Colorado State University GLOBAL CAMPUS

CSC400: DATA STRUCTURE AND ALGORITHMS

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

COURSE DESCRIPTION AND OUTCOMES

COURSE DESCRIPTION:

This course provides an overview of data structures including arrays, lists, trees, graphs, hashes, and files. Students will apply techniques to analyze algorithms and to compare data structures.

COURSE OVERVIEW:

Data Structures and Algorithms will focus on understanding key concepts related to data structures and the analysis of algorithms. Data structures is a key computer science discipline that focuses on understanding how to organize data efficiently and effectively. Specifically, this course will present a number of advanced conceptual topics related to software maintainability, efficiency, and algorithm analysis. The topics presented in this course will range from introducing abstract data types (ADTs) such as bags, stacks, queues, deques, and priority queues, to further analyzing the efficiency associated with the ADTs and other algorithms.

COURSE LEARNING OUTCOMES:

1. Implement algorithms for solving problems using Java.
2. Construct a Graphical User Interface.
3. Create an application that utilizes appropriate discrete data structures to solve business problems.
4. Identify abstract data types that can be used in software development.

PARTICIPATION & ATTENDANCE

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical

problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

Course Materials

Required:

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson. ISBN-13: 9780134831695

(Access card is included with this ISBN.)

NOTE: *All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.*

Course Schedule

Due Dates

The Academic Week at CSU-Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and peer responses posted by Sunday 11:59 p.m. MT. Late posts may not be awarded points.
- **Opening Exercises:** Take the Opening Exercise before reading each week's content to see which areas you will need to focus on. You may take these exercises as many times as you need. The Opening Exercises will not affect your final grade.
- **Mastery Exercises:** Students may access and retake Mastery Exercises through the last day of class until they achieve the scores they desire.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

Weekly Reading and Assignment Details

Module 1

Readings

- Prelude (Designing Classes) & Chapter 1 in *Data Structures and Abstractions with Java*
- Segewick, R., & Wayne, K. (2015). 1.1 Programming model. *Algorithms* (4th ed.). Retrieved from https://algs4.cs.princeton.edu/13stacks/

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (30 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Program 1 Pre-Planning (Clothing Boutique)

This assignment will serve as a pre-planner for Program 1 due in Module 2. For this assignment, you are to consider constructing a class that can be used to implement a Bag in Java. Your program will store corresponding items for a clothing boutique. Specifically, your program should consider an item name and price.

The following values might be considered for data:
item_Name = “Red T-Shirt”;
Price = $29.99;

After selecting your values for data, what are the required operations that must be used to create the Bag Interface?

Your paper should meet the following requirements:

- Be two pages in length, not including an APA title page and APA reference page.
- Include at least one reference from the readings or an outside source. The CSU-Global Library is a good place to find your sources.
- Follow the *CSU-Global Guide to Writing and APA.*

Option #2: Program 1 Pre-Planning (Cologne Manufacturer)
This assignment will serve as a pre-planner for Program 1 due in Module 2. For this assignment, you are to consider constructing a class that can be used to implement a Bag in Java. Your program will store corresponding items for a cologne manufacturer. Specifically, your program should consider a Product_Name, Price, size (in ounces).

The following values might be considered for data:
product_Name = “Sea Scent Cologne”;
Price = $79.99;
Size = 4;

After selecting your values for data, what are the required operations that must be used to create the Bag Interface?

Your paper should meet the following requirements:

- Be two pages in length, not including an APA title page and APA reference page.
- Include at least one reference from the readings or an outside source. The CSU-Global Library is a good place to find your sources.
- Follow the *CSU-Global Guide to Writing and APA.*

PORTFOLIO PROJECT REMINDER
Option #1

Review the Portfolio Project assignment in the Module 8 Materials folder. As you complete programming assignments throughout the course, be aware that you will be resubmitting your programs with corrections as graded milestones in Week 5 and in Week 7.

Option #2

Review the Portfolio Project assignment and Portfolio Project Rubric in the Module 8 Materials folder. As you complete programming assignments throughout the course, be aware that you will be resubmitting your programs with corrections as graded milestones in Week 5 and in Week 7.

MODULE 2

Readings

- Chapter 2 & 3 in *Data Structures and Abstractions with Java*
- Bille, P, Christiansen, A. R., Ettienne, M. B., & Gortz, I. L. (2017). Fast dynamic arrays. In K. Pruhs & C. Sohler (Eds.), *Annual European symposium on algorithms* (ESA 2017), article 16, p. 16:1-16-3. Retrieved from http://drops.dagstuhl.de/opus/volltexte/2017/7830/pdf/LIPIcs-ESA-2017-16.pdf

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (60 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Program 1 (Clothing Boutique)

Demonstrate an understanding of working with bags in Java by implementing a class called ClothingBag that implements the BagInterface<MyType> in Java. Your implementation should include the following:

Fields:

- MyType[] myBag;
- int numberofProducts;
- Default_Capacity;

Methods:

- Constructor(): initialize bag
- add() – will be used to add a new entry.
- toArray() – provides an array of the allocated bag.
- isFull() – test whether the bag is at the maximum capacity.

