COMPILER DESIGN

PROF. SANTANU CHATTOPADHYAY
Dept. of Electronics and Electrical Communication Engineering
IIT Kharagpur

TYPE OF COURSE : New | Core | UG
INTENDED AUDIENCE : CSE, IT, B.Sc (CS),
MCA, MS (CS)

COURSE OUTLINE :
Compilers have become part and parcel of today’s computer systems. They are responsible for making the user’s computing requirements, specified as a piece of program, understandable to the underlying machine. There tools work as interface between the entities of two different domains – the human being and the machine. The actual process involved in this transformation is quite complex. Automata Theory provides the base of the course on which several automated tools can be designed to be used at various phases of a compiler. Advances in computer architecture, memory management and operating systems provide the compiler designer large number of options to try out for efficient code generation. This course on compiler design is to address all these issues, starting from the theoretical foundations to the architectural issues to automated tools. Being primarily targeted to a one - semester course for the undergraduate students, the course will follow the current GATE syllabus, enabling the students to prepare well for the same. It can also help all other participants looking for an introduction to the domain of compiler designs and code translators.

ABOUT INSTRUCTOR :
Santanu Chattopadhyay received his BE (CS), Calcutta University in 1990. He received M.Tech in Computer and Information Technology and PhD where from Indian Institute of Technology Kharagpur in 1992 and 1996, respectively. He is currently a Professor Prior to this, he had been a faculty member in the IIEST Sibpur and IIT Guwahati where he has taught the subject of Compiler Design several times. His research interests include Digital Design, Embedded Systems, System-on-Chip (SoC) and Network-on-Chip (NoC) Design and Test, Power- and Thermal-aware Testing of VLSI Circuits and Systems. He has published more than 150 papers in reputed international journals and conferences.

COURSE PLAN :
Week 01 : Introduction
Week 02 : Lexical Analysis
Week 03 - 5 : Parsing – Part I, II, III
Week 06 : Syntax Directed Translation
Week 07 : Type Checking and Symbol Tables
Week 08 & 09: Runtime Environment Management – Part I,II
Week 10 - 12 : Intermediate Code Generation – Part I,II,III

INDUSTRIES APPLICABLE TO : All software industries

COURSE DURATION : 12 weeks (28 Jan’19 - 19 Apr’19)
EXAM DATE : 27 April 2019