

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Discrete Mathematical Structures
3.	L-T-P structure	3-1-0
4.	Credits	4
5.	Course number	CSL 202
6.	Status (<i>category for program</i>)	Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	
----	---	--

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Over 75% overlap with MAL180 (Discrete mathematical structures)
8.2	Overlap with any UG/PG course of other Dept./Centre	No
8.3	Supersedes any existing course	CSL 105

9.	Not allowed for (<i>indicate program names</i>)	
----	--	--

10.	Frequency of offering	Alternate semesters
-----	-----------------------	---------------------

11.	Faculty who will teach the course	Sandeep Sen, Amitabha Bagchi, Naveen Garg, Amit Kumar, Ragesh Jaiswal
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> This course is meant to be an introductory course on Discrete Mathematical Structures. The objective of this course is to teach the basic mathematical tools and techniques that are used in Computer Science.
------------	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Propositional logic, Predicate Calculus and Quantifiers; Proof Methods; Sets, functions, relations, Cardinality, Infinity and Diagonalization; Induction and Recursion; Modular Arithmetic, Euclid's Algorithm, primes, Public Key Cryptography; Polynomials, finite fields and Secret Sharing; Coding Theory: Error correcting codes, Hamming codes, Hamming bound; Basic Counting - Pigeon hole principle; Advanced Counting - recurrence relations, generating functions, inclusion-exclusion; basic information theory, entropy, Kraft's inequality, mutual information, lower bounds; Probability - sample space, conditional probability, expectation, linearity of expectation, variance, Markov, Chebychev, probabilistic methods; Graph Theory - Eulerian, Hamiltonian & planar graphs, edge and vertex coloring.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Propositional logic, Predicate Calculus and Quantifiers, Proof Methods	9
2	Sets, functions, relations, Cardinality, Infinity and Diagonalization	4
3	Induction and Recursion	3
4	Modular Arithmetic, Euclid's Algorithm, primes, Public Key Cryptography	5
5	Polynomials, finite fields and Secret Sharing	2
6	Coding Theory: Error correcting codes, Hamming codes, Hamming bound	3
7	Basic Counting - Pigeon hole principle	2
8	Advanced Counting - recurrence relations, generating	2

	functions, inclusion-exclusion	
9	Basic information theory, entropy, Kraft's inequality, mutual information, lower bounds	3
10	Probability - sample space, conditional probability, expectation, linearity of expectation, variance, Markov, Chebychev, probabilistic methods	5
11	Graph Theory - Eulerian, Hamiltonian & planar graphs, edge and vertex coloring	4
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

The tutorial sessions will essentially be problem sessions where students will solve conceptual problems based on the material being discussed in the lectures.

17.	Brief description of laboratory activities: NA
------------	---

18.	Suggested texts and reference materials
------------	--

1. Kenneth H Rosen, Discrete Mathematics and its Applications, Tata McGraw Hills, 2011.

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	N.A.
19.2	Hardware	N.A.
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.

19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	N.A.
20.2	Open-ended problems	N.A.
20.3	Project-type activity	N.A.
20.4	Open-ended laboratory work	N.A.
20.5	Others (please specify)	100% This is a theoretical course. The students will be working mainly on problems that help in understanding the concepts.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Digital logic & System design
3.	L-T-P structure	3-0-4
4.	Credits	5.0
5.	Course number	CSL 215
6.	Status (<i>category for program</i>)	Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL100, EEL100
----	--	-----------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	EE???
8.3	Supersedes any existing course	EEL201 & CSL316

9.	Not allowed for (<i>indicate program names</i>)	Nil
----	---	-----

10.	Frequency of offering	1 st sem
-----	------------------------------	---------------------

11.	Faculty who will teach the course	M. Balakrishnan, Anshul Kumar, Kolin Paul, SmrutiRanjanSarangi, PreetiRanjan Panda
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> Digital devices have become central to all human activities; whether it be personal communication and entertainment or industrial control. This course would introduce the student to the study of modern digital design
------------	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> The course contents can be broadly divided into two parts. First part deals with the basics of circuit design and includes topics like circuit minimization, sequential circuit design and design of and using RTL building blocks. The second part is focused on ASIC style system design and introduces VHDL, FPGA as implementation technology, synthesis steps as well as testing techniques. Course ends with introducing the challenges of embedded design where software is becoming integral to all devices. The laboratory assignments are a key component of this course and requires students to design and implement circuits and sub-systems on FPGA kits covering almost all the topics covered in the lectures.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Introduction to digital design – application domains, technology and scaling issues, design challenges including power consumption,	2
2	Building primitive components like gates, flip-flops and memory cells (primarily CMOS technology)	5
3	Combinational circuit representation and minimization	4
4	Sequential circuits and FSM design	6
5	Basic building blocks like adders, comparators, decoders, multiplexers, state machine controllers	4
6	Introduction to VHDL for modeling and synthesis	6
7	FPGA architectures and mapping	3
8	Testing digital circuits	3
9	RTL synthesis steps and design issues	3
10	ASIC design case studies	3

11	Challenges in embedded system design; Hardware-software co-design, interfacing analog sensors, performance and power issues	3
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Demonstration of using the FPGA kit and tools (with video backup)	4
2	A simple design for familiarization with the FPGA kit and simulation tools	4
3	Two assignments on combinational circuit design including its timing and resource utilization understanding	10
4	Two assignments on sequential circuit design including clock period computation	10
5	A simple case study involving sub-system design	8
5	Design project – specs to implementation with I/O interfacing	20
COURSE TOTAL (14 times 'P')		56

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Digital Design by M. Morris Mano, Michael D. Ciletti, Pearson
--

2. Digital Design: Principles and Practices Package (4th Edition), John F. Wakerly, Prentice Hall
3. Laboratory part: http://www.digilentinc.com/Data/Textbooks/Intro_Digital_Design-Digilent-VHDL_Online.pdf

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	VHDL simulation and synthesis tools, FPGA design software
19.2	Hardware	In the laboratory
19.3	Teaching aides (videos, etc.)	Videos to initiate laboratory work – demos on tool usage and kit usage as well as supplement class room with more case studies
19.4	Laboratory	Digital design laboratory
19.5	Equipment	FPGA kits, PCs for simulating, synthesizing and downloading FPGA bit-files, FPGA kit accessories for input-output interfacing as well as sensors, test & measurement equipment like function generators, oscilloscopes, logic analysers on need basis
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	60% (in the theory part)
20.2	Open-ended problems	25% (in the theory part)
20.3	Project-type activity	50% (in the laboratory part)
20.4	Open-ended laboratory work	33% (in the laboratory part – design project)
20.5	Others (please specify)	

Note: The course is heavily focused on design; whether it be classroom or the laboratory.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Computer Architecture
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 216
6.	Status (<i>category for program</i>)	Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	EEL 201 (Digital Logic) (recommended)
----	--	---------------------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	EEL 308 – Computer Architecture(60%)
8.3	Supersedes any existing course	CSL 211

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>2ndsem</u>
-----	------------------------------	--------------------------

11.	Faculty who will teach the course	Smruti R. Sarangi, Anshul Kumar, Preeti R. Panda, Kolin Paul, M. Balakrishnan
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> This course is meant to give undergraduate students a thorough overview of both software and hardware aspects of computer architecture. The software aspects include assembly language programming, and design of high level languages for both serial and parallel machines. The hardware aspect includes the organization of a pipelined and microprogrammed processor, along with a detailed presentation of the memory and I/O sub-systems.
------------	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> History of computers, Boolean logic and number systems, Assembly language programming, ARM assembly language, Computer arithmetic, Design of a basic processor, Microprogramming, Pipelining, Memory system, Virtual memory, I/O protocols and devices, Multiprocessors
------------	--

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Number systems	2
2	Assembly language	6
3	ARM assembly language	5
4	Computer arithmetic	4
5	Organization of a basic processor	4
6	Pipelining	4
7	Memory System – caches, virtual memory	5
8	I/O protocols	3
9	Storage devices – hard disks, RAID, Flash	3
10	Overview of parallel programming	3
11	Design of multiprocessors	3
COURSE TOTAL <i>(14 times 'L')</i>		42

--	--	--

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Design an assembly language emulator	6
2	Implement complicated computer arithmetic algorithms (such as Karatsuba multiplication, FFT, Schonhage-Strassen algorithm) on this emulator	6
3	Design a complete processor in Logisim or VHDL	10
4	Implement a pipelined processor in Logisim or in VHDL	6
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Henessey Patterson, Computer Organization and Design, 5th, Morgan Kaufmann, 2010

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	C/C++/Java, GNU HDL (open source), and Logisim(open source)
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.

19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	
20.2	Open-ended problems	10%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Design Practices
3.	L-T-P structure	0-0-6
4.	Credits	3
5.	Course number	CSP 290
6.	Status (<i>category for program</i>)	Under Graduate, DE

7.	Pre Requisites (<i>course no./title</i>)	CSL201 (Data Structures)
----	--	--------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	CSP301

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>1st sem</u>
-----	------------------------------	---------------------------

11.	Faculty who will teach the course	Aaditeshwar Seth, Huzur Saran, Subodh Kumar, Smruti Sarangi, Sorav Bansal, Kolin Paul, Preeti Ranjan Panda, Prem Kalra, Anshul Kumar, M. Balakrishnan, Parag Singla, almost all CSE faculty members
12.	Will the course require any visiting	No

	faculty?	
--	----------	--

13.	Course objective (<i>about 50 words</i>): To introduce the students to programming abstractions, and make them build fairly large software projects involving for example, GUI creation, distributed state maintenance over a network, programming in different environments like desktop and handhelds, program parsing and compilation, etc.
------------	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): The contents may differ each year depending on the instructor. The course should involve 2-3 large programming projects done in groups of 2-4.

15.	Lecture Outline (<i>with topics and number of lectures</i>)
------------	--

N.A.

