Computer Science And Engineering
(Jan – May 2018)

GENERAL GUIDELINES

Do's:-

- Students should be on time for every lecture.
- Students are advised to show due respect to all faculty members.
- Students should keep the Classrooms, Laboratories and Workshops clean and tidy.
- Students must maintain absolute discipline and decorum, while on campus.
- **Students should come prepared with algorithm / flowchart / program / procedure for all the experiments before attending the laboratory session.**
- Students should bring the data sheets and laboratory records completed in all respects to the laboratory.
- Students are advised to clarify their doubts in the respective courses with the faculty.
- Students have to inform their parents that they should follow up the progress of their wards by being in touch with the institution authorities at regular intervals.
- **Students are advised to be present for the mentor meetings conducted by their respective Faculty Advisors, failing which appropriate disciplinary action will be taken.**

Don'ts:-

- Students are not permitted to attend the class without the identity card, once issued.
- **Ragging is strictly prohibited because it is punishable under Karnataka Education Act. Any student involved in ragging, will be severely punished – which includes handing over the case to Police, rustication from the college etc.**
- Writing on desks and walls is strictly prohibited, failing which the students will be fined heavily. If the identity of the individual is not established the entire class / students in the block will be fined.
- **Students must not use their cell phones during class hours. If any student is found using their cell phone during class hours it will be confiscated.**
- Students are not supposed to alter the configuration of the system / any software on the systems.

Computer Science And Engineering
(Jan – May 2018)

IV SEMESTER (2016-2020)

Sl. No.	Course Code	Course Title	Hours / week				Credits	Course Type
			L	T	P	S		
1.	UE16MA251	Linear Algebra & Its Applications	3	1	1	0	4	FC
2.	UE16CS251	Design and Analysis of Algorithms	4	0	0	0	4	CC
3.	UE16CS252	Data Base Management Systems	4	0	0	0	4	CC
4.	UE16CS253	Microprocessors & Computer Architecture	4	0	0	0	4	FC
5.	UE16CS254	Theory of Computation	4	0	0	0	4	CC
6.	UE16CS255	Design and Analysis of Algorithms Laboratory	0	0	2	0	1	CC
7.	UE16CS256	Data Base Management Systems Laboratory	0	0	2	0	1	CC
8.	UE16CS257	Microprocessor & Computer Architecture Laboratory	0	0	2	0	1	FC
9.	UE17MA151D	Engineering Mathematics-II (Applicable to Lateral Entry Students)	2	1	0	0	2	FC
		Total	**19/21**	**1/2**	**07**	**0**	**23/25**	

Computer Science And Engineering
(Jan – May 2018)

UE16MA251: Linear Algebra & its Applications(3-1-1-0-4)

# of Hours: 52

Class #	Chapter Title / Reference Literature	Topics to be Covered	% of Portion covered	
			% of Syllabus	Cumulative %
1	**Unit 1 Matrices and Guassian Elimination** T1:1.1-1.6	Introduction to Linear Algebra	**20**	**20**
2		The Geometry of Linear Equations		
3		The Geometry of Linear Equations		
4		Gaussian Elimination –Forward Elimination		
5		Gaussian Elimination-Back substitution		
6		Scilab Class Number1 – Gaussian Elimination		
7		Elimination Matrices		
8		Triangular Factors and Row Exchanges		
9		Inverses and Transposes		
10		Scilab Class Number 2- LU Decomposition		
11		Scilab Class Number 3-Inverses		
12	**Unit 2 Vector Spaces** T1: Chapter 2 2.1—2.4	Vector Spaces	**20**	**40**
13		Column Space		
14		Nullspace		
15		Linear Independence		
16		Linear Independence, Basis		
17		Basis and Dimensions		
18		The Four Fundamental Subspaces		
19		The Four Fundamental Subspaces		
20		Scilab Class Number 4– Span of Column Space of A		
21		Scilab Class Number 5 – Four Fundamental Subspaces of A		
22	**Unit 3 Linear Transformations and Orthogonality** T1: 3.1—3.3	Linear Transformations in two dimensions	**20**	**60**
23		Linear Transformations in two dimensions		
24		Linear Transformations in two dimensions		
25		Orthogonal Vectors and Subspaces		

