HOW WAS YOUR WEEKEND?

1. Read and Post on Piazza
2. Installed JDK & Spark
3. Submit Your Assignment 1 (Due Today 11:59pm, Blackboard)
4. Office hours: Tue 1-2, 6-7:30, Wed 6-7:30PM

"This weekend was too long" said nobody, ever.

Image source: http://www.liverunsparkle.com/its-a-long-weekend-up-in-here/
Dear Professors:

Last Fall, we launched a new service called Piazza, which connects students with other students and with potential employers. Many of the 31 enrolled students have already benefited from this service, and we believe that it will be even more valuable for you this semester. Piazza allows students to ask questions and receive answers from their peers, professors, and teaching assistants. This can help to facilitate learning and make it easier for students to get the support they need.

We hope that you will consider using Piazza in your classes and that you will find it to be a valuable tool for promoting student engagement and success. If you have any questions or concerns, please do not hesitate to contact us. We would be happy to provide you with more information about Piazza and how to use it.

Sincerely,

[Your Name]
SCALA CRASH COURSE

• “Stairs” in Italian

• Why Scala?

• Spark is written in Scala originally

• Quite fun
WHAT DO YOU KNOW ABOUT SCALA?
THINGS ABOUT SCALA

• Object-Oriented
  • classes can be extended
  • every value is an object

• Functional
  • every function is a value
  • so, every function is an object

• Statically typed
  • type inference saves us efforts to write explicit types

• Interoperates with Java
  • can use any Java class and can be called by Java
WHAT DO YOU WANT TO LEARN ABOUT SCALA?
SCALA FOR TODAY

• Syntax
  • define variables
  • define functions
  • closures
  • collection
  • control structures
• Compile using sbt
• A show-and-tell

A comprehensive step-by-step guide

Programming in Scala
Second Edition

Updated for Scala 2.8

artima

Martin Odersky
Lex Spoon
Bill Venners
LET’S WORK IN SCALA SHELL
VARIABLES

• var x: Int = 5

• var x = 5 // type inferred

• val myState = “free fall” // read-only, final, value cannot be changed
DATA TYPES

- **Byte**  8 bit signed value. Range from -128 to 127
- **Short** 16 bit signed value. Range -32768 to 32767
- **Int** 32 bit signed value. Range -2147483648 to 2147483647
- **Long** 64 bit signed value. -9223372036854775808 to 9223372036854775807
- **Float** 32 bit IEEE 754 single-precision float
- **Double** 64 bit IEEE 754 double-precision float
- **Char** 16 bit unsigned Unicode character. Range from U+0000 to U+FFFF
- **String** A sequence of Chars
- **Boolean** Either the literal true or the literal false

All the data types listed above are objects. There are no primitive types like in Java. This means that you can call methods on an Int, Long, etc.
FUNCTIONS

```
"def" starts a function definition
function name
parameter list in parentheses
function's result type
equals sign

def max(x: Int, y: Int): Int = {
    if (x > y)
        x
    else
        y
}
```

function body in curly braces
FUNCTIONS

first letter in function name needs to be lower case

- def square(x: Int): Int = x*x
- def square(x: Int): Int = { x*x }
- def announce(text: String) = { println(text) }
- def addTwo(x: Int): Int = x + 2
CLOSURES

• a function, whose return value depends on the value of one or more variables declared outside this function

  • var factor = 3

  • def multiplier = (i:Int) => i * factor // factor is the variable outside this function

    we could also say

    var multiplier = (i:Int) => i * factor

• What will be the output for

  • multiplier(1)

  • multiplier(2)
CLOSURES

- multiplier(1) // 3
- multiplier(2) // 6
CONTROL STRUCTURES

var x = 30;

if (x<20) {
    println ("free fall");
} else{
    println ("parachute");
}

Semicolon is optional
var x = 30;

var myState = "free fall";

while (x>0) {
    if (x< 15) { myState = "parachute" };
    println (myState);
    x = x - 1;
}


CONTROL STRUCTURES

• As such there is no built-in `break` nor `continue` statements available in Scala

• well, for the later versions of Scala 2.8, there are objects defined for the purpose.
COLLECTIONS IN SCALA

- Scala collections have mutable and immutable collections.
- A mutable collection can be updated or extended in place.
  - This means you can change, add, or remove elements of a collection.
- Immutable collections, by contrast, never change.
COMMON COLLECTIONS

- Mutable
  - Map, HashMap, ListMap, MutableList, LinkedList, Seq

- Immutable
  - List, Array, Vector, Set, String, Seq
val list = List(1, 2, 3)

list.foreach(x => println(x)) // prints 1, 2, 3

list.foreach(println) // same

list.map(x => x + 2) // returns a new List(3, 4, 5)

list.map(_ + 2) // same

list.filter(x => x % 2 == 1)// returns a new List(1, 3)

list.filter(_ % 2 == 1) // same
KEEP CALM AND EXERCISE
WHAT DO YOU GET?

```scala
import scala.collection.mutable
val map = mutable.Map.empty[String, Int]
map("hello") = 1
map("there") = 2
map
map.foreach(println)
map("hello")
map.filter(map("hello") == 1)
map.filter(_ == Pair("hello", 1))
map.filter(_ == Pair("there", 2))
map.filter(_ == Pair("there", 1))
```
> import scala.collection.mutable
> val map = mutable.Map.empty[String, Int]
> map("hello") = 1
> map("there") = 2
> map
> map.foreach(println)

