Computer Science And Engineering
(Aug '17 – Dec '17)

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.

Vth Semester (2015-2019)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Course Type
			L	T	P	S		
1	UE15CS301	Computer Networks	4	0	0	0	4	CC
2	UE15CS302*	Introduction to Operating Systems	4	0	0	0	4	CC
3	UE15CS303	Principles of Programming Languages	4	0	0	0	4	CC
4	UE15CS304	Computer Networks Laboratory	0	0	2	0	1	CC
5	UE15CS305	Introduction to Operating Systems Laboratory	0	0	2	0	1	CC
6	UE15CS306	Principles of Programming Languages Laboratory	0	0	2	0	1	CC
	Elective - I							
7	UE15CS311**	Advanced Algorithms	4	0	0	0	4	EC
8	UE15CS313$	Advanced Database Management Systems	4	0	0	0	4	EC
9	UE15CS314$	Big Data	4	0	0	0	4	EC
10	UE15CS315	Multimedia Computing	4	0	0	0	4	EC
	Elective - II							
11	UE15CS321**	Computer Graphics and Visualization	4	0	0	0	4	EC
12	UE15CS322$$	Data Analytics	4	0	0	0	4	EC
13	UE15CS323$$$	Fuzzy Logic	4	0	0	0	4	EC
14	UE15CS325**	Artificial Intelligence	4	0	0	0	4	EC
		Total	20	0	6	0	23	

Note: Prerequisite courses - * UE15CS202; ** UE15CS251, *** UE15CS253, $ UE15CS252, $$ UE15CS203, $$$ UE15CS205,

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS301 – Computer Networks (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title / Reference Literature	Topics to be Covered	% of portions covered	
			% of syllabus	Cumulative %
1	**Unit #1**	**Introduction to communication and networking**	**19.5**	**19.5**
2		**Network edge**		
3	**Introduction to Computer Networks & the internet**	Access Networks & Physical media		
4		**Network core**		
5		Circuit switching		
6		Packet Switching		
7	T1 :Chapter 1	**Layered architecture- Introduction**		
8	Sec 1.1,Sec 1.2, Sec 1.3,Sec 1.5 Sec 1.8	Responsibility and functions of Application layer Transport layer		
9		Responsibility and functions of Network layer		
10		Responsibility and functions of Link layer & Physical layer		
11	**Unit #2**	**Network applications-Principles**	**19.5**	**39.0**
12		**The WEB & HTTP**- Overview		
13	**Application layer**	HTTP Message format		
14	T1:	Web caching & Conditional GET		
15	Chapter 2	**DNS**- Services		
16	Sec 2.1	DNS Hierarchy		
17	Sec 2.2	DNS Records		
18	Sec 2.5	**Socket Programming** : Creating network applications		
19	Sec 2.7	Socket programming with UDP		
20	Sec 2.8	Socket programming with TCP		
21	**Unit #3**	**Introduction to Transport layer services**	**15.5**	**54.0**
22	**Transport Layer**	**Principles of Reliable data transfer**- Stop-N-Wait protocol		
23	T1:	Sliding window concepts- Go back N protocol		
24	Chapter 3	**Transmission Control Protocol**- Features		
25	Sec 3.1	TCP Header		
26	Sec 3.4	TCP Connection Management		
27	Sec 3.5	TCP Flow control and error control		
28	Sec 3.7	TCP Congestion control		
29	**Unit #4**	**Internet Protocol**- Datagram Format-Fragmentation		
30		IPV4 addressing – Classless addressing		
31	**Network Layer**	Subnet principles- Forwarding		
32	T1 :	**IPV6** - Features		

33	Chapter 4 Sec 4.4 Sec 4.5 Sec 4.6	DHCP	**23.00**	**77.00**
34		NAT		
35		ICMP		
36		**IPV6** Header		
37		**Routing algorithms**		
38		RIP- Distance vector algorithm		
39		OSPF-Link state algorithm		
40		Hierarchical routing-BGP		
41	**Unit #5** **Link layer Local Area Networks** T1 : Chapter 5 Sec 5.1,Sec 5.2, Sec 5.3,Sec 5.3.1, Sec 5.3.2,Sec 5.4 Sec 5.4.1,Sec 5.4.3 Sec 5.4.4,Sec 5.6 Sec 5.7	**Introduction to link layer**	**23.00**	**100**
42		Error detection and correction techniques-Principles		
43		Parity, CRC		
44		**Switched local area network** – Building blocks of LAN		
45		LAN Switch-working principles		
46		**Introduction to MAC protocols**		
47		CSMA/CD		
48		VLANs		
49		Data center networking *(Guest Lecture)*		
50		Retrospective : A day in the life of a Web Page request		
51		All protocols put together		
52		Consolidation of all the layers		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Computer Networking - A Top-down approach, James F Kurose, Keith W	6	Pearson	2012
Reference Book	R1	Computer Networks- A Top-down Approach -Behrouz A Forouzan, FirouzMosharraf	1 [Special Indian Edition]	Pearson	2012

