Semester:	1				
Course Code:	CO1010				
Course Name:	Programming for Engineers I				
Credit Value:	3 (Notional hours 150)				
Prerequisites:	None				
Core/Optional	Core				
Hourly Breakdown	Lecture hrs.	Tutorial hrs.	Practical class hrs	Design hrs	Independent Learning & Assessment hrs.
	15	10	30	10	85

Course Aim: To develop logical thinking through algorithms and structured programming constructs so that the students will be able to build software applications to analyze and solve engineering problems.

Intended Learning Outcomes:

On successful completion of the course, the students should be able to;

- **Construct** algorithms to solve engineering problems
- **Use** structured programming constructs and build software applications
- **Apply** good programming practices

Course Content:*(Only main topics & subtopics)*

- **Basics :** Variables. Operators and precedence. Data types. Number systems and numerical precision.
- **Control Structures:** Conditions and loops.
- **Modularization :** Standard libraries and functions. User-defined functions.
- **Input/Output:** Standard input/output. File input and file output
- **Data Structures:** List and list comprehension. String processing and formatting. Stack and Queue. Dictionaries.
- **Object-Oriented Concepts:** Classes and Objects. Accessing variables and functions within objects.
- **Quality Assurance:** Good programming practices. Testing. Debugging. Exception and error handling.
- **Algorithms :** Developing algorithms and writing programs for the solutions of well-defined problems related to Engineering.
- **Numerical Computations:** Introduce concepts of numerical packages/libraries such as numpy and the use of mathematical software such as Matlab to solve problems such as those listed under item 8

Teaching /Learning Methods:

Flipped classrooms, small group discussion classes, project-based learning.

Assessment Strategy:

Continuous Assessment 60%	Final Assessment 40%		
Details: Online class participation 5% Practicals 35% Assignments and Projects 20%	Theory (%) 40%	Practical (%) -	Other (%) -

Recommended Reading:

- John DeNero (2017), *Composing Programs, a free online introduction to programming and computer science*, 10 Oct 2019, http://composingprograms.com
- Ron Reiter (2018), *Interactive Python tutorial*, 10 Oct 2019,https://www.learnpython.org/