16.	Brief description of tutorial activities
------------	---

#

	N.A.
--	------

#

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Project on building GUI, performing distributed synchronization over a network (e.g., game)	40
2	E.g., Project on parsing a programming language and writing a simulator to interpret it. Or building an architectural simulator, etc.	30
3	Any other project (or sub-part of any of the previous projects)	14
COURSE TOTAL (14 times 'P')		84

18.	Suggested texts and reference materials Programming books, as may be prescribed by the instructor. E.g. C / Java tutorials
------------	--

1. Project specific

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	GNU/Linux : primarily free and open source software
19.2	Hardware	Commodity desktop, laptop and server machines
19.3	Teaching aides (videos, etc.)	Course webpage
19.4	Laboratory	Enough desktop terminals with all required software installed on them.
19.5	Equipment	None
19.6	Classroom infrastructure	Data projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	
20.3	Project-type activity	70%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Embedded System Design Project
3.	L-T-P structure	0-1-6
4.	Credits	4
5.	Course number	CSP315
6.	Status (<i>category for program</i>)	Under Graduate, DE

7.	Pre Requisites (<i>course no./title</i>)	Normally in each group some students should have the done CSL215 and CSL216 or equivalent courses
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	CSP315

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>2ndsem</u>
-----	------------------------------	--------------------------

11.	Faculty who will teach the course	M Balakrishnan, --, --, --
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): Objective of the course is to provide a curricular platform for innovation and design in the embedded space. Projects would typically aim at finding innovative	
-----	---	--

	solutions including a concept demonstrator/prototyping for problems/issues which are very contextual to India and other developing countries.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Students working in small groups of four to six are expected to deliver in one semester on an innovative solution for problems/challenges that are typical to India and perhaps other developing countries. The students would have to go through the full cycle of specification, design and prototyping/building a concept demonstrator. Key component of the assessment would be through a public demonstration of their solution. Learning to work in groups as well as planning and delivering a large task are other expected learnings.
------------	--

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

16.	Brief description of tutorial activities
------------	---

The fourteen contact hours would be used for presenting periodically the progress through different stages of the project as well as for discussion of case studies by visitors. These visitor presentations from experienced innovators are meant to inspire, build confidence as well as provoke the student groups to think on all aspects of the solution.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Group project	84
COURSE TOTAL (14 times 'P')		84

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	--

1. Project specific

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	Project specific
19.2	Hardware	Project specific
19.3	Teaching aides (videos, etc.)	Nil
19.4	Laboratory	A well-equipped laboratory with controller kits, various sensors as well as hardware and software tools.
19.5	Equipment	Project specific
19.6	Classroom infrastructure	Data projector
19.7	Site visits	Project specific

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	
20.3	Project-type activity	100%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Operating Systems
3.	L-T-P structure	3-0-4
4.	Credits	5
5.	Course number	CSL 331
6.	Status (<i>category for program</i>)	Under Graduate, DE

7.	Pre Requisites (<i>course no./title</i>)	CSL201 (Data Structures), CSP290 (Design Practices)
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	EE/Maths OS?
8.3	Supersedes any existing course	CSL373

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>2ndsem</u>
-----	------------------------------	--------------------------

11.	Faculty who will teach the course	Subhashis Banerjee, Huzur Saran, SmrutiSarangi, Sorav Bansal, Kolin Paul
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>):	
-----	--	--

	To introduce the design and implementation of OS interfaces and understand the performance and modularity tradeoffs involved, provide experience with non-trivial event-driven and concurrent programming, study synchronization primitives and their uses, understand handling of disk, network and other devices.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Primary UNIX abstractions: threads, address spaces, filesystem, devices, interprocess communication; Introduction to hardware support for OS (e.g., discuss x86 architecture); Processes and Memory; Address Translation; Interrupts and Exceptions; Context Switching; Scheduling; Multiprocessors and Locking; Condition Variables, Semaphores, Barriers, Message Passing, etc.; Filesystem semantics, design and implementation; Filesystem Durability and Crash recovery; Security and Access Control

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Introduction to UNIX Abstractions	4
2	Hardware Support for OS Abstractions (e.g., x86 architecture)	4
3	Virtual to Physical Address Translation	3
4	Processes and Memory	3
5	Process Creation	1
6	Interrupts and Exceptions	2
7	Context-Switching	1
8	Process Scheduling	1
9	Multiprocessors and Locking	2
10	More Synchronization Primitives: Semaphores, Condition Variables, Barriers, etc.	4
11	Virtual Memory Swapping and Page Replacement	3
12	Files and Disk I/O	4

13	Crash Recovery and Logging	4
14	Lock-free Synchronization: CAS, RCU, etc.	3
15	Security and Access Control	2
16	Review	1
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

#

	N.A.
--	-------------

#

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Initializing PC Hardware : Device Handling, Stack, Interrupt/Exception Handling, Privilege Levels and Protected Control Flow Transfer.	12
2	Implementing User-level Processes by implementing system calls, command-line argument passing, address spaces, and protection mechanisms (e.g., based on Pintos user programs assignment).	14
3	Implementing Virtual Memory Mechanisms: Swapping to disk, memory-mapped files, copy-on-write optimizations, page replacement algorithms, stack growth (e.g., based on Pintos virtual memory assignment).	15
4	Implementing File System Mechanisms: directory structure, buffer cache, fine-grained filesystem synchronization, write behind and read ahead optimizations for the buffer cache (e.g., based on Pintos filesystem assignment)	15
COURSE TOTAL (14 times 'P')		56

18.	Suggested texts and reference materials
------------	--

	Cox, Kaashoek, and Morris. Xv6: a teaching operating system, book draft as of Aug. 28, 2012. Silberschatz, Galvin, and Gagne. Operating System Concepts (Eighth edition). John Wiley & Sons, Inc., 2008. ISBN 0-470-12872-0.
--	--

1. Project specific

19. Resources required for the course *(itemized & student access requirements, if any)*

19.1	Software	GNU/Linux, pintos, bochs, qemu : all free and open source software
19.2	Hardware	Commodity desktop, laptop and server machines
19.3	Teaching aides (videos, etc.)	Course webpage
19.4	Laboratory	Enough desktop terminals with all required software installed on them.
19.5	Equipment	None
19.6	Classroom infrastructure	Data projector
19.7	Site visits	N.A.

20. Design content of the course *(Percent of student time with examples, if possible)*

20.1	Design-type problems	30%
20.2	Open-ended problems	
20.3	Project-type activity	50%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Principles of Artificial Intelligence
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 333/CSL 671
6. Status (<i>category for program</i>)	Undergraduate/Postgraduate

7. Pre-requisites (<i>course no./title</i>)	CSL 201 (Data Structures)
---	---------------------------

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	50% Overlap with CSL 720 (Advanced Artificial Intelligence)
8.2 Overlap with any UG/PG course of other Dept./Centre	Partial Overlap with EE758
8.3 Supersedes any existing course	

9. Not allowed for (<i>indicate program names</i>)	
--	--

10. Frequency of offering	Every sem 1 st sem X 2 nd sem Either sem
----------------------------------	---

11. Faculty who will teach the course	Mausam, Saroj Kaushik, Parag Singla, KK Biswas
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course introduces the ideas in the field of artificial intelligence. It is tailored to students who have not had any exposure of the subject. Since AI is quite a broad field, this course gives an overview, both of traditional and modern AI, and prepares a student for advanced elective courses.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i>
	Philosophy of artificial intelligence, problem solving, search techniques, constraint satisfaction, game playing (minimax, expectiminimax), automated planning, knowledge representation and reasoning through logic, knowledge representation and reasoning through fuzzy logic and Bayesian networks, Markov decision processes, machine learning, neural networks, reinforcement learning, soft computing, introduction to natural language processing.

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Course Outline; Philosophy of Artificial Intelligence	3
2	Uninformed Search, Heuristic Search and Local Search	6
3	Constraint Satisfaction	3
4	Game Playing	3
5	Automated Classical Planning	3
6	Knowledge Representation: Logic and Satisfiability	4
7	Uncertainty in AI: Fuzzy Logic and Bayesian networks	6
8	Decision theory and Markov decision processes	2
9	Agents and Reinforcement learning	3
10	Machine learning and Neural networks	4
11	Soft computing	2
12	Intro to Natural language processing	2
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Modeling problems as search	5
2	Modeling problems as satisfiability	6
3	Implementing inference and learning for Bayesian nets	6
4	Modeling problems as an MDP	6
5	Use of Weka for basic machine learning	5
COURSE TOTAL (14 times 'P')		28

18. Suggested texts and reference materials
STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", Prentice-Hall, Third Edition (2009)
2. Saroj Kaushik, "Artificial Intelligence", Cengage Learning, 2011.
3. Tom Mitchell. "Machine Learning" First Edition, McGraw-Hill, 1997.
4. Online Reading Material.

19. Resources required for the course <i>(itemized & student access requirements, if any)</i>
--

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository, UCI datasets). Matlab may be used for couple of assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course <i>(Percent of student time with examples, if possible)</i>

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	10%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Analysis and Design of Algorithms
3.	L-T-P structure	3-1-0
4.	Credits	4
5.	Course number	CSL 351
6.	Status (<i>category for program</i>)	UnderGraduate

7.	Pre Requisites (<i>course no./title</i>)	CSL201 (Data Structures and Algorithms)
----	---	---

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	75% overlap with MAL342 (analysis and design of algorithms)
8.3	Supersedes any existing course	CSL 356

9.	Not allowed for (<i>indicate program names</i>)	
----	--	--

10.	Frequency of offering	Every sem <u>1st sem</u> 2 nd sem Either sem
-----	-----------------------	--

11.	Faculty who will teach the course	Naveen Garg, Amit Kumar, Ragesh Jaiswal, Sandeep Sen, Amitabha Bagchi
12.	Will the course require any visiting faculty?	No

13.	Course objective (about 50 words): This course is an introduction to various techniques for the design and analysis of efficient algorithms.
------------	--

14.	Course contents (about 100 words) (Include laboratory/design activities): checking 2-edge, 2-node and strong connectivity using DFS. Strongly connected components. Greedy algorithms, minimum spanning trees (Prim/Kruskal), Union-find data structure. Matroids. Divide and conquer algorithms. Polynomial multiplication, DFT and FFT. Dynamic Programming, All pairs shortest paths (Bellman-Ford, Floyd Warshall). s-t flows, ford-fulkerson, edmonds-karp, applications of maxflow Intractability, NP-completeness, Polynomial time reductions. String matching, KMP and Rabin-Karp. Universal hashing and applications. Geometric algorithms like convex hulls, multidimensional data structures, plane sweep paradigm
------------	--

15.	Lecture Outline (with topics and number of lectures)
------------	---

Module no.	Topic	No. of hours
1	2-edge, 2-node and strong connectivity using DFS. Strongly connected components	4
2	Greedy algorithms, minimum spanning trees (Prim/Kruskal), Union-find data structure. Matroids.	6
3	Divide and conquer algorithms. Polynomial multiplication, DFT and FFT,	6
4	Dynamic Programming, All pairs shortest paths (Bellman-Ford, Floyd Warshall)	6
5	s-t flows, ford-fulkerson, edmonds-karp, applications of maxflow	6
6	Intractability, NP-completeness, Polynomial time reductions	6
7	String matching, KMP and Rabin-Karp	2
8	Universal hashing and applications.	2
9	Geometric algorithms like convex hulls, multidimensional data structures, plane sweep paradigm	4
	COURSE TOTAL (14 times 'L')	42

16.	Brief description of tutorial activities: The tutorials are problem solving sessions where the tools and techniques taught in lectures are applied to a wide-variety of problems.
------------	--

17.	Brief description of laboratory activities: NA
-----	---

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
-----	---

1.	Kleinberg, John and Tardos, Eva. Algorithm design. Pearson
2.	Dasgupta S., Papadimitriou C.H., Vazirani U.V., Algorithms. McGraw-Hill.

