15-121
Fall 2022 Midterm Exam
September 13, 2022

Name:

Andrew ID:

- You may not use any books, notes, or electronic devices during this exam.
- Show your work on the exam to receive credit.
- You may complete the problems in any order you'd like; you may wish to start with the free response problems, which are worth most of the credit.
- All code samples run without crashing unless we state otherwise.
- Assume any imports are already included as required.

Don't write anything in the table below.

Question	Points	Score
1	9	
2	10	
3	16	
4	16	
5	13	
6	20	
7	16	
Total:	100	

1. **Short Answer and Multiple Choice**

(a) (3 points) What is the output of the following?

```
public class IncrementorExercise {
    public static void main(String[] args) {
        int a = 6;
        int b = 7;

        System.out.println(b-- - --b - b++ + --a - --b - ++a);
        System.out.println(a);
        System.out.println(b);
    }
}
```

(b) (2 points) Which one of the following is an example of a checked exception?

- ◯ `NumberFormatException`
- ◯ `NullPointerException`
- ◯ `FileNotFoundException`
- ◯ `ArithmeticException`
- ◯ `ArrayIndexOutOfBoundsException`

(c) (2 points) What is the difference between a `null` String and an empty String?

(d) (2 points) Setting an instance variable `private` means that it cannot be directly accessed from outside the class. Why would a programmer want to do this?

2. (10 points) **Code Tracing**

Consider the following two classes, and indicate what is printed when the main method of the B class is executed. Place your answer (and nothing else) in the box under the code.

```
public class A {
    protected int a;
    protected int b;

    public A(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public int mystery() {
        return a-b;
    }

    public void printVals() {
        System.out.println("Cat");
    }
}

public class B extends A {
    public B(int a, int be) {
        super(be, a);
        System.out.println("P1: " + a);
        System.out.println("P2: " + b);
        b = a - 2;
        a = be + 2;
        System.out.println("P3: " + this.a);
        System.out.println("P4: " + this.b);
    }

    public void printVals() {
        System.out.println("Dog");
    }

    public static void main(String[] args) {
        B bob = new B(7, 14);
        System.out.println(bob.mystery());

        A andrew = bob;
        andrew.printVals();
    }
}
```

3. (16 points) **Big-Oh**

Consider the following class containing four methods. Determine the big-oh runtime of each of the methods, in terms of N, the number of items in `list`. Write your answer, and nothing else, in the box next to each method.

```
public class BigOh {
    private ArrayList<Integer> list;

    public BigOh() {
        this.list = new ArrayList<Integer>();
    }

    public void sneakyInsert(int item) {
        if (list.size() == 0) {
            // insert item at the front of the list.
            list.add(0,item);
            return;
        }

        for (int i = 0; i < list.size(); i++) {
            if (item <= list.get(i)) {
                // insert item at location i
                list.add(i, item);
                return;
            }
        }
        list.add(item);
    }

    // Assume that arr contains N items.
    public void sneakyInsertArray(int[] arr) {
        for (int e : arr) {
            this.sneakyInsert(e);
        }
    }

    public int f1() {
        for(int i = 0; i < this.list.size(); i += 2) {
            if (this.list.get(i) % 2 == 0) {
                // Set the value at the i'th location to 0.
                this.list.set(i, 0);
            }
        }
        return -1;
    }
}
```

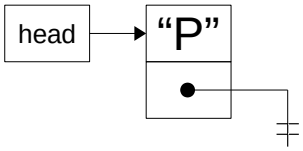
4. **Linked List Memory Diagram**

Consider the following program that creates a linked list. You may assume that the ListNode class exists and was defined as in class.

```
1  public class Quiz6CT1 {
2      public static void main(String[] args) {
3          ListNode<String> head = null;
4          ListNode<String> a = new ListNode<String>("B");
5          ListNode<String> tmp = null;
6
7          head = new ListNode<String>("P");
8          tmp = new ListNode<String>("T");
9          tmp.next = a;
10         head.next = tmp;
11         tmp = new ListNode<String>("X");
12         tmp.next = head.next;
13         head.next = tmp;
14         tmp = tmp.next;
15         tmp.next = new ListNode<String>("L");
16         tmp.next.next = a;
17     }
18 }
```

Draw the state of the linked list after the execution of each specified line of code. (The linked list is defined as starting at `head`.) The first one is drawn for you.