26		Orthogonal Vectors and Subspaces		
27		Cosines and Projections onto Lines		
28		Projections onto a Plane		
29		Least Squares		
30		Scilab Class Number 6-Projections by Least Squares		
31	**Unit 4 Orthogonalization, Eigen Values and Eigen Vectors** T1: 3.4 T1: 5.1—5.3 T2:2.14-2.15	Orthogonal Bases	**20**	80
32		**Orthonormal Basis**		
33		The Gram- Schmidt Orthogonalization		
34		QR Factorization		
35		Scilab Class Number 7- The Gram- Schmidt process		
36		Introduction to Eigen values and Eigen vectors,		
37		Power Method		
38		Diagonalization of a Matrix		
39		Diagonalization of a Matrix		
40		Scilab Class Number 8- Eigen Values and Eigen Vectors		
41		Scilab Class Number 9-The Power Method		
42	**Unit 5 Linear Programming Problem** T1:8.1-8.2 T2: 28.5.1—5.3 T2:33.5—33.8	Graphical solution of LPP	**20**	100
43		Graphical solution of LPP		
44		General LPP		
45		Canonical and Standard Forms		
46		Canonical and Standard Forms		
47		Scilab Class Number 10- Graphical Solution of LPP		
48		Simplex Method		
49		Simplex Method		
50		Simplex Method		
51		Repetition of Lab Hour/ Practice Test in Scilab		
52		Scilab – In Semester Assessment		

Literature

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Linear Algebra and its Applications by Gilbert Strang, ,	4th	Thomson Brooks/ Cole	2007
Text Book	T2	Higher Engineering Mathematics by B S Grewal	42nd	Khanna Publishers	

Computer Science And Engineering
(Jan – May 2018)

UE16CS251:Design and Analysis of Algorithms(4-0-0-0-4)

# of Hours: 52

Class #	Chapter Title / Reference Literature	Topics to be Covered	% of Portion covered	
			% of Syllabus	Cumulative %
1	**Unit 1** T1: Chapters 1 (1.1, 1.2, 1.3), and 2 (2.1, 2.2, 2.3, 2.4)	Motivation for the course. Scheme of the course. Introduction to Algorithms.	**16%**	**16%**
2		Fundamentals of Algorithmic problem solving. Important problem types – sorting, searching.		
3		Important problem types – string processing, graph problems, Combinatorial , Geometrical, numerical problems		
4		Analysis Framework, Asymptotic notations		
5		Basic Efficiency classes. Analysis of non recursive algorithms		
6		Analysis of Non recursive algorithms		
7		Examples of Recursive algorithms		
8		Examples of Non-recursive algorithms		
9	**Unit 2** T1: Chapters 3 (3.1, 3.2, 3.4), and 4 (4.1, 4.2, 4.3, 4.4, 4.5)	Brute-Force approach	**23%**	**39%**
10		Sequential Search		
11		Selection Sort and Bubble Sort		
12		Brute-Force String Matching		
13		Exhaustive Search – Travelling Salesman Problem		
14		Knapsack Problem, Assignment Problem		
15		Divide-and-Conquer approach		
16		Mergesort		
17		Quicksort		
18		Binary Search		
19		Multiplication of Large Integers		
20		Strassen's Matrix Multiplication		
21	**Unit 3** T1: Chapters 5 (5.1, 5.2, 5.3, 5.4, 5.5), and 6 (6.1, 6.2, 6.3, 6.4, 6.5)	Decrease-and-Conquer approach - Insertion Sort	**21%**	**60%**
22		Depth First Search, Breadth First Search		
23		Topological Sorting		
24		Algorithms for Generating Combinatorial Objects		
25		Decrease-by-a-Constant-Factor Algorithms		
26		Transform-and-Conquer approach - Presorting		
27		Gaussian Elimination		
28		Balanced Search Trees - AVL		
29		2-3 trees		
30		Heaps and Heapsort		
31		Horner's Rule and Binary Exponentiation		
32	**Unit 4** T1: chapters 7 (7.1, 7.2, 7.3), and 8 (8.1, 8.2, 8.4)	Space and Time Tradeoffs - Sorting by Counting	**17%**	**77%**
33		Input Enhancement in String Matching – Horspool's algorithm		
34		Boyer-Moore Algorithm		
35		Hashing		
36		Dynamic Programming approach		
37		Computing a Binomial Coefficient		
38		Warshall's Algorithm		
39		Floyd's Algorithm		