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS302: Introduction to Operating Systems (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title / Reference Literature	Topics to be covered	% of Portions Covered	
			% of syllabus	Cumulative %
1	**UNIT #1 Introduction to OS, Introduction to Processes** T1: Ch.1,2,3	Computer System Architecture	19.2	19.2
2		Protection and Security,Kernel Data Structure		
3		Open Source Operating System.		
4		Operating System Services		
5		System Programs		
		Operating System Structure,System calls		
6		Introduction to Processes:Process and its address space,Simple two state model		
7		Process life cycle		
8		Process Management: Advanced state model for processes		
9		Context switching, Examples of IPC		
10	**UNIT #2 Processes and Threads, Process Synchronization, Deadlocks** T1: Ch.4, 6,7	Introduction to Thread: Overview-Multithreaded Models	19.2	38.4
11		Implicit-Operating System Examples		
12		Process Synchronization: The Critical-Section Problem,Synchronization Hardware		
13		Semaphores		
14		Monitors		
15		Synchronization examples		
16		Deadlocks: Deadlock Characterization		
17		Methods for Handling Deadlocks, Deadlock Avoidance		
18		Deadlock Detection		
19		Recovery from Deadlock		
20	**UNIT #3 CPU Scheduling, Memory** T1: Ch. 5, 8,9	CPU Scheduling:Basic Concepts – Scheduling Algorithms	19.2	57.6
21		Multiple Processor Scheduling		
22		Memory: Background, Swapping		
23		Contiguous Memory Allocation		
24		Segmentation, Paging		
25		Paging		
26		Virtual Memory: Background, Demand Paging		
27		Copy-on-Write, Page Replacement		
28		Allocation of Frames,Thrashing,TLB		
29		Thrashing, TLB		

30	**UNIT #4 File System Interface,File-System Implementation** T1: Ch. 10,11	File System Interface: File Concept	19.2	76.8
31		Access Methods, Directory and Disk Structure		
32		File-System Mounting		
33		File-System Structure		
34		File-System Implementation		
35		Directory Implementation, Allocation Methods		
36		Allocation Methods		
37		Free-Space Management		
38		Efficiency and Performance		
39		Review of File System		
40	**UNIT #5 Mass Storage-Structure, I/O Systems** T1: Ch. 12,13	Mass Storage-Structure**:** Overview of Mass-Storage Structure	23.2	100
41		Disk Structure, Disk Attachment		
42		Disk Scheduling		
43		Disk Management, Swap-Space Management**.**		
44		I/O Systems:Overview, I/O Hardware		
45		I/O Hardware		
46		Application I/O Interface		
47		Kernel I/O Subsystem		
48		Transforming I/O Requests to Hardware Operations, Performance.		
49		Dynamic operations: booting an operating system		
50		Review of mass storage		
51		Review of I/O systems		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	9th	John Wiley&Sons	2013
Text Book	T2	Operating Systems, Internals and Design Principles, William Stallings	7th	Prentice Hall	2012
Text Book	T3	Operating Systems, Harvey Deitel, Paul Deitel, David Choffnes	3rd	Prentice Hall	2009
Text Book	T4	Modern Operating Systems, Andrew S Tannenbaum	3rd	Pearson	2013

Note: Pre-requisite for "Introduction to Operating Systems"(UE15CS302) is that students should have cleared "Data Structures" (UE15CS202)

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS303: Principles Of Programming Languages (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Preliminary Concepts; Names, Binding, Type Checking and Scopes;** T1:1 and 5	Preliminaries: Reasons for studying, concepts of programming languages, Programming domains	19%	19%
2.		Language Evaluation Criteria		
3.		influences on Language design, Language categories,		
4.		Programming Paradigms		
5.		Programming Paradigms - Java as Case study		
6.		Programming Language Implementation – Compilation and Virtual Machines		
7.		Names, Binding, Type Checking and Scopes: Names, Variables,		
8.		Type bindings, Type Inferencing, Type Checking,		
9.		Strong Typing, Type Equivalence, Scope,		
10.		Scope and Lifetime, Referencing Environments.		
11.	**Unit #2** **Data types, Control Structures:** T1: Ch 6.1 – 6.9, T1: Ch 7.1, 7.2, 7.4, 7.5 - 7.8, 8.1 - 8.6	Data types: Introduction, primitive data types,	23%	42%
12.		Character string types		
13.		User defined ordinal types		
14.		Array Types		
15.		Associative Arrays, record types		
16.		Union types		
17.		Pointer and reference types		
18.		Expressions and Statements: Arithmetic expressions		
19.		Relational and Boolean expressions, Short Circuit Evaluation		
20.		Assignment Statements , Mixed mode assignment		
21.		Control Structures: Selection Statements, Iterative Statements		
22.		Unconditional Branching, Guarded commands		

23.	**Unit #3** **Sub programs and Blocks, Abstract Data Types, Object OrientedProgramming concepts** T1 : 8, 9, 11 and 12	Subprograms and Blocks: Fundamentals of sub-programs, Design issues of subprograms	23%	65%
24.		local referencing environments, parameter passing methods		
25.		Parameters that are sub-program names, Overloaded sub-programs, Generic sub-programs		
26.		Design issues for functions user defined overloaded operators, co routines		
27.		Closures: Object closures and Function closures.		
28.		Abstract Data types and Encapsulation Constructs: Abstractions and encapsulation, introduction to data abstraction, design issues		
29.		language examples: C++ , Java		
30.		Parameterized ADT: C++,Java,		
31.		Encapsulation constructs		
32.		Support for object oriented programming with appropriate language examples		
33.		Support for object oriented programming with appropriate language examples		
34.		Support for object oriented programming with appropriate language examples		
35.	**Unit #4** **Concurrency, Exception Handling,** T1:13,14	Concurrency: Subprogram level concurrency,	16%	81%
36.		Concurrency control using semaphores, monitors,		
37.		message passing, Java threads,		
38.		Python threads (Handouts)		
39.		Exception handling : Exceptions, Specifications,		
40.		Exception Propagation		
41.		Exception handlers in Python (Handouts)		
42.		Exception handlers in Java.		
43.	**Unit #5** **Logic Programming and Functional Programming**	Guest Lecture		
44.		Logic Programming Languages : Introduction and overview of logic programming		
45.		Basic elements of prolog		
46.		Application of logic programming.		