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
-----	--

19.1	Software	NA
19.2	Hardware	NA
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)
-----	---

20.1	Design-type problems	
20.2	Open-ended problems	40%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Theory of Computation
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 352
6.	Status (<i>category for program</i>)	Under Graduate

7.	Pre Requisites (course no./title)	CSL202.
----	--------------------------------------	---------

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	No
8.2	Overlap with any UG/PG course of other Dept./Centre	75% overlap with MAL382 (theory of automata), CSL705
8.3	Supersedes any existing course	No

9.	Not allowed for (indicate program names)	
----	---	--

10.	Frequency of offering	Alternate semesters
-----	-----------------------	---------------------

11.	Faculty who will teach the course	Sandeep Sen, Naveen Garg, Amitabha Bagchi, Amit Kumar, Ragesh Jaiswal
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> This course will teach students the fundamentals of computation including different models their capabilities and their limitations.
------------	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Regular Languages, Finite Automata, equivalence, minimization, Myhill-Nerode Theorem, introduction to non-determinism, Context free grammars, Pushdown automata, equivalence and applications. Turing machines, Recursive and Recursively enumerable sets, non-determinism, RAMs and equivalence, Universal Turing Machines, undecidability, Rice's theorems for RE sets, Post machines, Basics of Recursive function theory. Equivalence, Church's thesis, computational complexity, space and time complexity of Turing Machines, Relationships, Savage's theorem, Complexity classes, Complete problems, NP-completeness, Cook-Levin theorem.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Languages, countable sets, finite represent, intro to diagonalization	3
2	Finite State machines: NFA, DFA. Regular expressions, equivalence, algorithmic conversions	6
3	Pumping lemma, closure properties, Myhill Nerode Thm, Minimization of FSM	6
4	CFG, Normal forms, PDA, equivalence, Pumping lemma, closure properties	6
5	Turing machine, recursive sets, re sets, undecidability, Many-one reduction, Rice's thm (recursive only).	9
6	Polynomial Space and Time	4
7	polynomial time reducibility, NP completeness, Cook-Levin thm	8
COURSE TOTAL		42

(14 times 'L')		

16.	Brief description of tutorial activities: NA
------------	---

17.	Brief description of laboratory activities: NA
------------	---

18.	Suggested texts and reference materials
------------	--

1.	John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education, 2012.
2.	Sipser, M, Introduction to theory of Computation, 3 rd Edition, Cengage learning

19.	Resources required for the course (itemized & student access requirements, if any)
------------	---

19.1	Software	N.A.
19.2	Hardware	N.A.
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course(Percent of student time with examples, if possible)
------------	---

20.1	Design-type problems	N.A.
20.2	Open-ended problems	N.A.
20.3	Project-type activity	N.A.

20.4	Open-ended laboratory work	N.A.
20.5	Others (please specify)	100% This is a theoretical course. The students will be working mainly on problems that help in understanding the concepts.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Introduction to Database Management Systems
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 362
6.	Status (<i>category for program</i>)	Post Graduate /Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL201 (Data Structures)
----	--	--------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	MAL710/Database Management Systems
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Every sem1 st sem2 nd sem Either sem
-----	------------------------------	--

11.	Faculty who will teach the course	S.K. Gupta, Maya Ramanath
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course will introduce the basics of Database Management Systems. While this
-----	--

	area is now extremely broad, the bulk of the focus will be on traditional and foundational data management techniques. Building on these basics, the course will then introduce students to recent topics, which they can pursue in advanced courses.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Data models (ER, relational models, constraints, normalization), declarative querying (relational algebra, datalog, SQL), query processing/optimization (basics of indexes, logical/physical query plans, views) and transaction management (introduction to concurrency control and recovery). Overview of XML data management, text management, distributed data management. Course project to build a web-based database application.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Data models	6
2	Declarative querying with relational algebra and SQL	3
3	Practical databases (constraints, views, triggers)	4
4	Index structures	3
5	Introduction to query processing/optimization	7
6	Declarative querying with datalog and recursive SQL	2
7	Transaction management	6
8	Introduction to XML data management	3
9	Introduction to Text data management	4
10	Introduction to Distributed data management	4
11		
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Data modeling	4
2	Learning to use Postgres	5
3	Learning to write SQL queries	9
4	Learning to read query plans	10
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", 2nd edition, Prentice Hall, 2008

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	Java, Perl/PHP, PostgreSQL
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector

19.7	Site visits	N.A.
------	-------------	------

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1	Department / Centre proposing the course	Computer Science and Engineering
2	Course Title (< 45 characters)	Introduction to Distributed and Parallel Programming
3	L-T-P structure	2-0-2
4	Credits	3
5	Course number	CSL 380
6	Status (category for program)	Undergraduate / Postgraduate

7.	Pre-requisites (course no./title)	CSL 201 (Data Structures), CSL 351 (Algorithms), CSL 331 (Operating Systems)
----	---	--

8	Status vis-à-vis other courses (give course number/title)	
8	Overlap with any UG/PG course of the Dept./Centre	CSL730
8	Overlap with any UG/PG course of other Dept./Centre	
8	Supersedes any existing course	

9.	Not allowed for (indicate program names)	
----	--	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem Either sem X
-----	------------------------------	---

11	Faculty who will teach the course	Subodh Kumar, Sandeep Sen
12	Will the course require any visiting faculty?	No

13	Course objective (about 50 words):	
3	This course introduces students to the concepts of parallel and distributed computing that entail logically simultaneous	

	execution of multiple threads of controls, whose operations interleave in complex ways.
--	---

14	Course contents (about 100 words) (Include laboratory/design activities): - Concurrency, Consistency of state and memory, Parallel architecture, Latency and throughput, Models of parallel computation, performance metrics and speedup, Message-passing and Shared-memory programming paradigms, Communication networks and primitives, Concepts of Atomicity, - Consensus, Conditions and Synchronization, Security, Fault tolerance, - Replication of state and memory
-----------	--

15.	Lecture Outline (with topics and number of lectures)
------------	---

Module no.	Topic	No. of hours
1	Overview: Introduction to concurrency, Race conditions, Atomicity, Semantics of concurrent programs, Examples of distributed algorithms, Client-Server paradigm	6
2	Parallel architecture, Flynn's classification	3
3	Shared-memory programming with reference to memory consistency, cache in-coherence, false sharing and mutual exclusion	3
3	Message passing, High level and collective constructs, Point-to-point communication, multicast and broadcast, Blocking versus non-blocking styles for communication, Message buffering	3
4	Theoretical models of parallel computation and algorithm analysis, Examples of reduction, prefix-sum	4
5	Performance metrics: Time, work, Scalability.	3
6	Task/Communication Dependence graphs, Task decomposition, Data-parallel decomposition, Pipelining	3
7	Synchronization, barriers, counters/semaphores, Progress, Livelock/Deadlock	3

COURSE TOTAL <i>(14 times 'L')</i>		28
--	--	-----------

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Example of generating and debugging race condition, and fixing them	6
2	OpenMP programming, Implement a shared data structure like distributed queue etc.	6
3	MPI based programming with emphasis on blocking messages, deadlocks and higher level communication primitives like multicast, scatter/gather etc. Graceful termination.	8
4	Many-core programming, Hybrid shared-memory/message passing programming	8
5		
COURSE TOTAL (14 times 'P')		28

18	Suggested texts and reference materials
.	STYLE: Author name and initials, Title, Edition, Publisher, Year.

1.	Joseph Jaja. An Introduction to Parallel Algorithms by (Addison-Wesley Professional), 1992.
2.	Ananth Grama, George Karypis, Vipin Kumar and Anshul Gupta. Introduction to Parallel Computing (Pearson), 2003.
3.	Michael J Quinn. Parallel Programming in C with MPI and openMP (McGraw Hill), 2003.
4.	Andrew S. Tanenbaum and Maarten van Steen. Distributed Systems: Principles and Paradigms (Prentice Hall), 2002.

19	Resources required for the course <i>(itemized & student access requirements, if any)</i>
.	

19.	Software	C/C++. OpenMP, MPI, CUDA
-----	----------	--------------------------

1		
1 9. 2	Hardware	Multiple computer (VM) access
1 9. 3	Teaching aides (videos, etc.)	No
1 9. 4	Laboratory	No
1 9. 5	Equipment	See Hardware
1 9. 6	Classroom infrastructure	Data Projector
1 9. 7	Site visits	No

2 0 .	Design content of the course <i>(Percent of student time with examples, if possible)</i>
----------------------	---

20.1	Design-type problems	50% of labs
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	20.00%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	B.Tech Project
3.	L-T-P structure	0-0-12
4.	Credits	6
5.	Course number	CSP 490
6.	Status (<i>category for program</i>)	Under Graduate, DC

7.	Pre Requisites (<i>course no./title</i>)	Completion of 120 Credits
----	--	---------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>Every Sem</u>
-----	------------------------------	------------------

11.	Faculty who will teach the course	All CSE faculty members
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is a Engineering Project for Students Not Opting for a Specialization. The students will work in a group of suitable size and Solve a Suitable Complex Engineering	
-----	--	--

	Problem. This will introduce the students to sophisticated programming abstractions, and make them build large software projects involving for example, GUI creation, distributed state maintenance over a network, programming in different environments like desktop and handhelds, program parsing and compilation, etc.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i>

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

N.A.

16.	Brief description of tutorial activities
------------	---

#

	N.A.
--	------

#

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
COURSE TOTAL <i>(14 times 'P')</i>		168

18.	Suggested texts and reference materials
	Material as may be prescribed by the Supervisor.

1. Project specific

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	
20.3	Project-type activity	70%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	B.Tech Project Part I
3.	L-T-P structure	0-0-12
4.	Credits	6
5.	Course number	CSP 492
6.	Status (<i>category for program</i>)	Under Graduate, DC

7.	Pre Requisites (<i>course no./title</i>)	Completion of 120 Credits
----	--	---------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>Every Sem</u>
-----	------------------------------	------------------

11.	Faculty who will teach the course	All CSE faculty members
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is a Thesis Project Part I for Students Opting for a Specialization.
-----	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i>

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

N.A.

16.	Brief description of tutorial activities
------------	---

#

	N.A.
--	------

#

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
COURSE TOTAL <i>(14 times 'P')</i>		168

18.	Suggested texts and reference materials Material as may be prescribed by the Supervisor.
------------	--

1. Project specific

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
------	----------------------	-----

20.2	Open-ended problems	
20.3	Project-type activity	70%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	B.Tech Project: Part II
3.	L-T-P structure	0-0-16
4.	Credits	8
5.	Course number	CSP 494
6.	Status (<i>category for program</i>)	Under Graduate, DC

7.	Pre Requisites (<i>course no./title</i>)	Completion of 120 Credits
----	--	---------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>Every Sem</u>
-----	------------------------------	------------------

11.	Faculty who will teach the course	All CSE faculty members
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is Part II of the Thesis Project for Students Opting for a Specialization. The students will work on suitable research/engineering problems in their area of	
-----	--	--

	specialization under the guidance of a Thesis Supervisor.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i>

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

N.A.