40		Knapsack problem		
41	**Unit 5**	Prim's Algorithm	**23%**	**100%**
42		Kruskal's Algorithm		
43	T1 : chapters 9 (9.1, 9.2, 9.3, 9.4), 11 (11.1, 11.2, 11.3), and 12 (12.1, 12.2)	Problems on Prim's and Kruskal's Algorithm		
44		Dijkstra's Algorithm		
45		Problem using Dijkstra's Algorithm		
46		Huffman trees		
47		Comparison of Design Strategies		
48		Lower Bound Algorithms		
49		Decision Trees		
50		P,NP and NP Complete problems		
51		Backtracking – N queens Problem, Hamiltonian circuit Problem, Sum of Subset Problem		
52		Branch and Bound- Assignment, knapsack problem, Travelling salesman problem		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	*Introduction to The Design and Analysis of Algorithms* Anany Levitin	2	Pearson	2011
Reference Book	R1	*Introduction to Algorithms* Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein	3	Prentice-Hall India	2009

Computer Science And Engineering
(Jan – May 2018)

UE16CS252: Database Management Systems (4-0-0-0-4)

# of Hours: 52

Class #	Chapter title / Reference Literature	Topics to be Covered	% of portions covered	
			% of syllabus	Cumulative %
1	**Unit 1 Introduction and ER Model** T1: Chapter 1,2,3	Introduction, characteristics, advantages	19	19
2		Data models, 3-schema architecture, data independence, database environment		
3		Conceptual Data Modelling – ERD diagram		
4		Entity, Relationships, Attributes, Keys		
5		Weak Entity, Roles, Structural Constraints		
6		Company ERD Example, Design issues		
7		ERD Exercises		
8		Introduction to SQL commands		
9		History, Database Languages, Interfaces		
10		Review of Chapters.		
11	**Unit 2 SQL language** T1: Chapter 8,9.x	SQL Data Definition and Data type	19	38
12		Specifying Basic Constraints in SQL		
13		Schema Change Statements in SQL		
14		Basic Queries in SQL		
15		More Complex SQL Queries		
16		SQL Review and Exercises		
17		Insert, Delete and Update		
18		Assertion, Trigger, Views, other SQL features		
19		Database programming		
20		SQL Review and Exercises		
21	**Unit 3 Relational Model** T1: Chapter 5,6,7.1	Relational Model , Constraints and Schemas	19	58
22		Specifying constraints in SQL		
23		Relational Algebra, Unary Relational Operations: SELECT and PROJECT		
24		Set Theory Operations and Examples		
25		Binary Relational Operations: JOIN. and: DIVISION		
26		Aggregate functions and grouping, Examples		
27		Review of Relational Algebra		
28		Relational Database Design Using ER-to Relational Mapping		
29		ER to Relational mapping examples		
30		Review of Relational Model and Exercises.		
31	**Unit 4 Relational Decomposition FDs, Normal Forma** T1:Chapter 10,11.x	Informal Design Guidelines for Schemas	19	77
32		Functional Dependencies		
33		Functional Dependencies, Minimal Cover		
34		Normal Forms Based on Primary Keys		
35		1st ,2nd and 3rd NF		
36		Boyce-Codd Normal Form and Examples (Lab 10)		
37		Properties of Relational Decompositions		
38		Schema Design		
39		Higher Normal Forms		
40		Chapter Review and Exercises		

