DATA STRUCTURES

COMPETENCIES:
4021.1.1: INTRODUCTION TO ABSTRACT DATA TYPES, ALGORITHMS, AND DATA STRUCTURES USING BAGS (UNORDERED LISTS) *- THE GRADUATE EXPLAINS HOW TO DESIGN ABSTRACT DATA TYPES (ADTS), DATA STRUCTURES TO REPRESENT AN ADT IN STORAGE, ALGORITHMS TO MANIPULATE ADTS (USING THE BAG ADT AS AN EXAMPLE), DESCRIBES BAG DATA TYPES, AND THE USE OF BOTH SEQUENTIAL AND LINKED ALLOCATION TO IMPLEMENT THEM.*
4021.1.2: INTRODUCTION TO ANALYSIS OF ALGORITHMS *- THE GRADUATE ANALYZES THE TIME AND SPACE COMPLEXITY OF BASIC ALGORITHMS.*
4021.1.3: STACKS, QUEUES, AND DEQUES *- THE GRADUATE DESCRIBES DESIGN, SPECIFICATION, IMPLEMENTATION OF STACKS, QUEUES, AND DEQUES, AND IMPLEMENTS A SIMPLE APPLICATION USING SEQUENTIALLY ALLOCATED QUEUES AS WELL AS STACKS THAT EMPLOY A LINKED ALLOCATION.*
4021.1.4: LISTS *- THE GRADUATE DESCRIBES DESIGN, SPECIFICATION, AND IMPLEMENTATION OF LIST STRUCTURES.*
4021.1.5: SORTING AND SORTED LISTS *- THE GRADUATE IDENTIFIES BASIC SELECTION AND INSERTION SORTING ALGORITHMS, AS WELL AS FASTER SORTING ALGORITHMS, AND DESCRIBES THE DESIGN AND IMPLEMENTATION OF SORTED LISTS.*
4021.1.6: BASIC SEARCHING ALGORITHMS AND ASSOCIATIVE ADTS *- THE GRADUATE DESCRIBES HOW TO USE SEARCHING ALGORITHMS FOR LISTS, AND EXPLAINS THE CONCEPT OF A DICTIONARY AS AN ASSOCIATIVE ADT.*
4021.1.7: HASHING ALGORITHMS AND STRUCTURES *- THE GRADUATE DESCRIBES HASH TABLES, BUCKET HASHING, THIS USE AS AN EFFICIENT WAY TO IMPLEMENT AN ASSOCIATIVE ADT, AND IMPLEMENTS A SIMPLE APPLICATION THAT USES BUCKET HASHING.*
4021.1.8: TREE STRUCTURES *- THE GRADUATE DESCRIBES TREE STRUCTURES AND BINARY TREES, AND IMPLEMENTS A SIMPLE APPLICATION INVOLVING BUILDING AND SEARCHING A BINARY TREE.*

INTRODUCTION:

FOR THIS PERFORMANCE ASSESSMENT, YOU WILL DEMONSTRATE YOUR ABILITY TO APPLY THE USE OF THE ALGORITHMS AND DATA STRUCTURES STUDIED IN THIS COURSE TO THE SOLUTION OF A REAL PROGRAMMING PROBLEM. YOU WILL IMPLEMENT AN ONLINE ADDRESS BOOK AND COMPARE THE USE OF A BINARY SEARCH TREE AND A BUCKET HASH STRUCTURE BY IMPLEMENTING BOTH FOR THE DESIGN OF THE ADDRESS BOOK.

SCENARIO:

YOUR EMPLOYER IS DEVELOPING AN ONLINE PHONE BOOK THAT STORES NAMES, PHONE NUMBERS, AND E-MAIL ADDRESSES. YOUR BOSS HAS ASKED YOU TO BUILD A PROTOTYPE IN JAVA TO HELP DECIDE ON THE BEST DATA STRUCTURES TO USE FOR THE IMPLEMENTATION. YOU WILL NEED TO IMPLEMENT AND TEST A SINGLE APPLICATION THAT HAS TWO IMPLEMENTATIONS OF THE ADDRESS BOOK BUILT IN SO THAT THE DATA MAY BE STORED USING EACH DESIGN.

REQUIREMENTS:

NOTE: DO NOT USE ANY OF THE DATA STRUCTURES PROVIDED BY THE JAVA API (E.G., JAVA.UTIL.ARRAYDEQUE, JAVA.UTIL.COLLECTIONS, JAVA.UTIL.HASHMAP, JAVA.UTIL.HASHSET, JAVA.UTIL.HASHTABLE). YOU MUST BUILD DATA STRUCTURES FROM THE "GROUND UP."

NOTE: YOU MUST DESIGN, WRITE, IMPLEMENT, AND DEBUG ALL CODE THAT YOU TURN IN FOR THIS ASSESSMENT. CODE DOWNLOADED FROM THE INTERNET OR ACQUIRED FROM

ANOTHER STUDENT OR ANY OTHER SOURCE MAY NOT BE SUBMITTED AND WILL RESULT IN AUTOMATIC FAILURE OF THE ASSESSMENT.

NOTE: YOU WILL NEED THE LATEST JAVA SE DEVELOPMENT KIT (JDK). THIS INCLUDES THE JAVA RUNTIME ENVIRONMENT (JRE) AND COMMAND LINE DEVELOPMENT TOOLS THAT ARE USEFUL FOR DEVELOPING APPLETS AND APPLICATIONS.