47.	**T1: 13, 14, 15 & 16**	Functional Programming Languages: Introduction	19%	100%
48.		Fundamentals of FPL, LISP		
49.		Fundamentals of Haskell		
50.		Application of Functional Programming Languages		
51.		Comparison of functional and imperative Languages		
52.		Practice problems using Prolog and Haskell		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Concepts of Programming Languages, Robert .W. Sebesta	10th	Pearson Education	2012
Reference Book	R1	Programming Language Pragmatics, Michael L. Scott	3rd	Elsevier	2009
	R2	Programming Languages Design and Implementation – Pratt and Zelkowitz	4th	PHI/Pearson Education	2001

UE15CS304: Computer Networks Laboratory(0-0-2-0-1)

SL.NO	LIST OF PROGRAMS
1)	**Learn and Understand Network Tools** **1.Wireshark,** 1.1 Examining an IPv4 Packet Header 1.2 Examining an IPv6 Packet Header 1.3 Analyse HTTP packet **2. TCPDUMP** **2.1** capture packets and try out all the options and record in your observation **3. PING** **3.1** Test the connectivity between 2 systems and try all the options **4.TRACEROUTE** 4.1 track the packet from source to remote host and try all the options **5.NETCAT** 5.1 Establish communication between client and server 5.2 send 100 bytes of data file from client to server
2)	**Use Simulator and Learn Networking:Cisco packet racer** **Excercise1:** Create simple client server topology and receive the packets from client. try connecting multiple clients **Excercise2:Create a given topology and perform the following** **1**) ping local networks 2) ping across the networks (doesn't work) 3) add routing tables 4) ping across the networks(it works now because knows the path
3)	**Understand persistent and non-persistent HTTP connections and corresponding performance impact.** Create a web page with N (e.g. 10) embedded images. Each image should be of minimum 2 MB size. Configure your browser (firefox) with following settings (each setting requires repeat of experiment) • Non persistent connection • 2 persistent connections • 4 persistent connections • 6 persistent connections

	• 10 persistent connections. **Observation:** Note down the time taken to display the entire page in each of the settings. Ensure that (cache is cleared before starting the web request). Explain the response time differences. What is the optimal number of persistent connections for best performance. Explain your answer.
4)	**Understand working of HTTP headers:** Conditional Get: If-Modified-Since HTTP Cookies: Cookie and Set-Cookie Authentication: Auth-Basic Design a web page that has one embedded page (e.g. image) and sets a cookie and enables authentication. You are required to configure the web server (e.g. apache) with authentication mechanism. Show the behavior of conditional get when embedded objects is modified and when it is not (you can just change the create date of the embedded object). Decode the Basic-Auth header using Base64 mechanism as per the password setup. **Observation**:Show the behavior of browser when is cookie is set and when cookie is removed.
5)	**Using Cisco packet tracer understand the life of packet in internet.** **Create the following topology in packet tracer.** /---DNS A – R1—R2 \--- Web Server Open the browser in A and access the webserver using sitename (not using IP Address). Traverse each packet (in simulation mode) and answer the following for each packet Src IP, Dstn IP, Src Mac, Dstn MAC, pkt type (e.g. DNS, ARP, HTTP, TCP) **Observation:** Does the number packets traversed in the network change with second invocation of web request.
6)	**Network programming:Client -Server Socket programming** 1. using TCP 2. using UDP 3. multiple clients
7)	**Create an Ipv4 subnet for the given topology**

	Study the stating routing in a real network. Establish the network setup as below A -- R1 -- R2 –-B R1 and R2 are the desktops with two network interfaces. These interfaces can be obtained by connected USB to Ethernet convertor. **Observation:** Define MTU at R1 as 800 bytes, and at R2 as 600 bytes.Send 2000 byte UDP packet from A to B. Explain the packet transmission behavior. How many packets are transmitted by A, how many packets are received by B (including fragmentation if any). Did A receive any packets? If so, explain what are those packets and why were they received.
8)	**Create an Ipv6 subnet for the given topology** Study the stating routing in a real network. Establish the network setup as below A -- R1 -- R2 –-B R1 and R2 are the desktops with two network interfaces. These interfaces can be obtained by connected USB to Ethernet convertor.
9)	**Understand ICMP Redirect and implement the following** 1) ICMP Echo Request/reply 2) ICMP Redirect
10)	**Understand TTL Expiry.** Create a network with following topology using USB to Ethernet adaptor on desktops A --R1 --R2 \ / \ / R3 Create a routing loop among R1, R2 and R3 using default route i.e. R1 sends to R2, R2 sends to R3 and R3 sends to R1. Send 1 ICMP packet from A to unknown destination. **Observation:**Analysis packets routing in the loop and finally generation TTL expiry message. Why is this TTL expiry message does not encounter routing loop. General the ICMP packet with different TTL Value and ensure TTL Expiry is generated by R2 or R3 and not by R1.