41	**Unit 5 Transaction Management, Security and Advanced Topics** T2 : Chapter 16,17,18, and 21	Transactions and ACID Properties, Schedules, Concurrency	23	100
42		Serializability and Recoverability		
43		Lock-Based Concurrency Control, 2PL, Strict 2PL		
44		Deadlocks, Precedence Graphs		
45		Concurrency Control without Locking , Introduction to Crash Recovery		
46		ARIES, Write-Ahead Log Protocol,		
47		Recovery from System Crash		
48		Database Security : Discretionary, Mandatory Access Controls		
49		Advanced topics		
50		Non Relational databases		
51		Review of Units 4 and 5		
52		Review of Course.		

Literature

Book Type	Code	Title & Author	Publication Info		
			Edition	Publisher	Year
Text Book	T1	Fundamentals of Database Systems Elmasri and Navathe	5th or latest	Pearson Education	2006
Text Book	T2	Database Management Systems by Gehrke and RaghuRam Krishnan	3rd or latest	Pearson Education	2003

Computer Science And Engineering
(Jan – May 2018)

UE16CS253: Microprocessors & Computer Architecture (4-0-0-0-4)

# of Hours: 52

Class #	Chapter Title / Reference Literature	Topics to be Covered	% of portions covered	
			% of syllabus	Cumulative %
1	**Unit 1: Introduction to Microprocessor Architecture** T3:1.6 , 2.3, T1:A-3, T3:pg no: 51-55 T4:Chapter 3.1 to 3.5	Overview of Microprocessor : Evolution and Introduction to INTEL Processor	20%	20%
2		Overview of Microcontroller: 8051		
3		Introduction to RISC and CISC Processors RISC Architecture		
4		Instruction set Architecture : Endian, ARM Programmer Model		
5		Addressing Modes		
6		Data Processing Instructions		
7		Conditional Execution Instructions		
8		Sample Programs		
9		Branch Instructions		
10		Multiple Register Transfer Instructions		
11	**Unit 2: Instruction set** T3:6.8 , 5.6, 4.1	Function & Procedures	25%	45%
12		Function & Procedures - continued		
13		Software Interrupts		
14		Sample Programs		
15		Instruction Encoding: Data Processing		
16		Instruction Encoding: Multiple Register Transfer		
17		Instruction Encoding: Multiplication		
18		Instruction Encoding: Branch Instruction		
19		Barrel Shifter		
20		Arm Data path Timing		
21		Introduction to Co-processor		
22		Sample Programs		
23	**Unit 3: Pipelining** T1: Appendix C-1 to C-4	Introduction to Pipelining, 3 stage pipelining	20%	65%
24		5 stage pipelining		
25		Pipeline Hazards		
26		Data Hazards		
27		Data Hazards continued..		
28		Data Hazards continued..		
29		Structural Hazards		
30		Branch Hazards		
31		Branch Hazards continued..		
32		Branch Hazards continued..		
33	**Unit 4 : Memory Hierarchy** T1:Appendix B	Introduction to Memory Hierarchy	20%	85%
34		Introduction to DRAM, SRAM.		
35		Fully Associative, Direct Map Cache Memory		
36		Set Associative Cache Memory		
37		Cache Performance Problems		
38		Basic cache Optimizations		
39		First & Second optimizations		

40		Third & Fourth Optimizations		
41		Fifth & Sixth Optimizations		
42		Fundamentals of Virtual Memory Concept		
43	**Unit 5 : Interrupts and Super Scalar Computers** T1,T2: Chapter - 3.1, 4.1, 4.2	Exceptions & Interrupts	15%	100%
44		Interrupt Handling Mechanism		
45		Introduction to Super Scalar Computers: Instruction Level Parallelism (ILP)		
46		ILP: continued..		
47		Data Level Parallelism (DLP).		
48		Introduction to Single Board Computers : Beagle Board , Intel Edison Board		
49		Arduino Board		
50		Raspberry PI and Sensors.		
51		Revision		
52		Revision		

