Name _____________________

Short Answer (3 Points)

1.) A(n) ___iterator_____ is a class whose purpose is to represent a position within a data structure.

2.) The _____break______ keyword can be used to prematurely terminate a loop.

3.) A(n) _Exception or ArrayIndexOutOfBoundsException_ occurs when we specify an invalid index into an array.

True or False (4 Points - Please circle your answer)

4.) (True / False) - Accessing random elements in a list is efficient.
   Rationale: Inefficient because we have to traverse nodes of the list to reach the random element

5.) (True / False) - Accessing sequential elements in a list can be efficient.
   Rationale: Efficient if we are using an iterator to maintain our position. If we are repetitively using a method such as getNth, which starts from the head of the list each time and traverses nodes to reach the desired position, this is very inefficient

6.) (True / False) - A singly-linked list can be traversed backwards.
   Rationale: There are no references to the previous node in a singly-linked list, so there is no way to traverse the list backwards

7.) (True / False) - A list can store a fixed number of elements.
   This question is true in the most general of senses and false in the strictest sense. The answer I was looking for was false, but I did not take off if you missed this question. The question should read: "A list can only store a fixed number of elements."

Multiple Choice (2 Points - Please circle your answer)

8.) Which of the following can be said about the following array? Select all that apply.

   ```java
   int[][] nums = { { 96, 99, 98, 85 },
                   { 85, 100 },
                   { 100, 67, 91, 93, 78 } };  
   ```

   a.) It is a multidimensional array.
   b.) It is a jagged array.
   c.) We can use nums[i].length to see how many elements are in each sub-array.
   d.) It is an array of int arrays.
Rationale: All answers are true of the array. It is multidimensional because it has two dimensions, it is jagged because all of its dimensions do not have the same length, we should always use the length property on a sub-array that we think might be jagged when traversing it, and finally, it is an array of int arrays.

Short Answer (2 Points)

9.) Use the continue keyword to prevent a null pointer exception in the following loop.

```java
Student[] students = { null, null, new Student() };  
for (Student student : students) {
    if (student == null) {
        continue;
    }  
    System.out.println(student.getGPA());
}
```