Spring 2013

CS 3190: Programming Language Workshop in Scala

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

CS 3190 Programming Language Workshop in Scala (1 Credit)

- **Instructor** : T. K. Prasad
- **Phone No.** : (937)-775-5109
- **Email** : t.k.prasad@wright.edu
- **Home Page**: http://knoesis.wright.edu/tkprasad

- **Quarter** : Spring 2013
  - **Office Hrs** : MW, 3:30-4:30pm , 395 Joshi (or by appointment)
  - **One and Only Class** : **January 9, Wednesday, 3:30-4pm, 399 Joshi**

Course Description

This course is designed as a self-study in Scala. You are expected to learn the language and solve a set of programming problems assigned to you using Scala available from http://www.scala-lang.org/. There are no exams. We officially meet only once in the semester (and attending this meeting is optional if the online document is clear). However, I will be available in the posted office hours for clarifications and discussions about the programming problems. Typically, most of the comunication is via emails.

Prerequisite

- Experience with programming in Java.

Reference Texts

- Martin Odersky, Lex Spoon, and Bill Venners: *Programming in Scala, 2nd Ed.* Artima, Inc. 2011, **ISBN-13:** 978-0981531649.
- Cay Horstmann : *Scala for the Impatient*, Addison-Wesley Professional, 2012, **ISBN-13:** 978-0-321-77409-5.

Grading

Each programming assignment will be graded as *Pass/Unsatisfactory*, and the letter grade 'P' or 'U' will be assigned at the end of the course.

Course Policies

1. All work must be turned-in by **April 17, 2013.**
2. Do not expect an incomplete for any reason. Each assignment has separate deadlines.
3. You must pass all the assignments to pass the course. The code you turnin must be your own creation. Copying code from available books, and/or cutting and pasting code from the Internet is strictly prohibited.
4. Each program should be well-documented and adequately tested.
5. You must turnin well-documented source code runnable using Scala. You should include (i) instructions for running and testing the program, and (ii) sample test inputs and outputs to indicate that you have tested your code adequately, in the source file as comments or as a separate ReadMe.txt . If you have multiple files, submit them as a single zip-archive. To turnin the i^{th} assignment (where i = 1,2,...), create `asgi.scala` or the archive `asgi.zip`, and upload it to the corresponding assignment DropBox on Pilot before the due date.
6. You may also be required to demonstrate your code in my office hours after the due date.

Assignments

(Assignment 1 - due 1/24) Gray Code Generation

An n-bit Gray code is a sequence of n-bit strings constructed according to certain rules. For example,
n = 1: C(1) = ("0", "1").
n = 2: C(2) = ("00", "01", "11", "10").
n = 3: C(3) = ("000", "001", "011", "010", "110", "111", "101", "100").
Find out the construction rules and write a function to generate Gray codes.

```
scala> gray(3)
res0 : List[String] = List(000, 001, 011, 010, 110, 111, 101, 100)
```

(Assignment 2 - due 2/13) N-Queens problem

Eight Queens problem is a classical problem in computer science. The objective is to place eight queens on a chessboard so that no two queens are attacking each other; i.e., no two queens are in the same row, the same column, or on the same diagonal. Write a program for solving the N-Queens problem.

Hint: Represent the positions of the queens as a list of numbers 1..*N*. For example: for n=8, `List(4, 2, 7, 3, 6, 8, 5, 1)` means that the queen in the first column is in row 4, the queen in the second column is in row 2, etc. Use the generate-and-test paradigm judiciously. Avoid copying the solution given in the text.

(Assignment 3 - due 3/20) Verbalizing numbers in English.

On financial documents, like cheques, numbers must sometimes be written in full words. For example: 175 must be written as 'one hundred and seventy five'. Write a function verbalize(num: Int) to print (non-negative) integer numbers in words.

```
scala> verbalize(000)
res3: String = zero
scala> verbalize(1234567890)
res4: String = one billion two hundred and thirty four million five hundred and sixty seven thousand eight hundred and ninety
scala> verbalize(1000020345)
res5: String = one billion twenty thousand three hundred and forty five
```

(Assignment 4 - due 4/17) Open Ended Problem.

Take a non-trivial Java application (say 150-200 lines of code in OOP style) that interests you and rewrite it in Scala. Turnin both the versions. Document and critique the generic changes to be made to the Java program to obtain the corresponding Scala program and the resulting reduction in code size in `ReadMe.txt`.