Literature

Book Type	Code	Title & Author	Publication Info		
			Edition	Publisher	Year
Text Book	T1	Hennessy Patterson	Fifth Edition	MK Morgan Kaufmann	2012
Text Book	T2	Raspberry PI cookbook	Third Reprint 2015	Simon Monk	2015
Text Book	T3	ARM System on Chip, Steve Furber	Second Edition,	Pearson Education	2000
Text Book	T4	ARM System Developer's Guide	Reprint 2009	Elsevier	2009

Computer Science And Engineering
(Jan – May 2018)

UE16CS254: Theory of Computation (4-0-0-0-4)

# of Hours: 52

Class #	Chapter title / Reference Literature	Topics to be Covered	% of portions covered	
			% of syllabus	Cumulative %
1.	**Unit 1 Finite Automata** T1: Chapter 1,2,3	Computers, Computation, Computability	19	19
2.		Automata		
3.		Languages and Grammars		
4.		Deterministic Finite Automata		
5.		Deterministic Finite Automata		
6.		Deterministic Finite Automata		
7.		Non–Deterministic Finite Automata		
8.		Constructing Finite Automata		
9.		Equivalence of Deterministic and Non–Deterministic Finite Automata		
10.		Minimizing Finite Automata		
11.	**Unit 2 Regular Grammar and Languages** T1: Chapter 4,5,6	Regular Expressions,	19	38
12.		Equivalence of Regular Expressions		
13.		Regular Languages and Finite Automata		
14.		Regular Expressions in Practice		
15.		Regular Grammars		
16.		Regular Expressions Construction		
17.		Equivalence to Finite Automata		
18.		Closure Properties of Regular Languages		
19.		Answering Questions about Regular Languages		
20.		Pumping Lemma and identifying Non–Regular Languages		
21.	**Unit 3 Context-Free Grammars** T1: Chapter 7	Idea of Context–Free Grammars	19	58
22.		Nature of Context-Free Grammars		
23.		Constructing CFGs:Linear Grammar		
24.		Constructing CFGs:Non-Linear Grammar		
25.		Introduction to Parsing and Ambiguity		
26.		Parsing and Ambiguity		
27.		Constructing Context–Free Grammars		
28.		Conversion to Chomsky and Greibach Normal Forms		
29.		A Membership Algorithm for Context–Free Languages		
30.		Simple and Linear Grammars		
31.		Machines for Context-Free Languages		

32.	**Unit 4 Pushdown Automata** T1:Chapter 8,9	Constructing Pushdown Automata	19	77
33.		Converting CFGs to PDAs		
34.		Converting PDAs to CFGs		
35.		Non–Deterministic Pushdown Automata		
36.		Nature of Context-Free Languages		
37.		Equivalence of Pushdown Automata and Context–Free Grammars		
38.		Closure Properties		
39.		Questions about Context–Free Languages		
40.		Pumping Lemma for Context–Free Languages		
41.	**Unit 5 Turing Machines and Computability** T1 : Chapter 10,11,12	The Standard Turing Machine	23	100
42.		Constructing Turing Machines		
43.		Variations of Turing Machines		
44.		Universal Turing Machine, Church–Turing Thesis		
45.		Recursive and Recursively Enumerable Languages		
46.		Diagonalization		
47.		Unrestricted Grammars		
48.		Context–Sensitive Grammars and Languages		
49.		Linear–Bounded Automata		
50.		Deterministic Pushdown Automata		
51.		Deterministic Context–Free Languages		
52.		Post Correspondence Problem,Undecidable Problems		

Literature

Book Type	Code	Title & Author	Publication Info		
			Edition	Publisher	Year
Text Book	T1	Theory of Computation: A Problem–Solving Approach, Kavi Mahesh.	Latest	Wiley India	2012

Computer Science And Engineering
(Jan – May 2018)

UE16CS255:Design and Analysis of Algorithms Laboratory (0-0-2-0-1)

