Birla Institute of Technology & Science, Pilani
Pilani Campus

Centre for Software Development, SDET Unit

Course Handout: Second Semester 2019-2020

Course Title:	Programming - Python, and Functional (p:paf)
Instructor-in-charge:	Ishan Bhanuka (email: f2016075@)
Instructor:	Satvik Golechha (email: f20170117@)
SDET Website:	http://discovery.bits-pilani.ac.in/CSDCourse/
Course Website:	p-paf.github.io
Lectures:	Tue, Sat (6 to 7:30 pm)

Course Description: Python is one of the most popular programming languages of the 21st century. It used for developing desktop applications, websites, and web applications. It is also **the** language to learn for becoming a data scientist.

Functional programming (FP) is a way of thinking about software construction that emphasizes functional purity and immutability. Due to the ease of testing, verification and debugging of functional code, the use of functional programming is continuously on the rise in the industry. According to Stack Overflow's Developer Survey 2019, 5 of the 7 highest paying programming languages enforce the functional approach.

The goal of this course is to teach programming in Python (3.8) and the functional programming (FP) design methodology (through Haskell), with a focus on hands-on learning and industry relevance.

Learning Outcomes: Throughout this course, you will:

- Learn to program in Haskell, a purely functional language
- Think functionally, whichever language you later choose
- Code proficiently in Python (and it's functional features)
- Build a project (with an AI Game Player as a playground assignment)
- Have an amazing classroom experience

Textbooks and References: There will be no need to purchase any books/software. All resources would be open and provided by us. Also, throughout the course, we'll be putting up loads of articles/code for you to try out. You can access them on the course website. Some excerpts (which we will provide) will be taken from:

- *Learn Python 3 The Hard Way by Zed Shaw*
- *Thinking Functionally with Haskell by Richard S. Bird*

Course Structure:

Lecture	Title	Content
1	Recursion	*in which we introduce the course, and see the power of recursion through towers, trominoes, and a fairy*
2	Python Dance	*in which we learn python syntax (3.8) through a myriad of examples and TV Show references*
3	The FP Shake	*where we understand inherently recursive data structures through lists and trees in Haskell*
4	You Haskell	in which we learn Haskell's syntax, reading documentation pages and figuring things out through compile errors
5	Python Fangs	*where we face the challenge of recursion through DFS, Minimax, and other recursive implementations*
6-7	Curry Recipe	*where we understand the most important concepts of higher-order functions, currying, and lazy evaluation*
8	Python Scales	*in which we delve deeper into the more advanced and functional aspects of Python - decorators, generators, lambdas, etc*
9	Bread Butter, and Haskell	*where we understand and work with map, filter, and list comprehension - some fundamental definitions Haskell uses*

10-11	Fold and Behold	*in which we look at left and right folds, reduce and function composition and why people love them*
12	Lambda Calculus	*where we build the smallest computer in the universe and use it to build Haskell and a startup incubator*
13	Inherit a Haskell	*where we understand data, type classes and polymorphism in Haskell, and why it's a sin to kill a mockingbird*
14-15	Throw a Haskell	*in which we become pro, by understanding how Functors and Applicatives work*
16	Peanut Butter and Jelly	*a surprise fun-day*
17-18	Save a Haskell	*where we meet impure functions, contexts, and monads because nothing is perfect*
19	Invent with Python	*where we build a game GUI and learn writing better Python code and PEP because it's useful*
20	Think you know Haskell?	*where we understand streams, laziness and generating random structures*
21	Think you know Python?	*in which we understand Floyd-Hoare logic, invariance, and how to prove our code's correctness*
22	This Feeling . . .	*where we depart with conclusions, future directions, and some advice for the journey ahead*

Evaluation Details: There will be no pen-and-paper evaluative for this course, and the focus will be on implementation, learning, and involvement. There will be three Assignments (10% each) and two Nalanda online quizzes (10% each), and a final Term Project (worth 40%). The last 10% constitutes 'Karma Points', which include class participation, involvement, and anything you do to improve the effectiveness of learning for yourself and everyone else.

For more details and FAQs, please visit the course website.

Ishan Bhanuka, Satvik Golechha