11)	**Implementation of OSPF/BGP protocol** Study and configure Link state routing protocol OSPF. Create the given network topology Run OSPF among routers R1, R2, R3 and R4. **Observation:**Once routing has converged. Start continuous ping from A to B and note down the path. Once ping starts responding, break the link R1-R2 which is being used. Note how much time it takes to converge to new routes by OSPF. When you restore the link back, how much time it takes to restore.
12)	**Study unknown unicast, broadcast flooding and broadcast storm in L2 network.**Create the following network i.e. connects two switches on two ports. A --- S1---S2 --- C **Observation**: Send one ICMP packet from A to C. Run wireshark on both A and C. Study the packets received. Explain the behavior. Is it broadcast flooding or storm?Now reset the network. Define static ARP entry for C in A. Now send the same single ICMP packet from A to C. What is the behavior change. Explain why. In the above setup (after reset), add another link between S1 and S2. Send the same a series of ICMP packets from A to C. Explain the behavior of network and what happens to responses.
13)	**Understand SDN** Understand controller functionality in SDN network and using it to change the functionality of network dynamically from hub to a L2 switch. Create the following topology H1 --- S --- H2 | H3 Invoke each of the host in its separate window. Invoke from H1 to H2 and run wireshark for each of H1, H2 and H3. Using controller first configure (or invoke) the controller so that it responds with hub functionality. When the network (Mininet is running), reinvoke the controller with L2 switch functionality. What is the difference you see between two invocations, especially at H3. Explain.

UE15CS305: Introduction to Operating Systems Laboratory (0-0-2-0-1)

SL.NO	LIST OF PROGRAMS
1	Execute and familiarize with Linux environment and commands.
2	Write a program to create a child process, terminate and get status.
3	Write a set of programs to create child process which does computation and returns the result of computation to parent process.
4	Write a program to implement classical inter process communication problem (Producer – Consumer)
5	Write a program to implement Banker's algorithm for mutual exclusion problem.
6	Write program to implement the solution to Diners Philosopher problem.
7	Write a program to implement Round Robin scheduling algorithm. Write a program to implement Priority Scheduling algorithm.
8	Write a program to implement Shortest-Job-First scheduling algorithm.
9	Write a program to implement paging using Best-fit and Worst-fit algorithms.
10	Write a program to implement Least Recently Used page replacement algorithm.
11	Write a program to implement optimal page replacement algorithm.
12	Write a program to implement FCFS/SCAN/C-SCAN disk scheduling algorithms.
13	Simulate the inode structure. Implement data in a file through pointers. Support read, write and seek operations.
14	Simulate directory structure.Support functions like mkdir, rmdir, mv, ln, rm, ls and touch

UE15CS306: Principles Of Programming Languages Laboratory (0:0:2:0:1)

SL.NO	LIST OF PROGRAMS
1	Using debuggers, examine runtime features. i. Write a sample C program with errors for debugging purpose: Use gdb ii. Experiment with various options available in gdb (Reference Command list : b, run, bt, p, c, n, s, quit, help, INTERRUPT, break, finish, until .....etc) iii. Use nm utility and note the observations. iv. Use ulimit command and note the observations.
2	Write C programs to demonstrate following cases for static variables: File scope Function scope Can the static variable be externed? Declaring the static variable in the header file and examining the behavior in multi ple implementation files Scope of a static variable within a compilation unit External and internal linkages (Note down your observation and conclusion in each case.) a) Write programs in C to demonstrate the differences between static and global variables. b) Study Name mangling with respect to Variable and Function names in C++. How does extern C linkage prevent name mangling? Demonstrate with suitable code.
3	Using Valgrind and memcheck tools, check the dynamic memory utilization and memory leaks. (memory profilers)
4	Examine assignment operators - i. in python– list() and sublist(), nested list, copy and deep copy. ii. in C – Assignment of Structures. iii. Examine structural and name equivalence in C- typedef.

5	Demonstration of Scope of Goto in C, i. Use Goto statement in a function. Demonstrate the transfer of control within the function. ii. Use Goto statement in one function and try to transfer control to another function by defining label in the second function. What is your observation? iii. Use Goto statement in one file and try to transfer control to another file by defining label in that file. What is your observation? iv. Can you use Goto statement in C to jump into/out of a block? Verify with suitable code. v. Non-local goto a) Understanding the use of setjmp() and longjmp() in C. Demonstrate the use of setjmp() and longjmp() functions to transfer control from one function to another.
6	Variable # of arguments for a function(/method) i. in C: Study the macros that are defined in stdarg library. (You can use man stdarg.) a) Study va_list data type, macros va_start, va_arg, va_end macros for manipulating list of arguments b) Observe and note the parameters needed to pass variable no. Of arguments in C ii. In Python, check for a) var. Args, b) Var. Dictionary (key-value pairs) c) Combination of both: Note down the restriction in the usage of these combinations. iii. Check the possibility of combining more than one type of values while passing variable no. Arguments in C, Python and Java. iv. Are the arguments type checked? Try passing wrong types. v. When the parameter passed to a function is an array, should it be considered as variable no. Of arguments? Write your comment on this.