Lab#	Program No	Title of the program/ Problem Statement
1	Program 1	Introduction to the lab environment. - Compile and execution of a C program in Linux. - Handling Input-Output formats with large number of test-cases. Brute Force: Implementation of **Sequential Search** algorithm Find the key element in an array of integers using the sequential search algorithm.
2	Program 2	Brute Force: Implementation of **String Matching** algorithm Find a pattern of length m in a text of length n using the naive string matching algorithm.
3	Program 3	Brute Force: Implementation of **Bubble Sort** algorithm Sort a given array of integers using the bubble sort algorithm. Brute Force: Implementation of **Selection Sort** algorithm Sort a given array of integers using the selection sort algorithm.
4	Program 4	Brute Force: Solution for **Traveling Salesperson Problem** Find a solution to the traveling salesperson problem using the exhaustive search method.
5	Program 5	Divide and Conquer: Implementation of **Merge Sort** Sort a given array of integers using the merge sort algorithm. Divide and Conquer: Implementation of **Binary Search** Search for a key element in a sorted array of integers.
6	Program 6	Divide and Conquer: Implementation of **Quick Sort** Sort a given array of student records using the quick sort algorithm.
7	Program 7	Decrease and conquer: Implementation of **Insertion Sort** algorithm Sort a given array of student records using the insertion sort algorithm. Decrease and conquer: Demonstration of **BFS** Find the number of components of an undirected graph given in the form of an adjacency matrix using BFS technique.
8	Program 8	Decrease and conquer: Demonstration of **DFS** algorithms Find the number of components of an undirected graph given in the form of an adjacency matrix using DFS technique. Decrease and conquer: **Topological Sorting** of vertices in a digraph Find a topological order of a directed acyclic graph using the DFS technique.
9	Program 9	Transform and Conquer: Implementation of **Heap Sort** algorithm Sort a given array of student records using the heap sort algorithm. Use bottom up approach for the heap construction. Space and Time Tradeoffs: Implementation of **Sorting by Distribution Counting** algorithm Sort a given array of student records using the distribution counting sort algorithm.
10	Program 10	Space and Time Tradeoffs: Implementation of **Horspool's** algorithm Find a pattern of length m in a text of length n using the Horspool's algorithm.
11	Program 11	Dynamic Programming: Implementation of **Warshall's** algorithm Find the transitive closure of a graph given in the form of an adjacency matrix. Dynamic Programming: Implementation of **Floyd's** algorithm Find all-pairs-shortest-paths of a weighted graph given in the form of a cost matrix.
12	Program 12	Dynamic Programming: Solution for the **Knapsack Problem** Find the solution to a 0/1 Knapsack problem using Dynamic Programming technique.
13	Program 13	Greedy Technique: Implementation of **Prim's** algorithm Find a minimum spanning tree of a weighted connected undirected graph using the Prim's algorithm. Greedy Technique: Implementation of **Dijkstra's** algorithm Find single-source-shortest-paths of a weighted connected graph using the Dijkstra's algorithm.

Computer Science And Engineering
(Jan – May 2018)

UE16CS256:Database Management Systems Laboratory (0-0-2-0-1)

Lab#	Program No	Title of the program/ Problem Statement
1	Program 1	Introduction to database and PostgresSQL. Learn basic psql commands. Create a sample database, populate with data, execute given queries, explain the output produced.
2	Program 2	ERD diagram – Draw the ERD for the given case studies. The case studies will be shared with the students in the lab (a different case study per batch will be used).
3	Program 3	Create the following databases with proper constraints and populate the corresponding tables with at least 10 records. Create the Railway database called **mytraindb** for the given schema of tables. • Trainhalts(id, seqno, stcode, timein, timeout) • Track(stcode1, stcode2, distance) • Station(stcode, name) • Train(id, name) Write the following Queries for the above Railway Schema • Find pairs of stations (station codes) that have a track (direct connection) with distance less than 20Kms between them. • Find the IDs of all the trains which have a stop at THANE • Find the names of all trains that start at MUMBAI. • List all the stations in order of visit by the train 'CST-AMR_LOCAL'. • Find the name of the trains which have stop at Thane, before the 6th station in the route of the train.
4	Program 4	create the following databases with proper constraints and populate the corresponding tables with at least 10 records. Create the university database for the given schema of tables. • Student(id, name, dep_name, tot_credits) • Takes(id, course_id, sec_id, sem, year, grade) • Section(course_id, sec_id, sem, year, building, room_no, time_slot_id) • Time_slot(time_slot_id, day, start_time, end_time) • Classroom(building, room_no, capacity) • Teaches(i_id, course_id, sec_id, sem, year) • Instructor(i_id, name, dep_name, salary) • Dept(dep_name, building, budget) • Advisor(id, i_id) • Course(course_id, title, dep_name, credits) • Prereq(course_id, prereq_id)

