

**CS-230 Data Structures**

**Ungraded Quiz 2 – 30 Minutes**

True or False (2 points each)

1. Big O notation provides us a way to compare algorithm performance as the number of items (N) grows.
   1. **True**
2. The Binary Search algorithm does not require ordered data
   1. FALSE
3. A class provides a blueprint containing attributes and methods from which objects of the class can be created.
   1. TRUE
4. Polymorphism means “many forms” and refers to the fact that the actual method that gets executed depends on the type of the object at runtime.
   1. TRUE
5. Object Oriented Programming allows the use of access modifiers to protect unwanted access to class and object methods.
   1. TRUE
6. Java has an automatic garbage collection feature which releases memory if there are no references to the data stored in the memory.
   1. TRUE
7. Java treats primitive types differently than objects in regards to reference and data storage.
   1. TRUE
8. Int, Float, and Double types in Java are called primitive types.
   1. TRUE
9. Primitive types in Java are stored in memory without references.
   1. TRUE
10. Java objects use a reference which is not the object itself but an address of the actual object in memory.
    1. TRUE
11. Objects are created using the “new” keyword in Java.
    1. TRUE
12. The Linear Search algorithm involves starting at the middle of the dataset, and then making a determination whether the item under consideration is in the first half of the dataset or the second half of the data set. The algorithm then proceeds to “half” each remaining subset of the data until the value under consideration is found.
    1. FALSE-Binary Search
13. The Bubble Sort algorithm involves a temporary variable used to store a value from the dataset, which makes room for data values to be shifted inside the original array.
    1. FALSE-Insertion
14. Please fill in the Execution Times in Table1.

**Table 1 – Operations and Algorithms and their Execution Time in Big O Notation**

|  |  |
| --- | --- |
| **Operation/Algorithm** | **Execution Time in Big O Notation** |
| Linear Search of an Unordered Array | O(N) |
| Binary Search of an Ordered Array | O(logN) |
| Insertion in an Unordered Array | O(1) |
| Insertion in an Ordered Array | O(N) |
| Deletion in an Unordered Array | O(N) |
| Deletion in an Ordered Array | O(N) |
| Bubble Sort of an Unordered Array | O(N^2) |
| Selection Sort of an Unordered Array | O(N^2) |
| Insertion Sort of an Unordered Array | O(N^2) |

1. You are working on a programming project that requires lots of searching, but very little insertion or deletion. The project is a very large employee database with employee names and other information about the employee. Your application will frequently need to search the database for employees and make updates to their respective records. The application specifications indicate that searching speed is the absolute priority in the design of the product.

Considering the provided specification and description of the application, would you choose to store the data in an ordered array according to last name or unordered array? What type of search would you use to find the employees? (The actual data would most likely be stored in a database, so would you choose to store the data in the database in an unordered fashion or an ordered fashion according to last name? Providing shortcuts to find data in a database easily is called indexing.)