7	Demonstrate i. Closures in Python. Also check the following. a) Variable with scope in outer function will be having a read only scope in the inner function. Check this fact, by reading the non local variable, by writing to it - (will this be taken as a local variable, when you assign?), try reading the value and then assign a value in the next statement. What would happen, if the variable in the outer scope is a mutable object as a list? b) Try the same things mentioned in the above question on list variable - which is mutable object. c) What is non local variable and global variable? Are they same? What is the effect of this on closures? ii. Understand the concept of decorators in Python. Decorate the division function written by you with an additional functionality to check for division by zero. iii. Demonstrate Callbacks a) In C – use any appropriate example. b) In python - use any appropriate example. c) Use of Functors in C++
8	Demonstrate stack smashing in C. Is it possible to instruct the C compiler not to look for stack smashing situation? If Yes, how? ii. Understand the difference between tail recursion and normal recursion? iii. List out the languages that do tail call optimization (TCO) and which do not? iv. Write a tail recursive function for factorial computation and string reversal in C. v. Try compiling the C program with -O3 option (highest optimization level) and check if it does TCO. (Hint: compile with -g option, then use objdump -d a.out to look at the disassembly of the compiled code, to see whether the recursive call is taken as a function call or as a jump statement) vi. Demonstrate the function call by keyword parameters in python. Can we do the same in C & Java? vii. Among C, Python, Java, demonstrate the default parameters ?
9	Interface and inner classes in Java: show the accessibility of enclosing class members from the enclosed class. Nesting can be implemented as in the following cases. Test the accessibility of members of outer class when they are declared as private or public or final. Test the modifiability of these outer class members from inner class. • Using Static nested class • Using non-static nested class(inner class) • Also demonstrate the declaration and usage of a local class

10	Generic Methods in Java. i. Illustrate with example how we can print values of different type using a single Generic method ii. Illustrate use of extends in Java Generic Method declaration (for the generic parameter) iii. Demonstrate the use of wild card parameters. Can we define bounded wild card parameter?
11	i. Demonstrate use of @Override annotation in context of Inheritance in Java ii. Examine usage of final in Java for : variables, classes, methods. Note down your observations. iii. Compare private method and final method in Java. iv. Demonstrate downcasting in Java/C++. Note down your observations. v. Demonstrate with suitable code, multiple inheritance in Java &Python.
12	Examine and observe i. The concurrent execution using multithreading. ii. Synchronization among multiple threads.

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS311: Advanced Algorithms (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Basics of Complexity** T1 Ch 3, 17	Motivation; Introduction to the course;	15.5	15.5
2.		Asymptotic notation		
3.		Standard notations and common functions		
4.		Examples of complexity analysis		
5.		Aggregate Analysis		
6.		The accounting method		
7.		The potential method		
8.		Dynamic tables		
9.	**Unit #2** **String algorithms** T1 Ch 32 **Notes on Boyer-Moore algorithm and Suffix Trees**	The naive string matching algorithm Boyer-Moore algorithm	23	38.5
10.		The Robin-Karp algorithm		
11.		String matching with finite automata		
12.		The Knuth-Morris-Pratt algorithm		
13.		Suffix Trees		
14.		Mathematical properties of suffix trees		
15.		Applications of suffix trees		
16.		Applications of suffix trees		
17.		Regular expression searches using suffix trees		
18.		Regular expression searches using suffix trees		
19.		Finding all maximal string matches		
20.		Finding all maximal string matches		
21.	**Unit #3** **Graph / Tree algorithms** T1 Ch 26, 18, 19, 20	Flow Networks and The Ford-Fulkerson method	15.5	54
22.		Maximum bipartite matching		
23.		B-Trees		
24.		B-Trees		
25.		Binomial Heaps		
26.		Binomial Heaps		
27.		Fibonacci Heaps		
28.		Fibonacci Heaps		
29.		Representation of polynomials	23	87

30.	**Unit #4** **Polynomials** T1 Ch 30	DFT		
31.		FFT		
32.		Efficient FFT implementations		
33.		Efficient FFT implementations		
34.		Efficient FFT implementations		
35.		Efficient FFT implementations		
36.		Efficient FFT implementations		
37.		Efficient FFT implementations		
38.		Efficient FFT implementations		
39.		Efficient FFT implementations		
40.		Efficient FFT implementations		
41.	**Unit #5** **Number Theoretic Algorithms; Randomized Algorithms** T1 Ch 31, 5, 34	Elementary number-theoretic notations and GCD	23	100
42.		Modular arithmetic		
43.		The RSA public-key cryptosystem		
44.		The RSA public-key cryptosystem		
45.		Primality testing		
46.		Integer factorization		
47.		Indicator random variables		
48.		Randomized algorithms		
49.		Randomized algorithms		
50.		NP-completeness		
51.		NP-completeness		
52.		Intractability notion		

Literature

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Introduction to Algorithms, T. H Cormen, C E Leiserson, R L Rivest and C Stein	4th	Prentice-Hall of India	2010
	T2	The Algorithm Manual, Steven Skiena	2nd	Springer	2012
	T3	Randomized Algorithms, Rajeev Motwani and PrabhakarRaghavan		Cambridge University Press	1995

Note: Pre-requisite for "Advanced Algorithms"(UE15CS311) is that students should have cleared "Design and Analysis of Algorithms" (UE15CS251).