16.	Brief description of tutorial activities
------------	---

#

	N.A.
--	------

#

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
COURSE TOTAL <i>(14 times 'P')</i>		224

18.	Suggested texts and reference materials
	Material as may be prescribed by the Supervisor.

1. Project specific

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	
20.3	Project-type activity	70%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Compiler Optimization
3.	L-T-P structure	3-0-3
4.	Credits	4.5
5.	Course number	CSL 722
6.	Status (<i>category for program</i>)	DE (UG) & PE (Dual & PG)

7.	Pre Requisites (<i>course no./title</i>)	CSL 302 (Programming Languages)
----	--	---------------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Alternate Sem
-----	------------------------------	---------------

11.	Faculty who will teach the course	P. R. Panda, SoravBansal, Kolin Paul
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This objective of this course is to introduce the student to theoretical and practical aspects of optimization techniques employed in modern compilers.	
-----	---	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): Program representation – symbol table, abstract syntax tree; Control flow analysis; Data flow analysis; Static single assignment; Def-use and Use-def chains; Early optimisations – constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation; Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination; Register Allocation; Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining; Optimizing for memory hierarchy – code placement, scalar replacement of arrays, register pipelining; Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling; Function inlining and tail recursion; Dependence analysis; Just-in-time compilation; Garbage collection. Laboratory component would involve getting familiar with internal representations of compilers and the design and implementation of novel compiler optimizations.
------------	---

15.	Lecture Outline (<i>with topics and number of lectures</i>)
------------	--

Module no.	Topic	No. of hours
1	Program representation	3
2	Control flow analysis	2
3	Data flow analysis	4
4	Static single assignment	2
5	Early optimisations: constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation	4
6	Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination	3
7	Register Allocation	4
8	Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining	4
9	Optimizing for memory hierarchy – code placement, scalar replacement of arrays, register pipelining	4
10	Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling	4

11	Function inlining and tail recursion	2
12	Dependence analysis	2
13	Just-in-time compilation	2
14	Garbage collection.	2
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Getting familiar with internal representation	6
2	Design of a compiler optimisation algorithm	10
3	Software implementation of optimisation pass	20
4	Testing of implementation	6
COURSE TOTAL (14 times 'P')		42

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. S. S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufman, 1997
2. R. Allen and K. Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kaufman, 2001
3. J. Aycock, A brief history of just-in-time, ACM Computing Surveys, 35(2), 2003
4. M. Arnold, S. J. Fink, D. Grove, M. Hind, and P. F. Sweeney, A Survey of Adaptive Optimization in Virtual Machines, Proc. of the IEEE, 93(2), 2005

5. Paul R. Wilson, Uniprocessor Garbage Collection Techniques. Proc. Intl. Workshop on Memory Management (IWMM), Springer-Verlag, 1992

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	C/C++/Java compiler
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	White board, Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	30%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Department of Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	ADVANCED COMPUTER NETWORKS
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 724
6.	Status (<i>category for program</i>)	Programme Elective for CS Dual, MCS, CSY programmes Departmental Elective for B. Tech. CS / Dual CS.

7.	Pre-requisites (<i>course no./title</i>)	CSL374 Computer Networks or equivalent.
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	N/A
8.2	Overlap with any UG/PG course of another Dept./Centre	N/A
8.3	Supersedes any existing course	CSL858

9.	Not allowed for (<i>indicate program names</i>)	N/A
----	---	-----

10.	Frequency of offering	Every alternate semester
-----	------------------------------	--------------------------

11.	Faculty who will teach the course 1. Huzur Saran, 2. V. Ribeiro, 3. Aaditeshwar Seth	
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words; motivation and aims of this course</i>): The course will expose students to advanced concepts such as wireless MAC, BGP routing, MPLS, QOS scheduling and flow control, TCP variants, etc, with the objective of taking them into further details of computer networks. Assignments and projects will focus on teaching research methods such as simulations and performance evaluation.	
14.	Course contents (<i>about 100 words; this is a topics wise listing that will appear in the Courses of Study</i>) (<i>Include laboratory/design activities</i>): Review of the Internet architecture, layering; wired and wireless MAC; intra- and inter-domain Internet routing, BGP, MPLS, MANETs; error control and reliable delivery, ARQ, FEC, TCP; congestion and flow control; QoS, scheduling; mobility, mobile IP, TCP and MAC	

	interactions, session persistence; multicast; Internet topology, economic models of ISPs/CDNs/content providers; future directions
--	--

15.	Lecture Outline (<i>with topics and number of lectures</i>)	
Module no.	Topic	No. of hours
1	Internet architecture recap and principles	4
2	Error & Flow control; TCP	6
3	Routing (BGP, Policy Routing, MANET, Cost functions, Dynamics)	6
4	Network performance; flow control; scheduling & active queue mgmt	6
5	IP Mobility & IPv6	2
6	Wireless MAC, RLP and challenges	4
7	Internet topology and traffic characteristics – measurements & models:	4
8	MPLS & Traffic Engineering	2
9	Overlay Networks & Content Distribution	4
10	Future directions of the Internet architecture	4
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
	N/A

17.	Brief description of laboratory activities	
Module no.	Experiment description	No. of hours
1	Lab Exercise one – Bandwidth/latency measurements off distributed Internet nodes (PlanetLab)	4
2	Lab Exercise two – TCP performance in wireless on a local testbed	6
3	Lab Exercise three – Multicast Overlay Simulation with Mobility	8
4	Lab Project	10
5		
6		
7		
8		
COURSE TOTAL (14 times 'P')		

18.	Suggested texts and reference materials 1. Keshav, S. <i>An Engineering Approach to Computer Networks</i> 2. Shivkumar, <i>Network Architecture: Principles, Guidelines</i>, RPI 2006. 3. Peterson and Davie (book), <i>Computer Networks: A Systems Approach</i> 4. Relevant papers	
------------	--	--

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)	
19.1	Software	Network Simulation Software
19.2	Hardware	Lab computing resources and Routers (available)
19.3	Teaching aides (videos, etc.)	
19.4	Laboratory	Proxy quota to run Internet measurements
19.5	Equipment	

19.6	Classroom infrastructure	
19.7	Site visits	
20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)	
20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Numerical Algorithms
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 726
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 100
----	--	---------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	MAL 230 (80% overlap)
8.3	Supersedes any existing course	CSL 361

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	<u>Either semester</u>
-----	------------------------------	------------------------

11.	Faculty who will teach the course	SubhashisBannerjee, Amit Kumar
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is intended to give students tools for understanding sources of errors in computation and analyzing numerical stability of algorithms. They will also learn various	
-----	---	--

	tools in numerical linear algebra for solving problems arising in regression, estimation and optimization.
--	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Number representation, fundamentals of error analysis, conditioning, stability, polynomials and root finding, interpolation, singular value decomposition and its applications, QR factorization, condition number, least squares and regression, Gaussian elimination, eigenvalue computations and applications, iterative methods, linear programming, elements of convex optimization including steepest descent, conjugate gradient, Newton's method.
------------	--

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Number representation and rounding errors	2
2	Conditioning and stability	2
3	Polynomials and root finding	4
4	Interpolation	2
5	Matrix norms and singular value decomposition	4
6	QR factorization, Householder, Givens	4
7	Least squares, regression	4
8	Gaussian elimination and its numerical stability	2
9	Eigenvalues and applications	4
10	Power method and its convergence	2
11	Iterative methods	2
12	Linear programming and simplex algorithm	4
13	Convex programming and applications	2
14	Steepest descent, conjugate gradient	4
COURSE TOTAL		42

(14 times 'L')		

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Introduction to matlab and matrix computations	4
2	Singular Value Decomposition and applications	4
3	Least squares and regression	4
4	Applications of eigenvalues	6
5	Linear programming	4
6	Convex optimization	6
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Lloyd N. Trefethen and David Bau III, Numerical Linear Algebra, 1st edition, Society for Industrial and Applied Mathematics, 1997 2. Michael Heath, Scientific Computing: An introductory survey, 2nd edition, Tata McGraw-Hill Publishing Company, 2011. 3. Gene H. Golub, Charles van Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, 1996.
--

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	Matlab
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	Computers with matlab
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	10%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Compiler Design
3.	L-T-P structure	3-0-3
4.	Credits	4.5
5.	Course number	CSL 728
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 302 (Programming Languages)
----	--	---------------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	1 st sem
-----	------------------------------	---------------------

11.	Faculty who will teach the course	P. R. Panda, Sorav Bansal, Kolin Paul
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This objective of this course is to introduce the student to theoretical and practical aspects of optimization techniques employed in modern compilers.	
-----	---	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Program representation – symbol table, abstract syntax tree; Control flow analysis; Data flow analysis; Static single assignment; Def-use and Use-def chains; Early optimisations – constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation; Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination; Register Allocation; Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining; Optimizing for memory hierarchy – code placement, scalar replacement of arrays, register pipelining; Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling; Function inlining and tail recursion; Dependence analysis; Just-in-time compilation; Garbage collection. Laboratory component would involve getting familiar with internal representations of compilers and the design and implementation of novel compiler optimizations.
------------	--

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Program representation	3
2	Control flow analysis	2
3	Data flow analysis	4
4	Static single assignment	2
5	Early optimisations: constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation	4
6	Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination	3
7	Register Allocation	4
8	Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining	4
9	Optimizing for memory hierarchy – code placement, scalar replacement of arrays, register pipelining	4
10	Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling	4

11	Function inlining and tail recursion	2
12	Dependence analysis	2
13	Just-in-time compilation	2
14	Garbage collection.	2
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Getting familiar with internal representation	6
2	Design of a compiler optimisation algorithm	10
3	Software implementation of optimisation pass	20
4	Testing of implementation	6
COURSE TOTAL (14 times 'P')		42

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. S. S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufman, 1997
2. R. Allen and K. Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kaufman, 2001
3. J. Aycock, A brief history of just-in-time, ACM Computing Surveys, 35(2), 2003
4. M. Arnold, S. J. Fink, D. Grove, M. Hind, and P. F. Sweeney, A Survey of Adaptive Optimization in Virtual Machines, Proc. of the IEEE, 93(2), 2005

5. Paul R. Wilson, Uniprocessor Garbage Collection Techniques. Proc. Intl. Workshop on Memory Management (IWMM), Springer-Verlag, 1992

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	C/C++/Java compiler
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	White board, Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	30%
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	30%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Compiler Optimisation
3.	L-T-P structure	3-0-3
4.	Credits	4.5
5.	Course number	CSL 729
6.	Status (<i>category for program</i>)	DE (CS1, CS5) & PE (MCS, CSY, CSZ, ANZ, SIY)

7.	Pre Requisites (<i>course no./title</i>)	CSL 226 (Programming Languages) CSL 211 (Computer Architecture)
----	--	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Alternate Semesters
-----	------------------------------	---------------------

11.	Faculty who will teach the course	P. R. Panda, Sorav Bansal, Kolin Paul
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This objective of this course is to introduce the student to theoretical and practical aspects of optimisation techniques employed in modern compilers.	
-----	---	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): Program representation – symbol table, abstract syntax tree; Control flow analysis; Data flow analysis; Static single assignment; Def-use and Use-def chains; Early optimisations – constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation; Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination; Register Allocation; Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining; Optimizing for memory hierarchy – code placement, scalar replacement of arrays, register pipelining; Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling; Function inlining and tail recursion; Dependence analysis; Just-in-time compilation; Garbage collection. Laboratory component would involve getting familiar with internal representations of compilers; profiling and performance evaluation; and the design and implementation of novel compiler optimisations.
------------	--

