How to Scala (… when things go wrong™)
Announcements

● AI Quiz on Autolab
  ○ Due Wednesday Night (~ 10 minutes)

● PA 0 (aka “set up Git and give us your Github username”) on Autolab
  ○ Due 1 week from today (< 30 minutes)

● PA 1 (aka “Parse CSV files in Scala”) on Autolab
  ○ Due 2 weeks from today
  ○ Submissions open a week from today
  ○ Start Early! (PA 1 template includes starter test cases)
GitHub + Autolab

- Help us open submissions early:
  - We’ll start them once ~90% of Sections A, B submit PA 0
- Allow 24 hours after submitting PA0 before you can submit PA1

![Typical Process Latency Diagram]
Testing
The REPL

- **From IntelliJ**: Ctrl+Shift+D
  - Highlight a line and press Ctrl+Shift+X to execute
  - Copy+paste line and press Ctrl+Enter to execute

- **From the command line**: scala
  - Paste commands in to run them
  - Type :help to get a list of extra commands

- **From SBT**: console
Unit Testing

- Remember breaking the big problem into smaller ones?
  - Test each small solution before combining into a whole

- Useful for debugging
  - Sanity check each step in the process to make sure it works
  - Separate the UI from the tests

- Useful way to encode your assumptions
  - Helpful reminder if your assumptions change later
Basic Debugging
class HelloWorldTest extends AnyFlatSpec {
  "HelloWorld.doThings()" should "return 5" in {
    assert(HelloWorld.doThings() == 5)
  }
  it should "not return 10" in {
    assert(HelloWorld.doThings() != 10)
  }
  "HelloWorld.x" should "have type Float" in {
    assert(HelloWorld.x.isInstanceOf[Float])
  }
  "Register(0).addToValue" should "return the input value" in {
    val reg = Register(0)
    for (i <- 1 to 10000) { assert(reg.addToValue(i) == i) }
  }
}
Debugging with ScalaTest

Demo
Profiling

- IntelliJ → Profilers

- SBT → HProf
  - [https://docs.oracle.com/javase/8/docs/technotes/samples/hprof.html](https://docs.oracle.com/javase/8/docs/technotes/samples/hprof.html)

```scala
fork in run := true
javaOptions in run += "-agentlib:hprof=cpu=samples,depth=10"
```

Load HProf

Tell SBT to start a new process when running

Sample CPU Usage

Stack Trace

Depth of 10
CPU SAMPLES BEGIN (total = 185) Fri Sep 3 02:24:48 2021
rank  self  accum  count  trace method
 1 44.86% 44.86%      83 300207 scala.collection.StrictOptimizedLinearSeqOps.drop
 2 35.14% 80.00%      65 300210 scala.collection.immutable.$colon$colon.tail
 3  5.95% 85.95%      11 300071 java.lang.ClassLoader.defineClass1
 4  2.16% 88.11%       4 300209 scala.collection.immutable.Range.foreach$mcVc$sp

_TRACE 300207:
      scala.collection.StrictOptimizedLinearSeqOps.drop(LinearSeq.scala:261)
      scala.collection.StrictOptimizedLinearSeqOps.drop$(LinearSeq.scala:257)
      scala.collection.immutable.List.drop(List.scala:79)
      scala.collection.immutable.List.drop(List.scala:79)

...
Profiling with HProf

Demo
Submission

- **Github Classroom**
  - You will get an invitation link for each individual project.
  - Clicking the invitation will create a GitHub Git repository for the project with template code.
- Edit the GitHub classroom repository with your submission
  - Create a new submission in Autolab when ready.
- **Requirements**
  - Make sure you are using Scala 2.13.x
  - Don’t add any outside packages