

# CS 3100: Paradigms of Programming

Jul.-Nov Semester 2019

'C' Slot; CS 34

**Slots are: Mon (10 – 10.50am); Tue (9 – 9.50am); Wed (8 – 8.50am); Fri (1 – 1.50pm);**

Dr. KC Sivaramakrishnan, BSB 371; Phone: 4359

Email: [kcsr@iitm.ac.in](mailto:kcsr@iitm.ac.in)

TA(s): Shashank Shekhar Dubey ([shashank.shekhardubey@gmail.com](mailto:shashank.shekhardubey@gmail.com)),

Diptanshu Kakwani ([dipkawkani@gmail.com](mailto:dipkawkani@gmail.com)),

Sumit Padhiyar ([sumitpad@cse.iitm.ac.in](mailto:sumitpad@cse.iitm.ac.in)),

Atul Dhiman ([cs18m014@smail.iitm.ac.in](mailto:cs18m014@smail.iitm.ac.in)),

M Prasanna Kumar ([CS15B050@smail.iitm.ac.in](mailto:CS15B050@smail.iitm.ac.in)),

Bikash Behera ([CS19M019@smail.iitm.ac.in](mailto:CS19M019@smail.iitm.ac.in))

Updated on July 29, 2019

Note: Course related communications will be on IITM Moodle site (CS3100); please regularly check the email that is linked to your email account.

## 1 Course objectives

The aim of the course is to teach you about the different paradigms of programming, their underlying concepts, and the relationships between them. We will mostly look at functional, logic and concurrent programming, touching upon imperative programming concepts as we go through the course. We will study functional and concurrent paradigm through OCaml and logic paradigm through Prolog (specifically SWI-Prolog).

## 2 Learning Outcomes

- To learn about functional programming paradigm through the study of OCaml programming language.
- To learn about logic programming paradigm through the study of Prolog programming language.
- To learn about concurrent programming paradigm through concurrency extensions of OCaml.
- Identify what language features are fundamental and what features are syntactic sugar.
- To be able to write clear, correct, robust, reusable code.
- To be able to identify the best paradigm for the given problem.

## 3 Course prerequisite(s)

CS2200, CS2600, CS2700, CS2710 – or equivalent for each course.

## 4 Classroom Mode

Traditional Lectures.

## 5 Textbooks

**cs3110** “Functional Programming in OCaml”, Spring 2019 Edition, CS 3110 Cornell University, <http://www.cs.cornell.edu/courses/cs3110/2019sp/textbook/>.

**PLCC** Ravi Sethi, “Programming Languages : Concepts and Constructs”, 2nd edition, Addison-Wesley.

## 6 Reference Books

Material from the following books will be used as necessary.

**RWO** Anil Madhavapeddy, Yaron Minsky, Jason Hickey, “Real World OCaml”, O’Reilly Publishers, <https://dev.realworldocaml.org/>.

**TAPL** Benjamin C. Pierce, “Types and Programming Languages”, The MIT Press.

## 7 Course Requirements

You are *required* to attend all the lectures. If you miss any of them it is your responsibility to find out what went on during the classes and to collect any materials that may be handed out.

Class participation is strongly encouraged to demonstrate an appropriate level of understanding of the material being discussed in the class. Regular feedback from the class regarding the lectures will be very much appreciated.

## 8 Planned Syllabus

The following topics will be covered, but not necessarily in the order listed below:

1. Introduction to Programming Languages.
2. Functions, Pattern Matching, Algebraic Data Types
3. Lists, Trees, Sets, Hash Tables, Graphs
4. Lambda Calculus:  $\alpha$  equivalence,  $\beta$  reduction,  $\eta$  expansion.
5. Lambda Calculus: Church encoding, Y-combinator
6. Simply Typed Lambda Calculus, Progress and Preservation
7. Modular Programming in OCaml
8. Monads and Applicative Functors, Generalised Abstract Data Types, Row Polymorphism
9. Mutability, Streams and Laziness, Promises, Memoization
10. Logic Programming, Horn Clauses, Semantics
11. Prolog examples, Arithmetic, Backtracking
12. Generate and Test, Symbolic Execution
13. Lists, Difference Lists, Graph Search
14. Negation as Failure, Cut
15. Databases, Constraint Logic Programming, Databases
16. Continuations, Delimited Continuations, Effect Handlers
17. MVars, Locks and Condition Variables, Selective Communication
18. Asynchronous I/O, Stream Programming, Transactional Memory

## 9 Tentative Grading Policy

The following allocation of points is tentative. These may change during the semester.

Quiz 1 (Date: Sep. 4, 2019):	15%
Quiz 2 (Date: Oct. 16, 2019):	15%
6 Assignments (written / OCaml / Prolog):	30%
Final Exam (Date: Nov 19, 2019):	40%

## 10 Academic Honesty

Academic honesty is expected from each student participating in the course. NO sharing (willing, unwilling, knowing, unknowing) of assignment code between students, submission of downloaded code (from the Internet, Campus LAN, or anywhere else) is allowed.

Academic violations will be handled by IITM Senate Discipline and Welfare (DISCO) Committee. Typically, the first violation instance will result in ZERO marks for the corresponding component of the Course Grade and a drop of one- penalty in overall course grade. The second instance of code copying will result in a 'U' Course Grade and/or other penalties. The DISCO Committee can also impose additional penalties.

Please protect your Moodle account password. Do not share it with ANYONE. Do not share your academic disk drive space on the Campus LAN.