5	Program 5	Write the following simple SQL Queries on the University Schema • Find the names of all the students whose total credits are greater than 100 • Find the course id and grades of all courses taken by any student named 'Tanaka' • Find the ID and name of instructors who have taught a course in the Comp. Sci. department, even if they are themselves not from the Comp. Sci. department. To test this query, make sure you add appropriate data, and include the corresponding insert statements along with your query. • Find the courses which are offered in both 'Fall' and 'Spring' semester (not necessarily in the same year). Additional queries 1. Find the names of all the instructors from Comp. Sci. department 2. Find the course id and titles of all courses taught by an instructor named 'Srinivasan' 3. Find names of instructors who have taught at least one course in Spring 2009
6	Program 6	Write the following SQL Queries on the University Schema (use nested queries) • Find the id and title of all courses which do not require any prerequisites. • Find the names of students who have not taken any biology dept courses • • Write SQL update queries to perform the following • Give a 10% hike to all instructors • Increase the tot_creds of all students who have taken the course titled "Genetics" by the number of credits associated with that course. • For all instructors who are advisors of at least 2 students, increase their salary by 50000. • Set the credits to 2 for all courses which have less than 5 students taking them (across all sections for the course, across all years/semesters).
7	Program 7	SQL DDL commands and Updates • Each offering of a course (i.e. a section) can have

		many Teaching assistants; each teaching assistant is a student. Extend the existing schema (Add/Alter tables) to accommodate this requirement. • According to the existing schema, one student can have only one advisor. • Alter the schema to allow a student to have multiple advisors and make sure that you are able to insert multiple advisors for a student. • Write SQL queries on the modified schema. You will need to insert data to ensure the query results are not empty. • Find all students who have more than 3 advisors • Find all students who are co-advised by Prof. Srinivas and Prof. Ashok. • Find students advised by instructors from different departments. etc. Write SQL queries for the following: • Delete all information in the database which is more than 10 years old. Add data as necessary to verify your query. • Delete the course CS 101. Any course which has CS 101 as a prereq should remove CS 101 from its prereq set. Create a cascade constraint to enforce the above rule, and verify that it is working.
8	Program 8	Introduction to MongoDb. Connecting to the database, creating database and collection and populating it with documents. Documents with varying fields such as books, film, person, hospital etc. can be added to the collection. Use find() and findOne() for querying the database.
9	Program 9	Write complex queries to retrieve the documents from the collection using various $opertors
10	Program 10	Mini-Project
11	Program 11	Mini-Project
12	Program 12	Mini-Project
13		Final Assessment

Computer Science And Engineering
(Jan – May 2018)

UE16CS257:Microprocessor & Computer Architecture Laboratory (0-0-2-0-1)

Lab#	Program No	Title of the program/ Problem Statement
1	Program 1	Debugging assembly language code using a simulator.
2	Program 2	Program using data processing instructions.
3	Program 3	Program using logical and looping instructions.
4	Program 4	Program using functions and subroutine instructions.
5	Program 5	Program to perform operations on string data.
6	Program 6	Program using recursion.
7	Program 7	Program to interface with input / output devices.
8	Program 8	Install setup and configure a single board microcomputer system.
9	Program 9	Mini Project
10	Program 10	Mini Project
11	Program 11	Mini Project
12	Program 12	Mini Project
13	Program 13	Mini Project