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS313 : Advanced Database Management Systems (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Review of Relational model** R2 R1: Ch 25 T1: Ch 27, 28 +Notes	Course Introduction, Lesson Plan, Review of Relationa model and databases,	20	20
2.		ER Model,SQL,Transactions,Data warehousing, Star Schema		
3.		Data warehousing, Star Schema examples, SQL Analytical Functions		
4.		Partitioning, Indexing, Materialized Views, Query rewrite		
5.		Case Studies : Postgres, Oracle		
6.	**Unit #2** **Parallel and distributed databases** T1: Ch 17, 18, 19 +Notes	Scaling RDBMS: Parallel & Distributed Databases, Architectures for parallel & distributed databases	20	40
7.		Distributed databases and Issues, Catalogs, Queries		
8.		Transactions, Concurrency & Locking		
9.		CAP Theorem, NoSQL: Need for NoSQL, NoSQL Introduction: Categories		
10.	**Unit #3** **NoSQL Databases-Column oriented and Document stores**: T2: Ch 9, 10 T3: Ch 5 +Notes	NoSQL Column Oriented databases and use cases. Data Model & access	20	60
11.		Column Oriented databases : Replication, Sharding, Consistency Models, Locking & Storage		
12.		NoSQL Document Oriented databases and use cases,		
13.		Case Studies: Parallel & Distributed		
14.		Document Oriented databases : Replication, Sharding, Consistency Models, Locking &StorageCase Studies: Cassandra &MongoDB		
15.		Case Studies: Cassandra &MongoDB		
16	**Unit #4**	NoSQL Key-value stores and use cases, Data Model & access	20	80

17	**NoSQL Databases-Key-Value stores and Graph Database** T2: Ch 8, 11 T3, Ch 3, 7 +Notes	Key-value stores : Replication, Sharding, Consistency Models, Locking & Storage		
18		NoSQL Graph databases and use cases, Data Model and access		
19		Graph databases : Replication, Sharding, Consistency Models, Locking & Storage		
20		Case Studies: Riak, Neo4J		
21	**Unit #5** **In-Memory databases & Specialty Databases** T2: Ch 8 T3: Ch 4 +Notes	In Memory Databases, VoltDB(RDBMS) and Redis(Key-value store)	20	100
22		Hadoop, HDFS, Map Reduce programming Model & Spark		
23		Spatial Databases, Geographic Information Systems (GIS)Other Specialty Databases, Temporal, Deductive Genomic Databases		
24		Other Specialty Databases, Temporal, Deductive Genomic Databases		
25		Case Studies: In-memory databases, Hadoop, HDFS, Map-reduce & Spark		
26		Guest Lecture		

Literature

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Introduction to Algorithms, T. H Cormen, C E Leiserson, R L Rivest and C Stein	6th	McGraw Hill	2013
	T2	NoSQL Distilled, Pramod J. Sadalage and Martin Fowler		Addison Wesley	2012
	T3	Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, Eric Redmond & Jim R. Wilson		O'Reilly	2012
Reference	R1	Database Management Systems, Raghu Ramakrishnan	3rd	McGraw-Hill	2002
	R2	Fundamentals of Database Systems, Elmasri and Navathe	7th	Pearson Education	2015
	R3	Readings in Database Systems, Michael Stonebraker, et. al. (Editors)	5th		2013

Note: Pre-requisite for "Advanced Database Management Systems"(UE15CS313) is that students should have cleared "DBMS" (UE15CS252)

Computer Science And Engineering
(Aug '17 – Dec '17)

UE15CS314: Big Data (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Introduction** T1	Introduction: Data & Domain	19.23	19.23
2.		Introduction – Big Data Characteristics, 4Vs,		
3.		HDFS		
4.		Map-Reduce architecture 1		
5.		Algorithms 1: Matrix Vector Multiplication and algorithm complexity		
6.	**Unit #2** **Big Data Infrastructures for Compute/Storage** T2	Hadoop Ecosystem		
7.		Algorithms 2: Relational Operators		
8.		HIVE		
9.		HBASE	19.23	38.46
10.		MapReduce - clustering algorithms		
11.	**Unit #3** **Algorithms for Big Data** T3	Project Description		
12.		Introduction to Scala		
13.		Alternatives to MR - Workflow, Spark		
14.		Spark details and architecture		
15.		Guest Lecture 1		
16.	**Unit #4** **Real time Big Data Analytics T1,T2**	Introduction to Stream processing and stream processing algorithms	19.23	57.69
17.		Streaming Spark		
18.		Storm		
19.		Stream Clustering Algorithms		
20.	**Unit #5** **Machine Learning and Resource Management for Big Data**	Collaborative filtering and content based filters	15.38	73.07
21.		Distributed Deep learning at scale, tensorflow		
22.		MLIB/Mahout		
23.		MapReduce algorithms for text processing		
24.		YARN resource manager		
25.		MESOS resource manager	26.92	100
26.		Scalable Graph Algorithms		

