CS310 Midterm Exam, Nov. 2, 2020 NAME:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

University ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Open books, handouts, solutions, including online materials, JDK API. No person-to-person communication is allowed, except to ask me questions by email to elizabeth.oneil@umb.edu. Please attend class and have audio and video on. Submit this exam to Gradescope by 8:30pm. If you can’t submit it for some reason, email it to me in pdf format. Please check your email occasionally in case I need to provide a correction. Show all work in the provided areas, expanding them if needed. Each problem is worth 20 points, for a total of 100 points.

1. **Choosing the right JDK Collections class**. For the following scenarios, choose the appropriate JDK Collections classes to use from those we studied in detail: List, Set, Map. Use the simplest class (first on the list here) that has the power you need.

a. You are a grader and decide to use a program to help you manipulate homework scores for students. Each student has a score (integer) for each of “hw1”, “hw2”, “pa1”, etc. How can you hold all the scores for one student so it is easy (one method call) to find out the score for hw1 or pa1, or whatever?

JDK Collections API concrete class, including appropriate generic types:

How to create one such (empty) container:

1. In the same scenario, you realize you want to control the exact order of display of data, so that “hw1” is always first, “hw2” is second, etc. How can you hold just this particular ordering of these strings? (Don’t worry about holding the scores here.)

JDK Collections API concrete class, including appropriate generic types:

How to create one such (empty) container:

c. The students themselves are ID’d by StudentID, an integer that fits in 32 bits. For the last programming assignment in the class, you have set up teams of students. How should you represent a single programming team of students?

JDK Collections API concrete class, including appropriate generic types:

How to create one such (empty) container:

d. Suppose each team has chosen a string name to identify it. How can you hold not only all these team names but also the membership of each named team?

JDK Collections API concrete class, including appropriate generic types:

How to create one such (empty) container:

1. **Big-Oh analysis.** Consider this Java method:

public static int f(int n)

{

int i, j, sum = 0;

for (i=0; i < n; i++) /\* loop 1 \*/

sum += i\*i;

for (i=0; i < n; i++) /\* loop 2 \*/

for (j=0; j < 3; j++) /\* loop 3 \*/

sum += i\*j;

return sum;

}

Let T(n) be the running time of this method in terms of its argument n. Analyze the contributions to T(n) from:

* 1. Loop 1
  2. Loop3, for a single value of i.
  3. Loop2, the whole double loop.
  4. The whole function.

1. **What JDK Collections objects can and can't do**.  Which of the following things can be done by pure use of the API as defined in the Java Collections?  If you can do it, outline how, else say that it is impossible.
2. The program has set up a Map from String to Integer, i.e. Map<String, Integer>. Now it wants to make two different keys, “x” and “y” map to the same Integer 6.
3. The program has set up a Map from String to Integer. Now it wants to make one key, “x”, map to two different Integers 6 and 7.
4. A program has added 10 elements to a Set<String> and no longer has individual references to them. Now it wants to determine which element was the first one added to the set.
5. The program has set up a List of Strings, List<String>, and added several elements to it. Now it wants to determine the string that comes first in dictionary order.
6. **A Map/Set App.** You are helping the post office automate the sorting of mail into postal routes, corresponding to the actual bags of mail given to letter carriers. The address scanning machines pick up address strings off the letter's address, for example the zip code, that each specify a certain set of postal routes. The crucial construct you need is a Map from these address strings to Sets of postal route numbers (themselves integers). Here is an example simplified Map:

"Cambridge" 🡪 { 201, 202, 203, 404, 405 }

"Boston" 🡪 {500, 501, 502, 503, 504}

"02138" 🡪 { 201, 203} (a zipcode entirely in Cambridge)

"Ware St." 🡪 {203} (a tiny street in Cambridge)

1. Give the code for constructing an empty instance of this Map, using appropriate generic types:
2. Give the code (3 lines of code, for example) for adding the given information for “Ware St.”
3. For address strings with O(1) postal routes, what is the performance of looking up the address string in the map? Discuss any dependence on what Map implementation is chosen, that is, HashMap vs. TreeMap. What is N here?
4. If the machine picks up two strings off the same envelope, the Map provides 2 sets of route numbers. How should we combine these two sets to get an answer set of route numbers consistent with both strings? In particular, what Set method should we use?

1. **Interfaces.** Consider the Stack API from pg. 121:

public class Stack<Item> implements Iterable<Item>

Stack() *create an empty queue*

void push(Item item) *add an item*

Item pop() *remove the most recently added item*

boolean isEmpty() *is the stack empty?*

int size() *number of items in the stack*

* 1. Write a Java interface for the methods of this API, with name StackIf (If for interface, saving “Stack” for the implementing class). Start it with “public interface Stack<Item> extends Iterable<Item>” to handle the Iterable part. This will make StackIf a subtype of Iterable<Item>.
  2. What can you add to the top line of the source for class Stack of page 149 to say it complies with this interface?
  3. Recall our Bag implementation using LinkedList in homework 2. That approach is possible here too, to implement Stack.java using LinkedList or ArrayList. The usual way is to push onto the end of the list and pop from there too. What List method(s) would you use for push (just give method names)? What List method(s) are needed for pop? Include any methods from any iterator obtained from the List.