15.	Lecture Outline (<i>with topics and number of lectures</i>)
------------	--

Module no.	Topic	No. of hours
1	Program representation	3
2	Control flow analysis	2
3	Data flow analysis	4
4	Static single assignment	2
5	Early optimisations: constant folding, algebraic simplifications, value numbering, copy propagation, constant propagation	4
6	Redundancy Elimination – dead code elimination, loop invariant code motion, common sub-expression elimination	3
7	Register Allocation	4
8	Scheduling – branch delay slot scheduling, list scheduling, trace scheduling, software pipelining	4
9	Optimising for memory hierarchy – code placement, scalar replacement of arrays, register pipelining	4
10	Loop transformations – loop fission, loop fusion, loop permutation, loop unrolling, loop tiling	4

11	Function inlining and tail recursion	2
12	Dependence analysis	2
13	Other topics: Just-in-time compilation, Garbage collection, Processor-specific optimisation	4
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Getting familiar with internal representation	6
2	Design of a compiler optimisation algorithm	10
3	Software implementation of optimisation pass	20
4	Testing of implementation, Profiling	6
COURSE TOTAL (14 times 'P')		42

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. S. S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufman, 1997
2. R. Allen and K. Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kaufman, 2001
3. J. Aycock, A brief history of just-in-time, ACM Computing Surveys, 35(2), 2003
4. M. Arnold, S. J. Fink, D. Grove, M. Hind, and P. F. Sweeney, A Survey of Adaptive Optimization in Virtual Machines, Proc. of the IEEE, 93(2), 2005

5. Paul R. Wilson, Uniprocessor Garbage Collection Techniques. Proc. Intl. Workshop on Memory Management (IWMM), Springer-Verlag, 1992

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	C/C++/Java compiler
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	White board, Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	30%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	DEPARTMENT OF COMPUTER SCIENCE & ENGG.
2.	Course Title (< 45 characters)	VIRTUALIZATION AND CLOUD COMPUTING
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL732
6.	Status (category for program)	DE for CS1 PE/DE for CS5 (Dual) PE for MCS/CSY/SIY/ANZ/CSZ
7.	Pre-requisites (course no./title)	CSL373

8.	Status vis-à-vis other courses (give course number/title)	
8.1	Overlap with any UG/PG course of the Dept./Centre	-
8.2	Overlap with any UG/PG course of another Dept./Centre	-
8.3	Supersedes any existing course	-

9.	Not allowed for (indicate program names)	
----	--	--

10.	Frequency of offering	EITHER SEMESTER
-----	------------------------------	-----------------

11.	Faculty who will teach the course 1. Dr Sorav Bansal 2. Prof. Huzur Saran	
12.	Will the course require any visiting faculty?	NO

13.	Course objective (about 50 words; motivation and aims of this course): The course introduces the students to state-of-the-art research in the fields of Machine Virtualization and Cloud Computing.	
-----	---	--

14.	Course contents (<i>about 100 words; this is a topics wise listing that will appear in the Courses of Study</i>) (<i>Include laboratory/design activities</i>): • Introduction to Virtualization and Cloud Computing • Binary Translation • Hardware Virtualization • Memory Resource Management in Virtual Machine Monitor • Application of Virtualization • Cloud-scale Data Management and Processing • I/O Virtualization
------------	--

15.	Lecture Outline (<i>with topics and number of lectures</i>)	
Module no.	Topic	No. of hours
1	Introduction, Binary Translation, Hardware, Virtualization	6
2	Para-Virtualization	2
3	Memory Management in Virtual Machine Monitor	4
4	I/O Virtualization	7
5	Application of Virtualization	6
6	Distributed File System	4
7	Large Cloud-scale Data Management	4
8	Cloud-scale Data Processing	4
9	Cloud Pricing	2
10	Nested Virtualization	3
11		
12		
COURSE TOTAL (<i>14 times 'L'</i>)		42

16.	Brief description of tutorial activities
	None

17.	Brief description of laboratory activities	
Modulen o.	Experiment description	No. of hours
1	VMM Benchmarking	9
2	Binary Translation and Shadow Page Tables	10
3	Hardware and I/O Virtualization	9
4		
5		
6		
7		
8		
9		
10		
COURSE TOTAL (<i>14 times 'P'</i>)		28

18.	Suggested texts and reference materials Research Papers (no suitable book). Some papers: 1. A Comparison of Software and Hardware Techniques for x86 Virtualization (Adams, Agesen, 2006) 2. Xen and the Art of Virtualization (Barham et. Al., 2003) 3. Memory Resource Management in VMware ESX Server (Waldspurger 2002) 4. Accelerating Two-Dimensional Page Walks for Virtualized Systems (Bhargava et. al., 2008) 5. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor (Sugerman et. al., 2001) 6. ELI: Bare-metal Performance for I/O virtualization (Gordon et. al., 2012) 7. The Google Filesystem (Ghemawat et. al., 2003) 8. Bigtable: A Distributed Storage System for Structured Data (Chang et. al., 2006) 9. MapReduce: Simplified Data Processing on Large Clusters (Dean et. al., 2004) 10. The Turtles Project: Design and Implementation of Nested Virtualization (Ben-Yehuda et. al., 2010) Other resources (for lab programming) 1. Linux/KVM (www.linux-kvm.org) 2. Binary translation based VMMs 3. Online cloud implementations like Amazon EC2, Google AppEngine, Microsoft Azure	
19.	Resources required for the course (itemized & student access requirements, if any)	
19.1	Software	KVM, Linux (all open source)
19.2	Hardware	X86-based computers
19.3	Teaching aides (videos, etc.)	
19.4	Laboratory	
19.5	Equipment	
19.6	Classroom infrastructure	
19.7	Site visits	
20.	Design content of the course (Percent of student time with examples, if possible)	
20.1	Design-type problems	20%
20.2	Open-ended problems	20%
20.3	Project-type activity	40%
20.4	Open-ended laboratory work	20%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Algorithmic Graph Theory
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 751
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 351 (Analysis and Design of Algorithms)
----	---	---

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	60% overlap with MAL656 (Graph Theory)
8.3	Supersedes any existing course	CSL851

9.	Not allowed for (<i>indicate program names</i>)	
----	--	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>
-----	-----------------------	---

11.	Faculty who will teach the course	Naveen Garg, Amit Kumar
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>):	
-----	---	--

	This course is meant to be a graduate level introduction to algorithmic techniques for graph theoretic problems. The emphasis would be on designing efficient algorithms and to identify graph parameters/classes for which even hard problems can be solved efficiently.
--	---

14.	Course contents (about 100 words) (Include laboratory/design activities): Algorithms for computing maximum s-t flows in graphs: augmenting path, blocking flow, push-relabel, capacity scaling etc. Max-flow min-cut theorem and its applications Algorithms for computing min-cuts in graphs, structure of min-cuts. Min-cost flows and applications: cycle cancelling algorithms, successive shortest paths, strongly polynomial algorithms. Maximum matching in bipartite and general graphs: Hall's theorem, Tutte's theorem, Gallai-Edmonds decomposition. Weighted bipartite matching, Edmonds Algorithm for Weighted Non-Bipartite Matching, T-joins and applications. Factor-Critical Graphs, Ear Decompositions, Graph orientations, Splitting Off, k-Connectivity Orientations and Augmentations, Arborescences and Branchings, Edmonds theorem for disjoint arborescences. Planar graphs, algorithms for checking planarity, planar-separator theorem and its applications. Intersection graphs, perfect graphs: polyhedral characterization, the strong perfect graph theorem, kinds of perfect graphs and algorithms on them. Treewidth, algorithm for computing tree width, algorithms on graphs with bounded tree width.
------------	--

15.	Lecture Outline (with topics and number of lectures)
------------	---

Module no.	Topic	No. of hours
1	Algorithms for computing maximum s-t flows in graphs: augmenting path, blocking flow, push-relabel, capacity scaling etc.	7
2	Max-flow min-cut theorem and its applications	3
3	Algorithms for computing min-cuts in graphs, structure of min-cuts	3
4	Min-cost flows and applications: cycle cancelling algorithms, successive shortest paths, strongly polynomial algorithms	4
5	Maximum matching in bipartite and general graphs: Hall's theorem, Tutte's theorem, Gallai-Edmonds decomposition.	3
6	Weighted bipartite matching, Edmonds Algorithm for Weighted Non-Bipartite Matching, T-joins and applications	4

7	Factor-Critical Graphs, Ear Decompositions, Graph orientations, Splitting Off, k-Connectivity Orientations and Augmentations, Arborescences and Branchings, Edmonds theorem for disjoint arborescences	5
8	Planar graphs, algorithms for checking planarity, planar-separator theorem and its applications.	3
9	Intersection graphs, perfect graphs: polyhedral characterization, the strong perfect graph theorem, kinds of perfect graphs and algorithms on them.	6
10	Treewidth, algorithm for computing tree width, algorithms on graphs with bounded tree width.	4
	COURSE TOTAL (14 times 'L')	42

16.	Brief description of tutorial activities: NA
-----	--

17.	Brief description of laboratory activities: NA
-----	--

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
-----	---

1.	Reinhard Diestel. Graph Theory. Springer-Verlag 2010
2.	T. Kloks. Advanced Graph Algorithms.
3.	Alexander Schrijver. A course in combinatorial Optimization. Springer 2012

19.	Resources required for the course (itemized & student access requirements, if any)
-----	---

19.1	Software	NA
19.2	Hardware	NA
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector

19.7	Site visits	N.A.
------	-------------	------

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering	
2.	Course Title (<i>< 45 characters</i>)	Geometric Algorithms	
3.	L-T-P structure	3-0-2	
4.	Credits	4	
5.	Course number	CSL 752	
6.	Status (<i>category for program</i>)	Post Graduate /Under Graduate	

7.	Pre Requisites (<i>course no./title</i>)	CSL 351 (Algorithm Design and Analysis) or equivalent	
----	--	---	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)		
8.1	Overlap with any UG/PG course of the Dept./Centre		
8.2	Overlap with any UG/PG course of other Dept./Centre		
8.3	Supersedes any existing course	CSL 852	

9.	Not allowed for (<i>indicate program names</i>)		
----	---	--	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>	
-----	-----------------------	---	--

11.	Faculty who will teach the course	Sandeep Sen, Amit Kumar, Ragesh Jaiswal, Amitabha Bagchi	
12.	Will the course require any visiting faculty?	No	

13.	Course objective (<i>about 50 words</i>): The course is directed at designing algorithms for problems that are geometric that have applications including CAD/CAM, robotics, computer graphics, molecular biology, GIS, spatial databases, sensor networks, and machine learning. The course will emphasize several specialized data representations and data structures for such problems as well as relevant combinatorial geometric techniques that helps us in analysing the behavior of such algorithms.		
-----	--	--	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): Geometric Fundamentals: Models of computation, lower bound techniques, geometric primitives, geometric transforms Convex hulls: Planar convex hulls, higher dimensional convex hulls, randomized, output-sensitive, and dynamic algorithms, applications of convex hull, Intersection detection: segment intersection, line sweep, map overlay, halfspace intersection, polyhedra intersection, Geometric searching: segment, interval, and priority-search trees, point location, persistent data structure, fractional cascading, range searching, nearest-neighbor searching Proximity problems: closest pair, Voronoi diagram, Delaunay triangulation and their subgraphs, spanners, well separated pair decomposition		
-----	---	--	--