Literature

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Hadoop: The Definitive Guide – Tom White	4th	O'Reilly	2015
	T2	Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark, and More Hadoop Alternatives, Vijay SrinivasaAgneeswaran	1st	Pearson	2014
	T3	Mining of Massive Datasets, AnandRajaraman, Jure Leskovec, Jeffrey D. Ullman	2nd	Cambridge University Press	2014
Reference Book/Papers	R1	Spark: cluster computing with working sets ,Zaharia M, Chowdhury M, Franklin MJ, Shekhar S, Stoica I.. HotCloud. 2010			
	R2	Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, Zaharia, Matei, et al., Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.			
	R3	Learning Spark, MateiZaharia; Patrick Wendell; Andy Konwinski, Holden Karau	1st	O'Reilly	2015
	R4	Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. In NSDI 2011 Mar 30 (Vol. 11, pp. 22-22). Vavilapalli VK, Murthy AC, Douglas C, Agarwal S, Konar M, Evans R, Graves T, Lowe J, Shah H, Seth S, Saha B			
	R5	Apache Hadoop™ YARN: Moving Beyond Mapreduce and Batch Processing With Apache Hadoop™ 2 , Jeff Markham, Joseph Niemiec, Vinod Kumar Vavilapalli, Arun C. Murthy, Doug Eadline, "Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data 2014, Toshniwal A, Taneja S, Shukla A, Ramasamy K, Patel JM, Kulkarni S, Jackson J, Gade K, Fu M, Donham J, Bhagat N. Storm@ twitter. In Jun 18 (pp. 147-156). ACM.			

Note: Pre-requisite for "Big Data"(UE15CS314) is that students should have cleared "DBMS" (UE15CS252)

Computer Science And Engineering
(Aug 17 – Dec 17)

UE15CS315: MULTIMEDIA COMPUTING (4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/ Reference Literature	Topic To be Covered	% of portions covered	
			% of syllabus	Cumulative %
1	**Unit #1 Introduction to Multimedia, Graphics and Image Data Representation, Color in Image and Video** T1: 1.1-1.4, 3.1-3.2, 4.1-4.3	What is multimedia?	**23**	**23%**
2		Multimedia and hypermedia		
3		World wide web		
4		Overview of the multimedia software tools		
5		Graphics/image data types		
6		Popular file formats		
7		Color science		
8				
9		Color models in images		
10				
11		Color models in video		
12				
13	**Unit #2 Fundamental Concepts in Video, Basics of Digital Audio** T1: 5.1-5.3, 6.1-6.3	Types of video signals	**15.4**	**38.4%**
14				
15		Analog video		
16		Digital video		
17		Digitization of sound		
18		MIDI: Musical instruments digital interface		
19		Quantization and transmission of audio		
20				
21	**Unit #3 Lossless Compression Algorithms, Lossy Compression Algorithms** T1: 7.1-7.7, 8.1-8.9	Introduction, Basics of information theory	**27**	**65.4%**
22		Run length coding, Variable length coding		
23		Dictionary based coding		
24		Arithmetic coding		
25		Lossless image compression		

26		Introduction to lossy compression algos.		
27		Distortion measures, Rate distortion theory		
28		Quantization, Transform coding		
29		Wavelet based coding		
30		Wavelet packets		
31		Embedded zero tree of wavelet coefficients		
32				
33		Set partitioning in hierarchical trees		
34				
35	**Unit #4 Image Compression Standards, Basic Video Compression Techniques** T1: 9.1-9.4, 10.1-10.5	JPEG standards	**17.1**	**82.5%**
36		JPEG 2000 standards		
37		JPEG-LS standard		
38		Bi-level image compression standards		
39		Intro to video compression		
40		Video compression based on motion compensation		
41		Search for motion vectors		
42		H.261		
43		H.263		
44	**Unit #5 MPEG Video Coding I – MPEG-1 and 2, MPEG Video Coding II – MPEG-4, 7 and Beyond.** T1: 11.1-11.3, 12.1-12.7	MPEG -1, MPEG - 2	**17.5**	**100%**
45		MPEG – 4		
46		Object based visual coding in MPEG 4		
47		Synthetic object coding in MPEG 4		
48		MPEG 4 object types		
49		Profiles and levels		
50		MPEG 4 part 10/H.264		
51		MPEG 7		
52		MPEG 21		

Literature:-

T1. Fundamentals of multimedia, by Ze-Nian Li and Mark S. Drew, 2004 Pearson Education Inc.

UE15CS321: COMPUTER GRAPHICS & VISUALIZATION(4-0-0-0-4)

# of Hrs: 52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Graphics Systems, Models, Implementation** **T1:** 1.1 to 1.9 **T2:** 3.1 to 3.4	Introduction: Applications of Computer Graphics	20	20
2.		A graphics system; Physical & Synthetic images		
3.		Imaging Systems; The synthetic camera model		
4.		Programmers Interface		
5.		Graphics Architectures		
6.		…contd.,		
7.		Programmable pipelines; Performance characteristics		
8.		Basic implementation strategies: Scan converting lines		
9.		Scan converting circles,		
10.		Scan converting ellipses		
11.	**Unit #2** **ClippingStrategies,Introduction to OpenGL** **T1: 2.3 to 2.7** **T2:** 3.5, 3.6, 3.11,3.14, 3.17	Basic Clipping Strategies: Line clipping	20	40
12.		Polygon clipping algorithm		
13.		Rectangle filling / Rasterization algorithm		
14.		Polygon filling / Rasterization algorithm		
15.		Hidden surface removal		
16.		Anti-aliasing techniques		
17.		Introduction to OpenGL		
18.		OpenGL API; primitives & attributes		
19.		Color; viewing		
20.		Control functions		
21.	**Unit #3** **Geometric objects and Transformations-1** **T1:** 4.1 to 4.7	Scalars, points, vectors	20	60
22.		…contd.,		
23.		Three dimensional primitives		
24.		Coordinate systems& Frames		
25.		Affine Transformations		
26.		Rotation		
27.		Translation		