Implement a BoutiqueBagDemo that tests your bag implementation class.

Methods:

- testAddl() – will add the contents to the Bag.
- displayBag() – will display the contents of the Bag.

Ensure that your program has the required class and a test class. Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

Option #2: Program 1 (Cologne Manufacturer)
Demonstrate an understanding of working with bags in Java by implementing a class called FragranceBag that implements the BagInterface<MyType> in Java. Your implementation should include the following:

Fields:

- MyType[] myBag;
- int numberofProducts;
- Default_Capacity;

Methods:

- Constructor(): initialize bag
- add() – will be used to add a new entry.
- toArray() – provides an array of the allocated bag.
- isFull() – test whether the bag is at the maximum capacity.

Implement a FragranceBagDemo that tests your bag implementation class.

Methods:

- testAddl() – will add the contents to the bag.
- displayBag() – will display the contents of the bag.

Ensure that your program has the required class and a test class. Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

MODULE 3

Readings

- Chapter 4 in *Data Structures and Abstractions with Java*
- Tutorialspoint. (n.d.). Design and analysis of algorithms tutorial. Retrieved from https://www.tutorialspoint.com//design_and_analysis_of_algorithms/index.htm

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)
Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Exercises 14, 16, 18 (70 points)
In Chapter 4 of *Data Structures and Abstractions with Java* (Carrano & Henry, 2019, p. 148-149), complete the following:

- Exercise #14
- Exercise #16
- Exercise # 18

For each exercise, show your work and all of the steps taken to determine the Big-Oh for each problem. Partial credit cannot be awarded without showing work. Submit all work in a Word document.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

Option #2: Exercises 15, 17, 19

In Chapter 4 of *Data Structures and Abstractions with Java* (Carrano & Henry, 2019, p. 148-149), complete the following:

- Exercise #15
- Exercise #17
- Exercise # 19

For each exercise, show your work and all of the steps taken to determine the Big-Oh for each problem. Partial credit cannot be awarded without showing work. Submit all work in a Word document.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

MODULE 4

Readings

- Chapter 5 & 6 in *Data Structures and Abstractions with Java*
- Princeton University. (n.d.) *Data structures—2x2* [Class handout]. Retrieved from http://www.cs.princeton.edu/courses/archive/fall09/cos126/lectures/16DataStructures-2x2.pdf

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option # 1: Program 2 (Exercise #8)

Create an appropriate Java Postfix calculator class (Carrano & Henry, 2019, p. 176). Your program should accept at least five elements (data1, data2, data3, data4, data5) and will perform an appropriate postfix operation given the following scenarios:

- data1 data + data3 * data4 –
- data1 data * data3 * data1 – / data4 data5* +

Use the following values for each input:

data1 = 1;
data2 = 2;
data3 = 4
data4 = 5
data5 = 3

Ensure that your program has the required class and a test class. Submit Exercise 8 and screenshots of your program's execution and output compiled into a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

Option # 2: Program 2 (Exercise #9)
Create an appropriate Java Infix calculator class (Carrano & Henry, 2019, p. 176). Your program should accept at least five elements (data1, data2, data3, data4, data5) and will perform an appropriate Infix operation given the following scenarios:

- data1 data + data3 * data4 –
- data1 data * data3 * data1 – / data4 data5* +

Use the following values for each input:

data1 = 1;
data2 = 2;
data3 = 4
data4 = 5
data5 = 3

Ensure that your program has the required class and a test class. Submit Exercise 9 and screenshots of your program's execution and output compiled into a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

MODULE 5

Readings

- Chapter 9 in *Data Structures and Abstractions with Java*
- Ben Gurion University of the Negev. (n.d.) Recursion in Java. Retrieved from https://www.cs.bgu.ac.il/~ipc151/wiki.files/Class_Java_6.pdf

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Program 3 (Exercise #5)

Implement the algorithm outlined in Exercise # 5 (Chapter 9 of Carrano & Henry, 2019, p. 288) in Java pseudocode. Analyze your algorithm in Big-Oh notation and provide the appropriate analysis.

Ensure that your program has the required class and a test class. Compile and submit Exercise 5, the Big-Oh evaluations, and screenshots of your program's execution and output in a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

Option #2: Program 3 (Exercise #6)

Implement the algorithm outlined in Exercise # 6 (Chapter 7 of Carrano & Henry, 2019, p. 288) in Java pseudocode. Analyze your algorithm in Big-O notation and provide the appropriate analysis.

Ensure that your program has the required class and a test class. Compile and submit Exercise 6, the Big-O evaluations, and screenshots of your program's execution and output in a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

Portfolio Milestone (25 points)

Make appropriate corrections to Program 1 (Module 2) and Program 2 (Module 4). Resubmit your programs with corrections (if necessary) that reflect feedback from your instructor**.**

MODULE 6

Readings

- Java Interlude 05 & Chapter 15 in *Data Structures and Abstractions with Java*
- Carnegie Mellon University. (n.d.). Introduction to sorting. Retrieved from http://www.cs.cmu.edu/~clo/www/CMU/DataStructures/Lessons/lesson8_1.htm
- Segewick, R., & Wayne, K. (2015). Algorithms and data structures cheatsheet. *Algorithms* (4th ed.). Retrieved from http://algs4.cs.princeton.edu/cheatsheet/

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Program 4 (Exercise #1)
Implement the algorithm outlined in Exercise #1 (Chapter 15 of Carrano & Henry, 2019, p. 456) in Java. Provide for sorting of an array in ascending and descending order. Explain the steps required to determine the Big-Oh notation for this algorithm.

