CS 152 Midterm Review Guide

1 Conceptual Questions

1. Understanding Object-Oriented Principles

- What is abstraction in object-oriented programming? What is the smallest and largest unit of abstraction? How is abstraction implemented in Python?
- What is encapsulation? Why is it important? How is encapsulation implemented in Python?
- You are building a video game like Grand Theft Auto using object-oriented programming. One of your first tasks is to design the Vehicle representation. Describe your design using an object-oriented approach.

2. Complexity Analysis

- List the complexity analysis values starting from the most efficient to the least efficient.
- Why is algorithm efficiency analyzed using Big-O notation?
- Describe the performance complexity of:
  - Iterating through a two-dimensional grid of unknown size.
  - Iterating through an array.
  - Sequential search of an array.
  - Accessing an item in an array.

3. Data Structure Design Scenario

- You are designing a sequential data structure that is fast yet flexible for storing ordered elements. You have implemented fixed-size arrays and a bag so far, and you know about stacks and queues.
- Describe the internal structure of your data structure.
- List the functions the ADT should support and their time complexities.

2 Code Question

Task: Implement a static method to merge two Array objects as described below.

Listing 1: Merge two Array objects

```
@staticmethod
def merge(array1: "Array", array2: "Array") -> "Array":
    """
    Merges two Array objects in a single order fashion.
    If either array is longer, merge the items in the back of the
        resulting array.

    Example:
        >>> print(array1)
        [5, 7, 17, 13, 11]
        >>> print(array2)
        [12, 10, 2, 4, 6]
        >>> new_array = Array.merge(array1, array2)
        >>> print(new_array)
        [5, 12, 7, 10, 17, 2, 13, 4, 11, 6]

    Args:
        array1 (Array): The first array to merge.
        array2 (Array): The second array to merge.

    Returns:
        Array: A new array containing the merged elements from
            array1 and array2.

    Raises:
        TypeError: If either array1 or array2 are not Array objects
            .
    """
    pass  # Implementation required
```