1.	**Name of Course**						**Theory of Computation**
2.	**Course Code**						**CCPS3603**
3.	**Name(s) of academic staff**						
4.	**Rationale for the inclusion of the course/module in the programme**						**Major** The theory of computation is a core subject in computer science that deals with whether and how efficiently problems can be solved on a model of computation, using an algorithm. It is a "thinking course", The core material of this course constitutes the fundamental concepts of computer science.
5.	**Semester and Year offered**						1/3
6.	**Total Student Learning Time (SLT)**	**Face to Face**					**Total Guided and Independent Learning**
	L = Lecture T = Tutorial P = Practical O= Others	**L** **28**	**T** **14**	**P**	**O**		Independent =84 Total= 126
7.	**Credit Value**						3
8.	**Prerequisite (if any)**						none
9.	**Objectives**: • To determine what can and cannot be computed, how quickly, with how much memory, and on which type of computational model. • To Explain the limitations of computing, the relative power of formal languages and the inherent complexity of many computational problems of practical importance;						
10.	**Learning outcomes:** By the end of the subject, students should be able to: • Define automata and computability, • Demonstrate a good understanding of automata and properties of formal languages, • Explain Chomsky hierarchy and computing power of each grammar/language type. • Compare algorithms with respect to time and space requirements.						
11.	**Transferable Skills:** • ability to undertake problem identification, formulation and solution • Increases problem-solving skills.						

12.	**Teaching-learning and assessment strategy** A variety of teaching and learning strategies are used throughout the course, including: • Classroom lessons. Lectures and Power Point presentations • Tutorial sessions: • brainstorming; • student-Lecturer discussion • collaborative and co-operative learning; • Independent study. Assessment strategies include the following: • Ongoing quizzes • Midterm tests • Performance Assessment (Assigned exercises, Presentation and final project) • Lecturer Observation
13.	**Synopsis:** The major area of studies include introduction of finite automata, and pushdown automata. It also includes an overview of context-free grammars and languages, Turing machines, regular and enumerable sets. The course will expose the students to increases the programming skills.
14.	**Mode of Delivery:** Lecture and Tutorial
15.	**Assessment Methods and Types:** **Coursework** **40%** • Quizzes 5% • Assignments 15% • Mid-Semester Exam 20% **Final Examination** **<u>60%</u>** **100%**

16. **Mapping of the course/module to the Programme Aims**

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
4	2	2	0	0	1	1	1	0	3	4	0

17. **Mapping of the course/module to the Programme Learning Outcomes**

LO1	LO2	LO3	LO4	LO5	LO6	LO7	LO8	LO9	LO10	LO11	LO12
4	3	1	0	1	0	4	2	2	1	1	0

18. **Content outline of the course/module and the SLT per topic**

	Details	**SLT**			
		L	T	Indep.	Total

	Topic 1	Introduction: • Complexity theory - Computability theory - Automata theory • Mathematical notions and terminology • Types of proof: Proof by construction - Proof by contradiction - Proof by induction	2	1	6	9
	Topic 2	Finite automata • Designing finite automata - The regular operations • Non-determinism: Equivalence of NFAs and DFAs - Closure under the regular operations • Regular expressions and their relationships to finite automata. • Non-regular languages The pumping lemma for regular languages	4	2	12	18
	Topic 3	Context-free grammars • Designing context-free grammars -Ambiguity - Chomsky normal form • Pushdown automata Formal definition of a pushdown automaton - Examples of pushdown automata -Equivalence with context-free grammars • Non-context-free languages The pumping lemma for context-free languages	4	2	12	18
	Topic 4	Computability theory • The Church-Turing Thesis • Turing machines • Variants of turing machines • Multitape Turing machines • Nondeterministic Turing machines • Hilbert's problems - Terminology for describing Turing machines • Decidable languages • The halting problem The diagonalization method - The halting problem is undecidable • A Turing-unacceptable language	6	3	18	27
	Topic 5	Reducibility • Undecidable problems from \\ language theory reductions via computation histories • A simple undecidable problem • Mapping reducibility Computable functions - Formal definition of mapping reducibility Advanced Topics in Computability Theory • The recursion theorem Self-reference - Terminology for the recursion theorem - Applications • Decidability of logical theories A decidable theory - An undecidable theory • Turing reducibility • Minimal length descriptions - Optimality of the definition - Incompressible strings and randomness	4	2	12	18
	Topic 6	Complexity theory • Measuring time complexity • Analyzing algorithms - Complexity relationships among models • The class P Polynomial time • The class NP • NP-COMPLETENESS Polynomial time reducibility • The Cook-Levin Theorem • NP-COMPLETE problems The vertex cover problem - The Hamiltonian path problem - The subset sum problem.	4	2	12	18

<table>
<tr><td rowspan="2"></td><td>Topic 7</td><td>Space Complexity
Savitch's theorem
The class pspace: The TQBF problem - Winning strategies for games - Generalized geography
The classes L and NL
NL-completeness Searching in graphs
Intractability
Hierarchy theorems
Exponential space completeness
Relativization Limits of the diagonalization method
Circuit complexity</td><td>4</td><td>2</td><td>12</td><td>18</td></tr>
<tr><td></td><td align="right">Total</td><td>28</td><td>14</td><td>84</td><td>126</td></tr>
<tr><td rowspan="2">19.</td><td colspan="6">Main references supporting the course:
1. Wayne Goddard, Introducing the Theory of Computation, Jones & Bartlett Publishers,2008
2. John Martin, Introduction to Languages and the Theory of Computation, McGraw-Hill Science,2010</td></tr>
<tr><td colspan="6">Additional references supporting the course:
1. Michael Sipser. Introduction to the Theory of Computation. Thomson, second edition, 2006.
2. Hopcroft, John E., and Jeffrey D. Ullman . Introduction to Automata Theory, Languages, and Computation. Addison-Wesley. 2006</td></tr>
<tr><td>20.</td><td colspan="6">Other additional information
All materials will be available to the students online.</td></tr>
</table>