	Arrangements: Arrangements of lines and hyperplanes, sweep-line and incremental algorithms, lower envelopes, levels, and zones, applications of arrangements Triangulations: monotone and simple polygon triangulations, point-set triangulations, optimization criteria, Steiner triangulation, Delaunay refinement Geometric sampling: random sampling and ϵ-nets, ϵ-approximation and discrepancy, cuttings, coresets Geometric optimization: linear programming, LP-type problems, parametric searching, approximation techniques. Implementation Issues : robust computing, perturbation techniques, floating-point filters, rounding techniques
--	---

15. Lecture Outline *(with topics and number of lectures)*

Module no.	Topic	No. of hours
1	Introduction and Visibility Problems	2
2	Plane sweep technique and applications	3
3	Convex hulls, Quickhull, Dynamic maintenance, Output sensitive algorithms	5
4	Dual Transformation, Intersection of half-planes	2
5	Lower bounds in Algebraic decision trees	2
6	Triangulation and Planar Point Location, Persistent Data Structures	5
7	Voronoi Diagrams, Delaunay Triangulation, Lifting Transform and Convex hulls	5
8	Randomized Incremental construction and random Sampling	4
9	Arrangements and Levels	3
10	Range Searching : Orthogonal and Simplex, Range Trees, Partition Trees	4
11	Epsilon nets and approximations, Quadrees and applications to Well-separated Pair decomposition, Geometric Optimization including fixed dimensional linear programming, Approximation Algorithms for ETSP, Set Cover	7
COURSE TOTAL (14 times 'L')		42

16. Brief description of tutorial activities

N.A.

17. Brief description of laboratory activities: Implementation of Convex hulls, Delaunay Triangulation, Range Trees, Robust implementation of geometric algorithms because of finite precision arithmetic, a short Project in groups of 3-4 students of a real life problem

Module no.	Experiment description	No. of hours
1	Planar convex hull	6
2	Voronoi diagrams/Delaunay Triangulation	6
3	K-d trees/Quadrees	6
4	A mini project (in groups)	10

COURSE TOTAL (14 times 'P')	28
------------------------------------	-----------

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	--

1.	Computational Geometry by M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, pub: Springer
2.	Computational Geometry in C, J. O' Rourke, Cambridge University Press
3.	Algorithms in Combinatorial Geometry, H. Edelsbrunner, pub: Springer-Verlag (EATCS Monograph)

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	C/C++/Java. Perl. Free Download. Matlab for doing assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Note: The course is designed to be a mix of theory as well as hands on experience. A background in Linear Algebra and Basic Probability is required.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Complexity Theory
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 753
6.	Status (<i>category for program</i>)	Post Graduate/Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL352 (Theory of Computation)
----	---	--------------------------------

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	No
8.2	Overlap with any UG/PG course of other Dept./Centre	No
8.3	Supersedes any existing course	CSL853

9.	Not allowed for (<i>indicate program names</i>)	
----	--	--

10.	Frequency of offering	Once every four semesters
-----	-----------------------	---------------------------

11.	Faculty who will teach the course	Ragesh Jaiswal, Sandeep Sen, Amitabha Bagchi, Naveen Garg, Amit Kumar
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is meant to be an introductory course on Computational Complexity Theory aimed
-----	---

	at advanced undergraduate and graduate students.
--	--

14.	Course contents (about 100 words) (Include laboratory/design activities): Modeling computation (Finite state machines, Non-determinism, Turing machines, class P etc.), NP and NP-completeness, Diagonalization (Time hierarchy and Ladner's theorem), Space complexity (PSPACE, NL, Savitch's theorem, Immerman-Szelepcsényi theorem etc.), Polynomial hierarchy, Boolean circuits (P/poly), Randomized classes (RP, BPP, ZPP, Adleman's Theorem, Gács-Sipser-Lautemann Theorem), Interactive proofs (Arthur-Merlin, IP=PSPACE), Cryptography (one-way functions, pseudorandom generators, zero knowledge), PCP theorem and hardness of approximation, Circuit lower bounds (Hastad's switching lemma), Other topics (#P, Toda's theorem, Average-case complexity, derandomization, pseudorandom construction).
------------	---

15.	Lecture Outline (with topics and number of lectures)
------------	---

Module no.	Topic	No. of hours
1	Modeling Computation	3
2	NP and NP-Completeness	3
3	Space Complexity	3
4	Polynomial Hierarchy	4
5	Boolean circuits	3
6	Randomized classes	4
7	Interactive proofs	4
8	Cryptography	3
9	PCP Theorem and Hardness of approximation	6
10	Other topics (counting classes, Average-case complexity, derandomization, pseudorandom constructions)	9
11		
COURSE TOTAL (14 times 'L')		42

--	--	--

16.	Brief description of tutorial activities: NA
-----	---

17.	Brief description of laboratory activities:NA
-----	--

18.	Suggested texts and reference materials
-----	--

1. Sanjeev Arora and Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009.
2. Oded Goldreich, Computational Complexity: A Conceptual Perspective, Cambridge University Press, 2008.

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
-----	--

19.1	Software	N.A.
19.2	Hardware	N.A.
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)
-----	---

20.1	Design-type problems	N.A.
20.2	Open-ended problems	N.A.
20.3	Project-type activity	N.A.

20.4	Open-ended laboratory work	N.A.
20.5	Others (please specify)	100% This is a theoretical course. The students will be working mainly on problems that help in understanding the concepts.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Mathematical Programming
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 756
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL351 (Analysis and Design of Algorithms)
----	---	--

8.	Status vis-à-vis other courses(<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	40% overlap with MAL210 (Optimization methods and applications), MAL638 (Applied non-linear programming) and MAL726 (Principles of Optimization theory). 60% overlap with MAL704 (Numerical Optimization)
8.3	Supersedes any existing course	CSL856

9.	Not allowed for (<i>indicate program names</i>)	
----	--	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>
-----	-----------------------	---

11.	Faculty who will teach the course	Naveen Garg, Amit Kumar
-----	-----------------------------------	-------------------------

12.	Will the course require any visiting faculty?	No
-----	---	----

13.	Course objective (about 50 words): This course is meant to be a graduate level introduction to linear, non-linear and convex optimization and integer programming. The emphasis will be on modeling optimization problems and on efficient algorithms for solving them.
-----	---

14.	Course contents (about 100 words) (Include laboratory/design activities): Linear programming: introduction, geometry, duality, sensitivity analysis. Simplex method, Large scale optimization, network simplex. Ellipsoid method, problems with exponentially many constraints, equivalence of optimization and separation. Convex sets and functions – cones, hyperplanes, norm balls, generalized inequalities and convexity, perspective and conjugate functions. Convex optimization problems – quasi-convex, linear, quadratic, geometric, vector, semi-definite. Duality – Lagrange, geometric interpretation, optimality conditions, sensitivity analysis. Applications to approximation, fitting, statistical estimation, classification. Unconstrained minimization, equality constrained minimization and interior point methods. Integer Programming: formulations, complexity, duality. Lattices, geometry, cutting plane and branch and bound methods. Mixed integer optimization
-----	--

15.	Lecture Outline (with topics and number of lectures)
-----	---

Module no.	Topic	No. of hours
1	Linear programming: introduction, geometry, duality, sensitivity analysis	6
2	Simplex method, Large scale optimization, network simplex	4
3	Ellipsoid method, problems with exponentially many constraints, equivalence of optimization and separation	4
4	Convex sets and functions – cones, hyperplanes, norm balls, generalized inequalities and convexity, perspective and conjugate functions,	4
5	Convex optimization problems – quasi-convex, linear, quadratic, geometric, vector, semi-definite. Duality – Lagrange, geometric interpretation, optimality conditions, sensitivity analysis	4
6	Applications to approximation, fitting, statistical estimation, classification	5

7	Numerical linear algebra background, unconstrained minimization, equality constrained minimization and interior point methods	7
8	Integer Programming: formulations, complexity, duality	3
9	Lattices, geometry, cutting plane and branch and bound methods	3
10	Mixed integer optimization	2
	COURSE TOTAL (14 times 'L')	42

16.	Brief description of tutorial activities: NA
-----	---

17.	Brief description of laboratory activities: NA
-----	---

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
-----	---

1.	Bertsimas, Dimitris, and Tsitsiklis, John. Introduction to Linear Optimization. Belmont, MA: Athena Scientific, 1997. ISBN: 9781886529199.
2.	Boyd, Stephen and Vandenberghe, Lieven. Convex Optimization. Cambridge University Press.
3.	Bertsimas, Dimitris, and Weismantel, Robert. Optimization over Integers. Dynamic Ideas 2005.

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
-----	--

19.1	Software	Matlab for doing assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Computer Science and Engineering	
2.	Course Title (<i>< 45 characters</i>)	Advanced Algorithms	
3.	L-T-P structure	3-0-2	
4.	Credits	4	
5.	Course number	CSL 758	
6.	Status (<i>category for program</i>)	Post Graduate/Under Graduate	

7.	Pre Requisites (<i>course no./title</i>)	CSL 351 (Algorithm Design and Analysis) or equivalent	
----	--	---	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)		
8.1	Overlap with any UG/PG course of the Dept./Centre		
8.2	Overlap with any UG/PG course of other Dept./Centre		
8.3	Supersedes any existing course		

9.	Not allowed for (<i>indicate program names</i>)		
----	---	--	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>	
-----	-----------------------	---	--

11.	Faculty who will teach the course	Sandeep Sen, Amit Kumar, Naveen Garg, SN Maheshwari	
12.	Will the course require any visiting faculty?	No	

13.	Course objective (<i>about 50 words</i>): The aim of the course is to make students familiar with contemporary algorithmic paradigms that are used to solve real life problems in frameworks not compatible with the Random Access Machine model, for example multicore models, streaming data, and caching issues. Since the design metrics and different, many new issues have to be addressed to design good algorithms in such computing environments.		
-----	---	--	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): The RAM model and its limitations, Introduction to alternate algorithmic models Parallel models like PRAM and Interconnection networks; Basic problems like Sorting, Merging, Routing, Parallel Prefix and applications, graph algorithms like BFS, Matching Memory hierarchy models ; Caching, Sorting, Merging, FFT, Permutation, Lower bounds Data Structures - searching, Priority queues Streaming Data models : Distinct items, frequent items, frequency moments,		
-----	--	--	--

	estimating norms, clustering
	On line algorithms: competitive ratio, list accessing, paging, k-server, load-balancing, lower-bounds

15. Lecture Outline <i>(with topics and number of lectures)</i>
--

Module no.	Topic	No. of hours
1	Idealized RAM model and its limitations	2
2	On line problems like list access, Competitive ratio	2
3	Paging :Deterministic strategy, lower bound, Randomized strategy	4
4	Load balancing, k-server problem	4
5	Parallel Computing models : shared memory, Interconnection networks	2
6	Sorting and selection in parallel models, Prefix Computation and applications	9
7	Matrix problems, BFS in graphs, Connectivity	4
8	Streaming Data model : read once, sublinear space sketch, Approximate expected bounds	3
9	Distinct elements, Frequent elements, random sampling, Frequency moment problem	6
10	Memory hierarchy models : Matrix Transpose, Sorting, FFT	4
11	Cache oblivious algorithms for searching, matrix transpose	2
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	--

N.A.

