6TH SEMESTER BSc. ITM(H)			
	SUB CODE		SUB NAME
MAJOR	CORE-I	PAPER-14	Theory of Computation
	CORE-I	PAPER-15	Python Programming
MINOR	CORE-III	PAPER-3	Electricity & Magnetism
SEC	PAPER-3		
VAC	PAPER-4		

Core XIV **Theory of Computation**

Course Objectives:

This course focuses on the basic theory of Computer Science and formal methods of computationlike automata theory, various machines, grammars and Turing Machines. To explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

Course Outcomes: On completion of this course, the students will be able to

- Understand the basic properties of formal languages and grammars.
- Differentiate among regular, context-free and recursively enumerable languages.
- Make grammars to produce strings from a specific language.
- Minimize the finite automata.
- Acquire concepts relating to the theory of computation and computational models including decidability and intractability.
- Design and deal with Turing machines.
- Get the basic foundation of compiler design.

Unit-I:

Alphabet, Languages, Grammars, Finite Automata (DFA, NFA), Regular operations, Regular Languages/Grammars, Regular Expressions, Finite Automaton With ε -Moves, Equivalence of NFA and DFA.

Outcome: The students will be able to understand the basic properties of formal languages and grammars, DFA & NFA.

Unit-II:

Minimization of Finite Automata, Closure Properties of Regular operations, Pumping Lemma of Regular Languages, Context Free Grammars, Context Free Languages, Derivation Tress, Ambiguity, Properties of Context Free Languages, Left and Right Linear Grammars.

Outcome: The students can minimize the finite automata, understand Pumping lemma and

Right linear and Left Linear grammar.

Unit-III:
Chomsky Normal Form (Elimination of Useless Symbols, Unit Productions, Null Productions), Pushdown Automata, Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Languages.
Outcome: The students can be able to Design Push down automata, convert a grammar to CNF'

Unit-IV:
Turing Machines, Turing Computability, Type 0 Languages, Techniques for Turing Machine Construction, Multihead And Multitape Turing Machines, Church Turing thesis, Recursive and Recursively Enumerable Set, Chomsky Hierarchy of Languages.
Outcome: The students will be able to Design and deal with Turing machines. Get the basic foundation of compiler design, Differentiate regular, context-free and recursively enumerable languages.

Text Books:

- ✓ *Introduction to the theory of Computation, Michael Sipser, Cengage Learning.*
- ✓ *Introduction to Automata Theory, Languages and Computation, J. E. Hopcrpft and J.D. Ullman, Pearson Education, 3rdEdition.*

Reference Books:

- ✓ *JFLAP - An Interactive Formal Languages and Automata Package Rodger, Finley,ISBN:0763738344*
- ✓ *JFLAP User Manual and Exercises, Tobias Fransson. Available in the Web.*

Core XIV- Practical/Tutorial: Theory of Computation Lab

Use Java Formal Language and Automata Language (FLAP) software Package (can be downloaded from www.jflap.org) to carry out the following experiments:

- Regular Language-Create: DFA, NFA, Regular Grammar, and Regular Expression.
- Regular Language – conversions: NFA to DFA to Minimal DFA, NFA to regular expression& vice versa.
- NFA to regular grammar & vice-versa.
- Context-free language–create: push-down automaton, context-free grammar.
- Context-free language – transform: PDA to CFG, CFG to PDA (LL parser), CFG to PDA (SLR Parser), CFG to CNF, CFG to LL parse table and parser, CFG to SLR parse table and parser.
- Recursively Enumerable language: Turing machine (1 tape), Turing machine (multi tape), Turing machine (building blocks), unrestricted grammar.

Core XV **Python Programming**

Course Objectives:
To acquire programming skills in core Python. To acquire Object Oriented Skills in Python. Todevelop the ability to write database applications in Python.

Course Outcome:
On completion of this course, the students will be able to

- Explain basic principles of Python programming language.
- Implement object-oriented concepts.
- Implement database and GUI applications

Unit-I:

Python: Features of Python , Installing Python for windows and setting up paths, writing and Executing of a python programs, Python Virtual machine, Frozen binaries, Comparison betweenC, Java and python , Comments , Doc strings ,How python sees variables, Data typesinPython,builtintypes,sequencesinpython,sets,literalsinPython,userdefineddatatypes, identifiers & reserved words, Naming convention in python.
Outcome: Students will be able to understand the syntax and basic concepts of python programming language.

Unit-II:

Various Operators in Python, Input & Output, Control statements, if statements, while loop, for loop, infinite loop, nested loop, else suit, break, continue, pass, assert, return statements, command line arguments.
Arrays in python, advantages using arrays, creating arrays, importing the array module, indexingand slicing on arrays, Processing the arrays, Comparing arrays.
Strings in Python, creating strings, Length of a string, indexing in strings, slicing strings, Concatenation and Comparing strings, Finding Sub Strings, Replacing a String.
Outcome: Students will be able to build basic programs using fundamental programming constructs

Unit-III:

Functions in Python, define a function, calling a function, return from function, pass by object Reference, Positional arguments, Default arguments, Recursive functions. Introduction to OOP, features of OOP, creating classes, the self-variable, constructor, types ofvariables, namespaces, types of methods.
Outcome: Students will be able to articulate the OOPs concepts as well as use of functions.

Unit-IV:

Inheritance: Define inheritance, types of inheritance, and constructors in inheritance, overridingsuperclass constructors& methods, the super () method, MRO Polymorphism: Duck typing philosophy of Python, operator overloading, method over riding,interfaces in python.
Exceptions: Errors in a python program, Exceptions, Exception handling, Types of Exceptions, the Exception block, the assert statement, user defined exceptions.

Python Database Connectivity: DBMS, types of databases used with Python, installation of MySQL database, setting path, verifying MySQL, installing MySQL connector, working with MySQL database, Using MySQL from python, retrieving rows, deleting rows, updating rows in a table.
Outcome: Students will be able to articulate the OOPs concepts such as inheritance and able to know how to handle exception and python database connectivity.

Text Books:

- ✓ *T.Budd, Exploring Python, TMH, 1stEd, 2011.*
- ✓ *Core Python Programming, Dr.R. Nageswar Rao, Dreamtech Press.*
- ✓ *Python Programming for Absolute Beginners, Michael Dawson, CENGAGE Learning.*

Reference Books:

- ✓ *Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: Learning with Python, Freely available online.2012.*

Online References:

- ✓ *Python Tutorial/Documentation www.python.or 2015*
- ✓ *http://docs.python.org/3/tutorial/index.html*
- ✓ *http://interactivepython.org/courselib/static/pythonds*
- ✓ *http://www.ibiblio.org/g2swap/byteofpython/read/*

Core XV- Software Lab based on Python Programming

- Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
- Write a Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria: Grade A: Percentage>=80, Grade B:

Percentage>=70 and <80Grade C: Percentage>=60 and <70Grade D: Percentage>=40and <60GradeE: Percentage<40.

- Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
- Write a Program to display the first n terms of Fibonacci series.
- Write a Program to find factorial of the given number.
- Write a Program to find sum of the following series for n terms: 1 –2/2! +3/3! -----
- Write a Program to calculate the sum and product of two compatible matrices.
- Install MySQL and connector and write Python programs to retrieve, inserting, delete,update rows in a table.