

# Midterm Review Sheet

This test will consist of 10 questions that are similar in many ways to the questions that you have had on the quizzes. There is also an extra credit question. You will have the full class period to complete the test. Generally I try to lay out tests so they will have 4 short answer questions, 3 coding questions, and 3 tracing questions. The test can include any material that we have covered in class or from the readings. Below is a list of the general topics we have covered and some of the main points that I might expect you to know about them. You will be able to reference the APIs (Scala and Java) and a one page cheat sheet during the exam, but these are the only resources you can use.

## Object-Orientation

- Encapsulation.

- Know what objects are and why we consider them useful.

- Know what you should do in code to use this model more effectively.

## Classes, Objects, and Traits

- You need to understand the basics of class based OOPs.

- What is the function of a class? How does it differ from an object? What is a trait?

## Inheritance in Scala

- What is inheritance?

- Why do we use inheritance?

- Be able to describe the limitations on inheritance in Scala.

## Polymorphism

- How is polymorphism achieved in Scala?

- Be able to write code that uses polymorphism or trace code that uses it.

- Know how parametric polymorphism works in Scala and when it is used.

- Be able to write code that uses parametric types or trace code that uses it.

## Threads

- You should understand what threads are.

- Know the pitfalls of multithreaded code and how to get around them in Scala.

- Be able to write code that would spawn a thread.

- Know the basics of the `java.util.concurrent` package and how to use elements of it.

## Files and Streams

- Know how the stream library is set up in `java.io`.

- Know how we can use it for things like file access, binary data, and object serialization.

## Networking

- Have a basic understanding of networking in Scala with `java.net` libraries.

- You should also understand how this fits in with streams.

## Stacks, Queues, and Priority Queues

- Understand the ADTs for stacks and queues.

- What is LIFO and what is FIFO? Which is what a stack does? What about a queue?

- Know how to implement both of these using arrays.

- How do you implement them using linked lists and how does that compare to arrays?

- Understand how a sorted linked list can be used to implement a priority queue.

- Be able to write or trace these.

## Linked Lists

- Understand what a linked list is.

- Be able to draw pictures showing basic manipulations on linked lists.

- Be able to write the code for the basic methods of a linked list.

- Understand the concepts of circular and doubly linked lists as well as sentinels.