28.		Scaling		
29.		problems		
30.		…contd.,		
31.	**Unit #4** **Geometric objects and Transformations-2** **T1:** 4.8 to 4.12	Geometrical objects and transformation	20	80
32.		Transformations in homogeneous coordinates		
33.		Homogeneous coordinate system		
34.		Problems		
35.		Concatenation of transformations		
36.		…contd.,		
37.		OpenGL transformations matrices		
38.		3D transformations		
39.		Interfaces to 3D applications		
40.		Quarternions		
41.	**Unit #5** **Viewing** **T1:** 5.1 to 5.5, 5.8, 5.9 **T2:**6.1, 6.2, 6.3	Classical Viewing	20	100
42.		Orthographic projections		
43.		Perspective viewing		
44.		Viewing with a Computer		
45.		Positioning of a camera		
46.		Viewing coordinate system		
47.		Simple projections		
48.		Projections in OpenGL		
49.		Introduction to parallel projection		
50.		Introduction toPerspective projection		
51.		Projection matrices		
52.		Revision		

Literature

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Textbook	T1	Interactive Computer Graphics, A top down approach using OpenGL by Edward Angel	5th	Pearson Education	2009
Textbook	T2	Computer Graphics : Principles & Practice	2nd	Pearson Education	2001
Textbook	T3	Interactive Computer Graphics, A top down approach using OpenGL WebGL	7th	Pearson Education	2016

Note: Pre-requisite for "Computer Graphics"(UE15CS321) is that students should have cleared "Design and Analysis of Algorithms" (UE15CS251)

Computer Science And Engineering
(Aug 17 – Dec 17)

UE15CS322: Data Analytics (4-0-0-0-4)

# of Hrs:52

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of syllabus	Cumulative %
1.	**Unit #1** **Exploratory Data Analysis and Visualization** T1: 7, 8, 9 T2: 2, 3, 4 T4: 2, 3, 6	Introduction to data analytics, data sources and representations	21	21
2.		The R programming environment		
3.		Exploring data - basic statistics		
4.		Data preprocessing - sampling, normalization, transformation		
5.		Dimensionality reduction		
6.		Other data preprocessing techniques		
7.		Data visualization – motivation, general concepts		
8.		Plotting data using R		
9.		Data visualization and R Graphics		
10.		Other relevant packages for visualization		
11.	**Unit #2** **Regression Analysis** T1: 4 T2: 7	Concept of training, validation and testing	18	39
12.		Distance and similarity measures		
13.		Correlation and simple regression		
14.		Case study		
15.		Multiple regression		
16.		Case study		
17.		Multivariate regression		
18.		Case study		
19.		Logistic regression		
20.		Case study		
21.	**Unit #3** **Recommender Systems** T3: 2, 3, 4	Introduction to recommender systems	23	62
22.		An overview of various types of recommender systems		
23.		Clustering at a glance		
24.		Case study		
25.		Classification at a glance		
26.		Case study		
27.		Concept of a k-itemset and frequent itemsets		
28.		Association rule mining		
29.		Evaluating associations		
30.		Review and case studies		
31.	**Unit #4** **Time Series Analysis**	Time series data, concept of stationarity and singularities		
32.		Trend analysis – simple smoothing (mean, median)		

33.	T1: 11	'Decomposing' a time series signal – into seasonal, trend and irregular components	17	79
34.		Trend analysis – exponential smoothing		
35.		The concept of 'filtering'		
36.		Different types of filters		
37.		Forecasting		
38.		Case study		
39.		ARIMA modeling		
40.		Case studies		
41.	**Unit #5** **Text Analytics** T1: 3 T4: 13	Text processing	21	100
42.		Case study		
43.		Lexical analysis		
44.		Case studies		
45.		Text clustering		
46.		Case study		
47.		Text classification		
48.		Case study		
49.		Analyzing different types of text		
50.		Case study		
51.		Review		
52.		Review		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Textbook	T1	R for Data Science, Dan Toomey	-	PACKT Publishing	2014
Textbook	T2	Practical Data Science with R, Nina Zumel and John Mount	-	Manning Publications	2014
Textbook	T3	Building a Recommendation System with R, Suresh R. Gorakala and Michelle Usuelli	-	PACKT Publishing	2015
Textbook	T4	Learning Predictive Analytics with R, Eric Mayor	-	PACKT Publishing	2015
Reference	R1	Data Analysis with Open Source Tools, Philip K. Janert	-		
Reference	R2	Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber and Jian Pei, The Morgan Kaufmann Series in Data Management Systems	3rd		

Note: Pre-requisite for "Data Analytics"(UE15CS322) is that students should have cleared "Introduction to Data Science" (UE15CS203)