(a) After Line 7



(b) (4 points) After Line 9

(c) (4 points) After Line 10

(d) (4 points) After Line 13

(e) (4 points) After Line 16

5. **Free Response:** Online Courses

Consider the following definition of the `Course` class:

```
public class Course {
    private String number;
    private String title;
    private ArrayList<Student> roster;
    private String roomNumber;

    public Course(String number, String title, String roomNumber) {
        this.number = number;
        this.title = title;
        this.roster = new ArrayList<Student>();
        this.roomNumber = roomNumber;
    }

    public String getNumber() {
        return this.number;
    }

    public String getTitle() {
        return this.title;
    }

    public ArrayList<Student> getRoster() {
        return this.roster;
    }

    public String getLocation() {
        return this.roomNumber;
    }
}
```

This question continues on the next page. Do not write any answers on this page.

(a) (8 points) An `OnlineCourse` is-a `Course`, but it also has a url (stored as a string) that contains a link to the Zoom meeting where lectures take place. When `getLocation` is called on an `OnlineCourse`, it returns a string containing both this Zoom URL and the traditional room number associated with the course. (`OnlineCourse`s still have traditional room numbers, because the professor needs a place to stream the lecture from.)

Write the `OnlineCourse` class. For full credit, you must use proper object-oriented design, including good inheritance and avoiding unnecessary code duplication.

(b) (5 points) Write the public static method `numOnlineCourses` which, given an `ArrayList` of `Course`s, returns how many of them (a number) are `OnlineCourse`s.

6. **Free Response:** Basic Gradebook

   In this question you will write the code for a very basic `Gradebook` class. The information for the gradebook will be read from a file and stored as a list of `GradebookEntry`s.

   (a) (10 points) Let's start with the `GradebookEntry` class. Fill in the code in the appropriate two methods of the class:

```
public class GradebookEntry {
    private String andrewId;
    private ArrayList<Integer> grades;

    public String getAndrewId() {
        return this.andrewId;
    }

    /**
     * Constructs a new gradebook object.
     *
     * @param andrewId The andrewId of this user
     * @param grades All of the numeric scores of this user
     */
    public GradebookEntry(String andrewId, int[] grades) {
        // Your code goes below here









    }

    /**
     * Calculate the average score of `grades`.  Uses a simple mean.
     *
     * @return The mean of all the scores in `grades`.
     */
    public double getAverage() {
        // Your code goes below here









    }

    public String toString() {
        return this.andrewId + " ("+this.getAverage()+") " + this.grades + "\n";
    }
}
```

(b) (10 points) Now we'll write the `Gradebook` class, which contains an `ArrayList` of `GradebookEntry`s. Consider the following sample input file, `gradebook.txt`:

```
rdriley@andrew.cmu.edu:10,20,40
joe@joe.com:10
joe@andrew.cmu.edu:1,2,3,4,5,6,7,8,9,10
```

Also consider the following test code:

```
Gradebook g = new Gradebook("gradebook.txt");
System.out.print(g);
```

Which outputs:

```
rdriley@andrew.cmu.edu (23.333333333333332) [10, 20, 40]
joe@joe.com (10.0) [10]
joe@andrew.cmu.edu (5.5) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Write the `Gradebook` class. You must make proper use of the `GradebookEntry` class you already wrote.

If you need it, there is additional space on the next page.

Additional space for question 6(b).

7. **Free Response:** Linked Lists

Consider the following partial code for a Linked List:

```
public class MyLinkedList<ListType> {
    public ListNode<ListType> head = null;

    /**
     * Constructor that creates a new list containing all of the elements of `arr`,
     * in the same order that they appear in `arr`.
     *
     * Hint: Since this is a constructor, you are guaranteed that the linked list
     * has no items in it already.
     *
     * This function must run in O(N) time.
     *
     * @param arr An ArrayList of items to be added to the new linked list.
     */
    public MyLinkedList(ArrayList<ListType> arr) {
        // You will write this code
    }

    /* Add a node to the head of the list in O(1) time. */
    public void addHead(ListType item) {
        // Code omitted, but assume this method works properly.
    }

    /* Add a node to the end of the list in O(N) time. */
    public void add(ListType item) {
        // Code omitted, but assume this method works properly.
    }

    /**
     * Adds copies of all of the items from `otherList` to the *end* of this list,
     * in order.
     *
     * This function must run in O(N) time.
     *
     * @param otherList The linked list containing items we want copied into this
     *                  list.
     */
    public void mergeList(MyLinkedList<ListType> otherList) {
        // You will write this code
    }

}

public class ListNode<NodeType> {
    private NodeType data;
    public ListNode<NodeType> next;

    public ListNode(NodeType data) {
        this.data = data;
    }

    public NodeType getData() {
        return this.data;
    }
}
```

(a) (8 points) Write the code for the new constructor:

```
public MyLinkedList(ArrayList<ListType> arr) {
```

(b) (8 points) Write the code for the method `mergeList`.

```
public void mergeList(MyLinkedList<ListType> otherList) {
```