Preparing for CS II (CS 143)

Reviewing CS I (CS 141) material

Computer Science is a subject where you typically learn by doing. This means that you also tend to forget by *not doing*. If it has been a quarter or few since you took CS I, it is recommended that you review CS I material prior to CS II. One way that you can do this is by working through several problems in each chapter of Building Java Programs - Chapters 1 through 10 on codestepbystep.com.

It is expected that coming into CS 143 you are *already comfortable with* the following concepts:

- using pseudocode to plan code
- identifying, creating, and using variables, data types, methods (including parameters and returns), conditionals, definite and indefinite loops, arrays, ArrayLists, Object Oriented Programming (making classes and inheritance)
- breaking down problems into logical pieces using methods
- testing and debugging code
- reading and working with unfamiliar code

Java Syntax Refresher

Why Preview CS 143 Material?

Previewing material allows you to to gain exposure to a concept before tackling the computational thinking behind the concept. When you see material at a high level beforehand, you are able to become familiar with vocabulary and reflect on how this new concept relates to the concepts you already know.

The purpose of previewing material is NOT to understand the material at 100%.

The purpose of previewing material is to gain a high level understanding of the concept.

CS 143 Preview Guiding Questions

Below is a collection of guiding questions that you should try to answer as you preview material.

- What is a Collection?
    - List one example of a Java Collection.
- What is Recursion?
    - What is a base case?
    - What is a recursive case?
- Name two searching algorithms.
- Name two sorting algorithms.
- What is Big-O (Efficiency)?
    - Why is it important?
- What is a stack?
    - Name two characteristics of a stack.
- What is a queue?
    - Name two characteristics of a queue.
- What is a linked list?
    - How is a linked list stored differently in memory than a (native) array.
    - What two things are contained inside a List Node?
- What is a tree?
    - What three things are contained inside a Binary Tree Node?

CS 143 Preview Resources

Below you will find links to various resources related to the concepts that you will see in CS143.

- Chapter 11: Java Collections
    - Introduction to Collections
- Chapter 12: Recursion
    - HackerRank - Recursion
- Chapter 13: Searching and Sorting / Big O
    - base.cs podcast - A friendly intro to Big O Notation

- Chapter 14: Stacks and Queues
    - base.cs blog - Stacks and Overflows
        - base.cs podcast - How do you make a stack overflow?
    - base.cs blog - To Queue Or Not To Queue
- Chapter 15: Implementing a Collection Class
- Chapter 16: Linked Lists
    - base.cs podcast - Linked Lists in Your Apartment
- Chapter 17: Binary Trees
    - HackerRank - Binary Trees

Textbook

Over the course of the entire quarter, we will cover the material in Chapters 11 through 17 of:

- Building Java Programs: A Back to Basics Approach by Stuart Reges and Marty Stepp
    - Chapter 11: Java Collection Framework
    - Chapter 12: Recursion
    - Chapter 13: Searching and Sorting
    - Chapter 14: Stacks and Queues
    - Chapter 15: Implementing a Collection Class
    - Chapter 16: Linked Lists
    - Chapter 17: Binary Trees

Publisher Slides of the Textbook

- Building Java Programs Publisher Slides

YouTube videos by Publishers of the Textbook

Below you will find some of the early chapter's publisher resources for the textbook for this course. It is recommended that you watch the videos listed below in Chapters 8 (review from CS 141), 10 (review from CS 141), and 11 before our first class as we will fly through the first couple chapters of material.

- Building Java Programs YouTube Videos
    - Review these to start with...
        - Chapter 8: Objects

        - Chapter 10: ArrayLists
        - Chapter 11: Java Collection Framework

PROCEED AT YOUR OWN RISK - MANY MORE RESOURCES BELOW

If you're interested in more resources, see below. **You do NOT need to preview every single link.** Several are provided so that you can see which types of links work best for you (e.g. maybe you prefer the podcasts or you prefer the wikipedia articles, etc).

Various resource links:

- Chapter 11: Java Collections
    - wikipedia - Java Collection framework
    - Introduction to Collections
- Chapter 12: Recursion
    - wikipedia - recursion
    - video - What is Recursion? - Recursion Explained in 3 minutes
    - HackerRank - Recursion
- Chapter 13: Searching and Sorting
    - wikipedia - search algorithm
    - wikipedia - sorting algorithm
    - base.cs podcast - A friendly intro to Big O Notation
    - Sorting Algorithm Animations
    - HackerRank - Binary Search
    - HackerRank - Mergesort
- Chapter 14: Stacks and Queues
    - wikipedia - stack
    - wikipedia - queue
    - HackerRank - Stacks and Queues
    - base.cs blog - Stacks and Overflows
        - base.cs podcast - How do you make a stack overflow?
        - base.cs podcast - Stacks IRL
    - base.cs blog - To Queue Or Not To Queue
        - base.cs podcast - Queues IRL

    - base.cs podcast - Cut the Queues
- Chapter 15: Implementing a Collection Class
- Chapter 16: Linked Lists
  - wikipedia - linked list
  - base.cs blog - What's a Linked List, Anyway? [Part 1]
  - base.cs blog - What's a Linked List, Anyway? [Part 2]
    - base.cs podcast - Linked Lists in Your Apartment
  - HackerRank - Linked Lists
- Chapter 17: Binary Trees
  - wikipedia - binary trees
  - base.cs blog - How To Not Be Stumped By Trees
    - base.cs podcast - Don't be stumped by ... trees
  - base.cs podcast - Trees IRL
  - base.cs podcast - What is a binary tree?
  - HackerRank - Binary Trees