17.	Brief description of laboratory activities: Take up some classic prototype problem in each of the models and actually measure the performance of the specially Designed algorithms. This may be done individually or in small groups.
------------	---

Module no.	Experiment description	No. of hours
1	Group project on some suitable on-line problem	8
2	Group project on a suitable streaming data problem	8
3	Implementation of sorting in multicore machine using openMP or MPI	6
4	Comparing cache misses between a cache sensitive and RAM based algorithm for matrix transpose (using processor specific tools to measure L1, L2 cache misses	6

COURSE TOTAL (14 times 'P')	28
------------------------------------	-----------

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	--

1.	Joseph Ja Ja, An Introduction to Parallel Algorithms, Addison Wesley 1992.
2.	Borodin and El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.
3.	J.S Vitter, Lecture notes on External Memory Algorithms and Data Structures, Survey article in ACM Computing Survey, Volume 33 Issue 2, June 2001 Pages 209-271
4.	Muthukrushnan, Data Streams: Algorithms and Applications, Foundations and Trends® in Theoretical Computer Science

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	LEDA platform including the EM extension
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Note: The course is designed to be a mix of theory as well as hands on experience. A background in Basic Probability is required.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Cryptography and Computer Security
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 759
6.	Status (<i>category for program</i>)	Post Graduate / Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL351 or equivalent and MAL250 or equivalent
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	No
8.2	Overlap with any UG/PG course of other Dept./Centre	MAL724 (30%) MAL730 (50%) MAL786 (50%)
8.3	Supersedes any existing course	No

9.	Not allowed for (<i>indicate program names</i>)	N.A.
----	---	------

10.	Frequency of offering	Every Alternate semester
-----	------------------------------	--------------------------

11.	Faculty who will teach the course	Shweta Agrawal, Ragesh Jaiswal
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>):	
-----	--	--

	The course offers an introduction to modern methods of cryptography. Cryptography is a rigorous study of the science of secret keeping. The course initiates students in distilling real world security problems into mathematical models and reasoning about the security of real world systems and protocols using these models. Rigorous guarantees of security require that breaking a cryptosystem be provably as hard as solving some well known difficult mathematical problem, such as factoring. The course studies these methods via various practically motivated examples such as encryption systems, digital signatures, multiparty computation protocols etc.
--	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Part 1: Foundations: Perfect secrecy and its limitations, computational security, pseudorandom generators and one time encryption, pseudorandom functions, one way permutations, message authentication and cryptographic hash functions. Part 2: Basic Constructions and proofs: Some number theory, symmetric key encryption, public key encryption, CPA and CCA security, digital signatures, oblivious transfer, secure multiparty computation. Part 3: Advanced Topics: Zero knowledge proofs, identity based encryption, broadcast encryption, homomorphic encryption, lattice based cryptography.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Introduction – relevance of cryptography to real world security, principles of cryptography: defining security, concrete mathematical assumptions, reductions to assumptions. Case study of secure encryption : how to define security, understanding what assumptions are needed.	2
2	Perfect and Computational indistinguishability, pseudorandom generator, hybrid arguments, one way permutations, hardcore predicates, one way functions, Goldreich Levin predicate.	4
3	Stream Ciphers, Block Ciphers, Pseudorandom functions, PRF from PRG	3
4	Hash functions: universal, collision resistant,	3

	universal one way, MACs	
5	Public key Encryption: Definitions: IND and SIM based security, CPA and CCA security. Constructions: RSA, DDH, QR, Lattice CPA secure PKE from Trapdoor OWP Cramer Shoup CCA secure cryptosystem	8
6	Identity Based Encryption	3
7	Digital Signatures from OWF, from DDH, Lattices	3
8	Zero Knowledge: interactive/non-interactive, statistical/computational -- definitions and constructions	4
9	Protocols: oblivious transfer, bit commitment, coin flipping, secret sharing, BitCoins	4
10	Secure Multiparty computation : Yao's garbled circuits	3
11	Homomorphic encryption, functional encryption, broadcast encryption.	5
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1		
2		

3		
4		
COURSE TOTAL <i>(14 times 'P')</i>		

18.	Suggested texts and reference materials 1. Introduction to Modern Cryptography by Jonathan Katz and Yehuda Lindell. Chapman & Hall/CRC, 2007. 2. Lecture notes on Cryptography by Shafi Goldwasser and Mihir Bellare. Available on author's website: http://cseweb.ucsd.edu/users/mihir/papers/gb.html
------------	--

1.	
-----------	--

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	N.A.
19.2	Hardware	Personal laptop or Computer access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	N.A.
20.2	Open-ended problems	N.A.
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	N.A.
20.5	Others (please specify)	80%.

		This is primarily a theoretical course. The students will be working mainly on problems that help in understanding the concepts.
--	--	--

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Advanced Data Management
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 760
6.	Status (<i>category for program</i>)	Post Graduate / Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 362 (Introduction to Database Systems)
----	--	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Every sem1 st sem2 nd sem <u>Either sem</u>
-----	------------------------------	---

11.	Faculty who will teach the course	Maya Ramanath
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>):	The course will introduce advanced data management techniques with the main focus being on the fundamentals of distributed data management, and the management of massive amounts of data (or “Big Data”).
-----	--	--

--	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Storage and file structures, advanced query processing and optimization for single server databases, distributed data management (including distributed data storage, query processing and transaction management), web-data management (including managing the web-graph and implementation of web-search), big data systems.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Database Architectures (client-server, distributed/parallel, cloud)	3
2	Storage and file structures	4
3	Advanced query processing and optimization	4
4	Distributed Data Storage	2
5	Distributed Query Processing and Optimization	8
6	Distributed Transaction Management	4
7	Web Data Management (Managing the Web Graph, Web Search)	6
8	Big Data Systems and data processing (Introduction and Motivation, Mapreduce, Pregel, Hive, Pig Latin, etc.)	11
9		
10		
11		
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Query Optimization for Complex Queries	4
2	Learning about the fundamentals of distributed query processing	4
3	Implementation of Web-search algorithms	10
4	Programming with big-data systems	10
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

• Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", 2nd edition, Prentice Hall, 2008 • Tamer Oszu, Patrick Valduriez, "Principles of Distributed Database Systems", 3rd edition, Springer, 2010 • Journal and conference papers

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	Java. Perl/PHP. PostgreSQL. Webserver, Hadoop/Giraph installations, Galago/Lucene installation
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.

19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Database Implementation
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 762
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 362 (Introduction to Database Systems)
----	--	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	Nil
8.2	Overlap with any UG/PG course of other Dept./Centre	Nil
8.3	Supersedes any existing course	CSL771

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Every sem1 st sem2 nd sem <u>Either sem</u>
-----	------------------------------	---

11.	Faculty who will teach the course	S. K. Gupta, Maya Ramanath
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>):The course addresses the issue of database implementation, query processing and query optimization for relational databases. Different file structures, indexing and hashing techniques and query processing will be
-----	--

	addressed.
--	------------

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Review of Relational Model, Algebra and SQL, File structures, Constraints and Triggers, System Aspects of SQL, Data Storage, Representing Data Elements, Index, Multi dimensional and Bit-map Indexes, Hashing, Query Execution, Query Compiler.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Review of Relational Model, Algebra and SQL	3
2	Constraints and Triggers	3
3	File Structures	6
4	Data Storage	4
5	Representing Data Elements	4
6	Index, Multi dimensional and Bit-map Indexes	10
7	Query Execution	6
8	Query Compiler	6
9		
10		
11		
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	SQL Query Parsing and Compilation	6
2	Transformation of SQL into relational algebra	6
3	Different Algorithm for Relational Algebra Operation	8
4	Query Processing and Optimization	8
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", 2nd edition, Prentice Hall, 2008
- Tamer Oszu, Patrick Valduriez, "Principles of Distributed Database Systems", 3rd edition, Springer, 2010
- Journal and conference papers

19.	Resources required for the course (<i>itemized & student access requirements, if any</i>)
------------	--

19.1	Software	Java. Perl/PHP. PostgreSQL. Webserver, mySQL
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course (<i>Percent of student time with examples, if possible</i>)
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Advanced Artificial Intelligence
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 770
6. Status (<i>category for program</i>)	Postgraduate

7. Pre-requisites (<i>course no./title</i>)	CSL 201 (Data Structures) or CSL 630 (Data Structures and Algorithms)
---	---

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	50% Overlap with CSL 333/CSL 671 (Fundamentals of Artificial Intelligence)
8.2 Overlap with any UG/PG course of other Dept./Centre	Partial Overlap with EE 758.
8.3 Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)
-----------	---

10. Frequency of offering	Every sem 1 st sem 2 nd sem X Either sem
----------------------------------	--

11. Faculty who will teach the course	Mausam, Saroj Kaushik, Parag Singla, KK Biswas
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course introduces several advanced ideas in the field of artificial intelligence. It is tailored to students who have already had some exposure of the subject, either as an undergraduate AI course or an online course. Since AI is quite a broad field, this course will take an overview approach, both of traditional and modern AI. Since the students would have had basic exposure of AI, the course will move fast and cover a lot more material than CSL 333.
--	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i>
	Philosophy of artificial intelligence, fundamental and advanced search techniques (A*, local search, suboptimal heuristic search, search in AND/OR graphs), constraint optimization, temporal reasoning, knowledge representation and reasoning through propositional and first order logic, modern game playing (UCT), planning under uncertainty (Topological value iteration, LAO*, LRTDP), reinforcement learning, introduction to robotics, introduction to probabilistic graphical models (Bayesian networks, Hidden Markov models, Conditional random fields), machine learning, introduction to information systems (information retrieval, information extraction).

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Course Outline; Philosophy of Machine Intelligence	3
2	Uninformed, Heuristic and Local Search	3
3	Search in AND/OR Graphs	2
4	Constraint Optimization	3
5	Temporal Reasoning	2
6	Knowledge Representation: Propositional Logic	3
7	First order logic and Logic programming	5
8	Game Playing and UCT algorithm	3
9	Planning under Uncertainty	3
10	Reinforcement learning	3
11	Probabilistic Graphical models (directed & undirected)	5
12	Machine Learning and Neural Networks	3
13	Introduction to Robotics	2
14	Introduction to Information Systems	2
COURSE TOTAL <i>(14 times 'L')</i>		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Modeling problems as AND/OR search	5
2	Modeling problems using Prolog	6
3	Using UCT to solve large games	5
4	Solving problems using planning under uncertainty algorithms	6
5	Modeling problems using conditional random fields	6

COURSE TOTAL (14 times 'P')	28
------------------------------------	-----------

18. Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", Prentice-Hall, Third Edition (2009)
2. Saroj Kaushik, "Artificial Intelligence", Cengage Learning, 2011.
3. Tom Mitchell. "Machine Learning" First Edition, McGraw-Hill, 1997.
4. Online Reading Material.