> map("hello")
// res25: Int = 1

> map.filter(map("hello")==1)
// <console>:14: error: type mismatch;
// found   : Boolean
// required: ((String, Int)) => Boolean
// map.filter(map("hello")==1)

> map.filter(_==Pair("hello",1))

> map.filter(_==Pair("there",2))

> map.filter(_==Pair("there",1))
PROCESSING COLLECTIONS

• map(f: T => U): Seq[U]  // Each element is result of f
• flatMap(f: T => Seq[U]): Seq[U]  // One to many map
• filter(f: T => Boolean): Seq[T]  // Keep elements passing f
• exists(f: T => Boolean): Boolean  // True if one element passes f
• forall(f: T => Boolean): Boolean  // True if all elements pass
LET’S WORK IN SCRIPTS
- USING SBT TO COMPILE
STEP 1: SETUP SBT

From the directory that you copy from the spark thumb drive

• Go to spark_disk/sbt

• NOT spark_disk/spark/sbt

• chmod a+x sbt

• mkdir -p src/main/scala
STEP 2: WRITE YOUR HELLOWORLD.SCALA

• Create a file called HelloWorld.scala using your text editor

  • in Mac, you could use emacs, vim or nano; You might want to open another Terminal window to work on the editor while keep the ./sbt directory active in one Terminal

  • in Windows, you could use NotePad or WordPad as the text editor

• Put the following line in your file

  object HelloWorld {
    def main(args: Array[String]) = println("Hi, cosc 282!")
  }

• mv HelloWorld.scala src/main/scala/.

• Note: Make sure there is only one .scala file in src/main/scala/. We will talk about how to build a package later. As for now, just compile one file
STEP 3: COMPILE AND RUN

- go back to the sbt directory
  - `cd ./spark_disk/sbt`
- type `./sbt`
- from the sbt prompt, type "run"
  > `run`
- keep typing "run", the program will be compiled and run again
  > `run`
YOU SHOULD GET SOMETHING LIKE THIS

cs-ad-d9fy11:sbt gh243$ ./sbt
[info] Set current project to sbt (in build file:/Users/gh243/Desktop/Teaching/cosc282/spark_disk/sbt/)
> run
[info] Running HelloWorld
Hi, cosc 282!
[success] Total time: 0 s, completed Sep 9, 2015 1:55:01 PM
> run
[info] Running HelloWorld
Hi, cosc 282!
[success] Total time: 0 s, completed Sep 9, 2015 1:55:03 PM
> run
[info] Running HelloWorld
Hi, cosc 282!
[success] Total time: 0 s, completed Sep 9, 2015 1:55:05 PM
>
FUN TIME
SHOW-AND-TELL

val states = List("blue sky", "crazy", "jump", "free fall", "parachute", "alive", "dead", "cloudy")
var myState = ""
println("I have a friend ")
println("Sometimes she is " + states(1))
print ("When it is ")

val r = scala.util.Random
var chance = r.nextInt(100)

if (chance >= 50) {
    myState = states(7)
    println(myState)
    println("She is " + states(5))
}
else {
    myState = states(0)
    println(myState)
    println("She " + states(2))
}
I have a friend
Sometimes she is crazy
When it is cloudy
She is alive

[success] Total time: 2 s, completed Sep 9, 2015 2:08:24 PM
> run
[info] Running HelloWorld
I have a friend
Sometimes she is crazy
When it is cloudy
She is alive

[success] Total time: 0 s, completed Sep 9, 2015 2:08:29 PM
> run
[info] Running HelloWorld
I have a friend
Sometimes she is crazy
When it is cloudy
She is alive

[success] Total time: 0 s, completed Sep 9, 2015 2:08:30 PM
> run
[info] Running HelloWorld
I have a friend
Sometimes she is crazy
When it is blue sky
She jump

[success] Total time: 0 s, completed Sep 9, 2015 2:08:31 PM
> run
[info] Running HelloWorld
I have a friend
Sometimes she is crazy
When it is blue sky
She jump

[success] Total time: 0 s, completed Sep 9, 2015 2:08:33 PM
ASSIGNMENT 2 - FINISH THE STORY

• Using control structures

• (Bonus) Using processes for collections

• What to submit:

  • your codes

  • screencapture of at least 4 random runs of results

• Due: Next Wed 9/16, 11:59pm
HERE COMES THE REAL SHOW-AND-TELL
a few videos are put on piazza. they are the demos and assignment related procedures that we have shown in class. Please check them out