A. DEVELOP A HASH TABLE DATA STRUCTURE FROM THE GROUND UP THAT DOES NOT USE ANY OF THE DATA STRUCTURE CLASSES AND THAT INCLUDES THE FOLLOWING:

NOTE: FOR ALL INSERTIONS, DELETIONS, AND LOOK-UPS, THE SUGGESTION IS TO APPLY AN UPPER CASE FUNCTION TO CONVERT ALL LETTERS IN BOTH THE FIRST AND LAST NAME TO UPPER CASE AND CONCATENATE TO FORM A SINGLE NAME STRING.

1. AN INSERTION FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT INSERTS THE DATA ELEMENT INTO THE HASH TABLE:
   - FIRST NAME
   - LAST NAME
   - E-MAIL ADDRESS
   - PHONE NUMBER
2. A DELETION FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT DELETES THE CORRESPONDING DATA ELEMENT FROM THE HASH TABLE:
   - FIRST NAME
   - LAST NAME
3. A LOOK-UP FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT RETURNS THE CORRESPONDING DATA ELEMENT:
   - FIRST NAME
   - LAST NAME
4. IMPLEMENTATION OF 13 BUCKETS IN THE BUCKET HASH STRUCTURE.

NOTE: THE HASH FUNCTION USED FOR THE HASH TABLE MAY CALL THE JAVA NATIVE METHOD HASHCODE() AS PART OF ITS IMPLEMENTATION.

B. DEVELOP A TREE DATA STRUCTURE THAT DOES NOT USE OF ANY OF THE DATA STRUCTURE CLASSES AND THAT INCLUDES THE FOLLOWING:

NOTE: FOR ALL INSERTIONS, DELETIONS, AND LOOK-UPS, THE SUGGESTION IS TO APPLY AN UPPER CASE FUNCTION TO CONVERT ALL LETTERS IN BOTH THE FIRST AND LAST NAME TO UPPER CASE, AND CONCATENATE TO FORM A SINGLE NAME STRING.

1. AN INSERTION FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT INSERTS THE DATA ELEMENT INTO THE TREE:
   - FIRST NAME
   - LAST NAME
   - E-MAIL ADDRESS
   - PHONE NUMBER
2. A DELETION FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT DELETES THE DATA ELEMENT FROM THE TREE:
   - FIRST NAME
   - LAST NAME
3. A LOOK-UP FUNCTION THAT TAKES THE FOLLOWING COMPONENTS AS INPUT AND THAT RETURNS THE DATA ELEMENT FROM THE TREE:
   - FIRST NAME
   - LAST NAME

C. DEVELOP AN APPLICATION THAT TESTS THE FOLLOWING FUNCTIONS:

NOTE: APPLICATIONS COPIED FROM ANOTHER STUDENT OR OTHER SOURCE MAY NOT BE SUBMITTED AND ARE NOT ELIGIBLE FOR PASSING THE ASSESSMENT.

1. INSERTION FUNCTION FOR THE HASH TABLE
2. DELETION FUNCTION FOR THE HASH TABLE
3. LOOK-UP FUNCTION FOR THE HASH TABLE
4. INSERTION FUNCTION FOR THE TREE DATA STRUCTURE
5. DELETION FUNCTION FOR THE TREE DATA STRUCTURE
6. LOOK-UP FUNCTION FOR THE TREE DATA STRUCTURE

D. USE THE ATTACHED TEST CASES TO VERIFY THAT THE HASH TABLE AND TREE DATA STRUCTURE RUN CORRECTLY BY PROVIDING **ONE** SCREENSHOT THAT DEMONSTRATES SUCCESSFUL TEST EXECUTION FOR *EACH* OF THE FOLLOWING:

NOTE: TEST EXECUTION SCREENSHOTS COPIED FROM ANOTHER STUDENT OR OTHER SOURCE MAY NOT BE SUBMITTED AND ARE NOT ELIGIBLE FOR PASSING THE ASSESSMENT.

1. INSERT FUNCTION FOR THE HASH TABLE
2. DELETE FUNCTION FOR THE HASH TABLE
3. LOOK-UP FUNCTION FOR THE HASH TABLE
4. INSERT FUNCTION FOR THE TREE DATA STRUCTURE
5. DELETE FUNCTION FOR THE TREE DATA STRUCTURE
6. LOOK-UP FUNCTION FOR THE TREE DATA STRUCTURE

E. SUBMIT A ZIP FILE THAT INCLUDES THE FOLLOWING:

- ALL OF YOUR .JAVA AND .CLASS FILES
- ALL SCREENSHOTS OF SUCCESSFUL TEST EXECUTION FROM PART D
- COMMENTS ON ALL JAVA CODE THAT ARE CLEAR ENOUGH FOR ANOTHER PERSON TO UNDERSTAND YOUR CODE
- A PREFIX OF YOUR INITIALS ON ALL COMMENTS

NOTE: FOR DEFINITIONS OF TERMS COMMONLY USED IN THE RUBRIC, SEE THE RUBRIC TERMS WEB LINK INCLUDED IN THE EVALUATION PROCEDURES SECTION.

FILE ATTACHMENTS:

DATA STRUCTURES TEST CASES