Ensure that your program has the required class and a test class. Compile and submit Exercise 1, the Big-Oh evaluation, and screenshots of your program's execution and output in a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

Option #2: Program 4 (Exercise #2)
Implement the algorithm outlined in Exercise #2 (Chapter 15 of Carrano & Henry, 2019, p. 456) in Java. Provide for sorting of an array in ascending and descending order. Explain the steps required to determine the Big-Oh notation for this algorithm.

Ensure that your program has the required class and a test class. Compile and submit Exercise 2, the Big-Oh evaluation, and screenshots of your program's execution and output in a single document. Also attach all appropriate source code in a zip file.

Carrano, F. M., & Henry, T. (2019). *Data structures and abstractions with Java* (5th ed.). Upper Saddle River, NJ: Pearson.

MODULE 7

Readings

- Chapter 16 in *Data Structures and Abstractions with Java*
- Big-O Cheat Sheet. (n.d.) Know thy complexities. Retrieved from http://bigocheatsheet.com/

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Portfolio Milestone (25 points)

Make appropriate corrections to Program 3 (Module 5) and Program 4 (Module 6). Resubmit your programs with corrections (if necessary) that reflect feedback from your instructor.

MODULE 8

Readings

- Chapter 7 & 8 in *Data Structures and Abstractions with Java*
- Code Java: Coding Your Passion. (2012-2018). Java queue collection tutorial and examples. Retrieved from http://www.codejava.net/java-core/collections/java-queue-collection-tutorial-and-examples

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Portfolio Project (300 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

Option #1: Final Program

Your Portfolio Project for CSC400 consists of the following:

- Milestone 1 (due in Module 5): Java source code (with corrections if necessary) for programs created in Module 2 and Module 4.
- Milestone 2 (due in Module 7): Java source code (with corrections if necessary) for programs created in Module 5 and Module 6.
- Lessons Learned Reflection
- Final Program

In Week 8, the components left to complete for your Portfolio Project are the **Lessons Learned Reflection** and the **Final Program**. Carefully review the requirements below:

Lessons Learned Reflection:

Write a two- to three-page summary that outlines the lessons you have learned in this programming course. Reflect on how these lessons can be applied towards more effective coding.

Final Program:

Write a program that creates a Person class that contains strings that represent the first and last name of a person and their age. You will need to create a Queue class that will store each person in the queue and can sort the queue based on last name or age.

Prompt the user of the program to add five people to the queue. Your program should provide the contents of the queue and then sort the queue using the quick sort in two ways:

- Descending order by last name.
- Descending order by age.

Assemble your Lessons Learned Reflection, your source code, and screenshots of the application executing and results into a single document. Submit your completed Portfolio Project by the posted due date.

Option #2: Final Program

Your Portfolio Project for CSC400 consists of the following:

- Milestone 1 (due in Module 5): Java source code (with corrections if necessary) for programs created in Module 2 and Module 4.

- Milestone 2 (due in Module 7): Java source code (with corrections if necessary) for programs created in Module 5 and Module 6.
- Lessons Learned Reflection
- Final Program

In Week 8, the components left to complete for your Portfolio Project are the **Lessons Learned Reflection** and the **Final Program**. Carefully review the requirements below:

Lessons Learned Reflection:
Write a two- to three-page summary that outlines the lessons you have learned in this programming course. Reflect on how these lessons can be applied towards more effective coding.

Final Program:
Write a program that creates an Instructor class that contains strings that represent the first and last name of the teacher and the number of courses the instructor teaches. You will need to create a queue class that will store each person in the queue and can sort the queue based on last name or the number of courses teaching.

Prompt the user of the program to add five instructors to the queue. Your program should provide the contents of the queue and then sort the queue using quick sort in two ways:

- Descending order by last name.
- Descending order by number of courses teaching.

Assemble your Lessons Learned Reflection, your source code, and screenshots of the application executing and results into a single document. Submit your completed Portfolio Project by the posted due date.

COURSE POLICIES

Course Grading

20% Discussion Participation
0% Opening Exercises
8% Mastery Exercises
37% Critical Thinking Assignments
35% Final Portfolio Project

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

IN-CLASSROOM POLICIES

For information on late work and incomplete grade policies, please refer to our **In-Classroom Student Policies and Guidelines** or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

Academic Integrity
Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /repurposing your own work (see *CSU-Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and Internet resources.

Citing Sources with APA Style
All students are expected to follow the *CSU-Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, please review the APA resources within the CSU-Global Library under the "APA Guide & Resources" link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement
CSU-Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette
Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults, or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.