet that produces a shared reference.

3. [1 points] Answer the question that was selected.

4. [1 points] For this question, write a Java method that takes in an array of integers as a parameter and does one of the following. Your TA will select which method you write.

- Compute the third highest integer in the array and print it to the console.
- Print out all of multiples of 10 in the array in backwards order.
- Print out all of the positive, even numbers.

```
public static void quizMethod(int[] a){
```

```
}
```

Module 3: Basic Java 3

Your TA will select two of the following short answer questions for you to answer:

- When writing our **Card** class in lecture. What was the purpose of the *constructor*? Be as precise as you can here.
- What is an enum? Give an example of when an enum is useful and describe the advantage of using one.
- Why is it important to write the `.equals()` method when writing a class? Give a concrete example.
- Describe why you might choose to make a field in your class private (instead of public).
- Write a small Point class. The class should contain two integer fields (`x` and `y`), a constructor that sets `x` and `y`, and a method called *distance(Point other)* which returns the distance between this point and the other point.
- In lecture, we wrote/saw a **Deck** class. Write a short *equals* method for the Deck class. Two decks are equal if and only if the array of cards contains the same cards in exactly the same order and the top indicator is equal as well.

5. [1 points] Answer the first question that was selected.

6. [1 points] Answer the second question that was selected.

Module 4: Vectors

Your TA will select one of the following short answer questions for you to answer:

- Describe the difference in efficiency of inserting at the front of a Vector versus the back. Are there any special cases involved?
- Verbally describe how removing from the end of a Vector works? In other words, how do we model the removal?
- List one strength and one weakness of using a Vector. Please describe these in sufficient detail.
- What is polymorphism? Why is it useful? Describe using the example of List and Vector from class.
- How efficient is it to grab the item at a specific index of a Vector? How about to find a specific item in the Vector (e.g., is 10 in the Vector?). Briefly describe why these are different.

7. [1 points] Answer the question that was selected.

8. [1 points] For this question, write one of the methods for a Vector class. Your TA will select one of the following methods: insert(T data), remove(T data), insertAt(int index, T data), or resize().

```
public class Vector<T> implements List<T>
    private T[] data;
    private int size = 0;
    private static final int INITIAL_CAPACITY = 100;

    //TODO: WRITE THE CHOSEN METHOD HERE.
```

Module 5: Linked Lists

Your TA will select one of the following questions for you to answer:

- Describe the runtime of these Linked List operations: insert at head, at tail, remove at head, at tail.
- Describe why retrieving the item at index i is slower with a Linked List than with a Vector or Array.
- Describe the difference in how Linked Lists are laid out in memory versus Arrays. Be as precise as possible.
- Suppose you have a doubly-linked-list in which each node stores one char of a word. Write psuedo-code describing a method that returns true if this linked list is a palindrome (e.g., racecar).
- Suppose you have a singly-linked list (head pointer + next pointers). Write psuedo-code for a method that reverses the order of the list (change references here, not the contents of the nodes).

9. [1 points] Answer the question that was selected.

10. [1 points] For this question, write one of the methods for a LinkedList class. Your TA will select one of the following methods: insertAtHead(T data), removeAtTail(), insertAt(int index, T data), get(int index) or find(T data).

```
public class LinkedList<T> implements List<T> {  
    private ListNode<T> head, tail;  
    private int size;  
  
    //TODO: WRITE THE CHOSEN METHOD HERE.
```

Module 6: Stacks and Queues

This quiz question is about the *space efficiency* of List and Array based stacks and queues. Your TA will randomly select the following things:

- Whether we are dealing with a stack or a queue
- What type is being stored in the queue (int, char)
- Whether the stack/queue is array-based (int fields where necessary, initial capacity of 100) or linked-list based (doubly-linked, head and tail pointer to dummy nodes).
- How many items have been inserted into the stack/queue ($n \leq 30$)

When answering, remember that integers are 8 bytes, chars are 1 byte, and references are 8 bytes. The size of objects (e.g., ListNode) is just the sum of the size of its fields.

11. [1 points] Given the constraints above, how many bytes of memory does the data structure consume? Do not include the reference to the stack/queue in your total
12. [1 points] For this question, write one of the methods for an array-based queue class. Your TA will select one of the following methods: enqueue(T data), T dequeue(). You can (and should) invoke resize() if necessary.

```
public class Queue<T> implements IQueue<T> {  
    private T[] data;  
    int size, front, back;  
  
    //TODO: WRITE THE CHOSEN METHOD HERE.
```