19. Resources required for the course (<i>itemized & student access requirements, if any</i>)
--

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository, UCI datasets). Matlab may be used for couple of assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course (<i>Percent of student time with examples, if possible</i>)

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	10%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Advanced Artificial Intelligence
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 770
6. Status (<i>category for program</i>)	Postgraduate

7. Pre-requisites (<i>course no./title</i>)	CSL 201 (Data Structures) or CSL 630 (Data Structures and Algorithms)
---	---

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	50% Overlap with CSL 333/CSL 671 (Fundamentals of Artificial Intelligence)
8.2 Overlap with any UG/PG course of other Dept./Centre	Partial Overlap with EE 758.
8.3 Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)
-----------	---

10. Frequency of offering	Every sem 1 st sem 2 nd sem X Either sem
----------------------------------	--

11. Faculty who will teach the course	Mausam, Saroj Kaushik, Parag Singla, KK Biswas
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course introduces several advanced ideas in the field of artificial intelligence. It is tailored to students who have already had some exposure of the subject, either as an undergraduate AI course or an online course. Since AI is quite a broad field, this course will take an overview approach, both of traditional and modern AI. Since the students would have had basic exposure of AI, the course will move fast and cover a lot more material than CSL 333.
--	--

14. Course contents (about 100 words) (Include laboratory/design activities):

Philosophy of artificial intelligence, fundamental and advanced search techniques (A*, local search, suboptimal heuristic search, search in AND/OR graphs), constraint satisfaction and optimization, temporal reasoning, knowledge representation and reasoning through propositional and first order logic, modern game playing (UCT), automated planning under uncertainty, reinforcement learning, introduction to robotics, introduction to probabilistic graphical models (Bayesian networks, Hidden Markov models, Conditional random fields), machine learning, application of machine learning to NLP problems, introduction to information systems (information retrieval, information extraction).

15. Lecture Outline (with topics and number of lectures)

Module no.	Topic	No. of hours
1	Course Outline; Philosophy of Machine Intelligence	3
2	Uninformed, Heuristic and Local Search	3
3	Search in AND/OR Graphs	2
4	Constraint Satisfaction and Optimization	3
5	Temporal Reasoning	2
6	Knowledge Representation: Propositional Logic	3
7	First order logic and Logic programming	5
8	Game Playing and UCT algorithm	3
9	Planning under Uncertainty	3
10	Reinforcement learning	3
11	Probabilistic Graphical models	5
12	Machine Learning and Neural Networks	3
13	Introduction to Natural Language Processing	2
14	Introduction to Information Systems	2
COURSE TOTAL (14 times 'L')		42

16. Brief description of tutorial activities

N.A.

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
1	Modeling problems as AND/OR search	5
2	Modeling problems using Prolog	6
3	Using UCT to solve large games	5
4	Solving problems using planning under uncertainty algorithms	6
5	Modeling problems using conditional random fields	6

COURSE TOTAL (14 times 'P')	28
------------------------------------	-----------

18. Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", Prentice-Hall, Third Edition (2009)
2. Saroj Kaushik, "Artificial Intelligence", Cengage Learning, 2011.
3. Tom Mitchell. "Machine Learning" First Edition, McGraw-Hill, 1997.
4. Online Reading Material.

19. Resources required for the course (<i>itemized & student access requirements, if any</i>)
--

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository, UCI datasets). Matlab may be used for couple of assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course (<i>Percent of student time with examples, if possible</i>)

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	10%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Natural Language Processing
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 772
6. Status (<i>category for program</i>)	Postgraduate

7. Pre-requisites (<i>course no./title</i>)	CSL 201 (Data Structures) or CSL 630 (Data Structures and Algorithms)
---	---

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	
8.2 Overlap with any UG/PG course of other Dept./Centre	
8.3 Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)
-----------	---

10. Frequency of offering	Every sem 1 st sem 2 nd sem X Either sem
----------------------------------	--

11. Faculty who will teach the course	Mausam, Saroj Kaushik, Maya Ramanath
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course teaches various basic and advanced ideas in the field of natural language processing. The students will learn fundamental algorithms for processing natural languages such as English. The course will provide all requisite background in machine learning and probabilistic graphical models before describing their applications to NLP. The students will also learn how these techniques can be used to solve various practical problems such as information search, question answering, document summarization, etc.
--	--

14. Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>):
--

	NLP concepts: Tokenization, lemmatization, part of speech tagging, noun phrase chunking, named entity recognition, coreference resolution, parsing, information extraction, sentiment analysis, question answering, text classification, document clustering, document summarization, discourse, machine translation. Machine learning concepts: Naïve Bayes, Hidden Markov Models, EM, Conditional Random Fields, MaxEnt Classifiers, Probabilistic Context Free Grammars.
--	---

15. Lecture Outline <i>(with topics and number of lectures)</i>
--

Module no.	Topic	No. of hours
1	Course Outline	1
2	Background: Naïve Bayes, Perceptron, Linear models	3
3	Text Classification	2
4	Language Models	2
5	Tokenization and Lemmatization	2
6	Background: HMMs and CRFs	4
7	NLP pipeline: POS, NP chunking, NER, Coreference	5
8	Background: PCFGs	2
9	Syntactic and Dependency Parsing	3
10	Information Extraction	3
11	Sentiment Analysis	2
12	Information Retrieval	2
13	Background: Latent Dirichlet Allocation	2
14	Document Clustering	2
15	Machine Translation	2
16	Question Answering	2
17	Document Summarization	2
18	Wrap up	1
COURSE TOTAL <i>(14 times 'L')</i>		42

16. Brief description of tutorial activities

N.A.

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
1	Classifying a document into known classes	5
2	Implementing a part of speech tagger	5
3	Implementing a relation extractor	5
4	Project	13

COURSE TOTAL (14 times 'P')	28
------------------------------------	-----------

18. Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. D. Jurafsky & James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, Second Edition, 2009.
2. C.D. Manning & H. Schuetze, Foundations of Statistical Natural Language Processing, Cambridge: MIT Press, 1999.
3. James F. Allen: Natural language understanding (2nd ed.). Benjamin/Cummings 1994

19. Resources required for the course (<i>itemized & student access requirements, if any</i>)
--

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository).
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course (<i>Percent of student time with examples, if possible</i>)

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	10%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Natural Language Processing
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 772
6. Status (<i>category for program</i>)	Postgraduate

7. Pre-requisites (<i>course no./title</i>)	CSL 201 (Data Structures) or CSL 630 (Data Structures and Algorithms)
---	---

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	
8.2 Overlap with any UG/PG course of other Dept./Centre	
8.3 Supersedes any existing course	

9. Not allowed for (<i>indicate program names</i>)	
--	--

10. Frequency of offering	Every sem 1 st sem 2 nd sem X Either sem
----------------------------------	--

11. Faculty who will teach the course	Mausam
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course teaches various basic and advanced ideas in the field of natural language processing. The students will learn fundamental algorithms for processing natural languages such as English. The course will provide all requisite background in machine learning and probabilistic graphical models before describing their applications to NLP. The students will also learn how these techniques can be used to solve various practical problems such as information search, question answering, document summarization, etc.
--	--

14. Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>):	NLP concepts: Tokenization, lemmatization, part of speech tagging, noun phrase
--	--

	chunking, named entity recognition, coreference resolution, parsing, information extraction, sentiment analysis, question answering, text classification, document clustering, document summarization, discourse, machine translation.
	Machine learning concepts: Naïve Bayes, Hidden Markov Models, EM, Conditional Random Fields, MaxEnt Classifiers, Probabilistic Context Free Grammars.

15. Lecture Outline <i>(with topics and number of lectures)</i>
--

Module no.	Topic	No. of hours
1	Course Outline	1
2	Background: Naïve Bayes, Perceptron, Linear models	3
3	Text Classification	2
4	Language Models	2
5	Tokenization and Lemmatization	2
6	Background: HMMs and CRFs	4
7	NLP pipeline: POS, NP chunking, NER, Coreference	5
8	Background: PCFGs	2
9	Syntactic and Dependency Parsing	3
10	Information Extraction	3
11	Sentiment Analysis	2
12	Information Retrieval	2
13	Background: Latent Dirichlet Allocation	2
14	Document Clustering	2
15	Machine Translation	2
16	Question Answering	2
17	Document Summarization	2
18	Wrap up	1
COURSE TOTAL (14 times 'L')		42

16. Brief description of tutorial activities

N.A.

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
1	Classifying a document into known classes	5
2	Implementing a part of speech tagger	5
3	Implementing a relation extractor	5
4	Project	13
COURSE TOTAL (14 times 'P')		28

18. Suggested texts and reference materials

STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. D. Jurafsky & James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, Second Edition, 2009.

2. C.D. Manning & H. Schuetze, Foundations of Statistical Natural Language Processing, Cambridge: MIT Press, 1999

19. Resources required for the course (*itemized & student access requirements, if any*)

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository).
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course (*Percent of student time with examples, if possible*)

20.1	Design-type problems	
20.2	Open-ended problems	
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	10%
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Machine Learning
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 774
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	MAL 250 (Introduction to Probability Theory and Stochastic Processes)
----	--	---

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	60% Overlap with CSL 341 (Fundamentals of Machine Learning)
8.2	Overlap with any UG/PG course of other Dept./Centre	60% Overlap with EE 709 (Pattern Recognition). 40% overlap with MAL 803 (Pattern Recognition).
8.3	Supersedes any existing course	

9.	Not allowed for (<i>Indicate program names</i>)	Students who have registered for EE 709, MAL 803, CSL 341
----	---	---

10.	Frequency of offering	Every sem1 st sem2 nd sem <u>Either sem</u>
-----	------------------------------	---

11.	Faculty who will teach the course	K KBiswas, Subhashis Banerjee, ParagSingla, Mausam, ManikVarma
-----	--	---

12.	Will the course require any visiting faculty?	No
------------	--	-----------

13.	Course objective <i>(about 50 words):</i> This course is meant to be the first graduate level course in Machine Learning. The field of Machine learning deals with building predictive models of data based on past experience. The subject has become particularly popular in the last decade or so with a wide range of applications. This course will focus on providing students with an in depth understanding of the field including choice of learning models, a variety of learning algorithms, relative merits of various algorithms and examples of practical applications. The course will also provide the required background for carrying out further research in the area.
------------	---

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Supervised learning algorithms: Linear and Logistic Regression, Gradient Descent, Support Vector Machines, Kernels, Artificial Neural Networks, Decision Trees, ML and MAP Estimates, K-Nearest Neighbor, Naive Bayes, Introduction to Bayesian Networks. Unsupervised learning algorithms: K-Means clustering, Gaussian Mixture Models, Learning with Partially Observable Data (EM). Dimensionality Reduction and Principal Component Analysis. Bias Variance Trade-off. Model Selection and Feature Selection. Regularization. Learning Theory. Introduction to Markov Decision Processes. Application to Information Retrieval, NLP, Biology and Computer Vision. Advanced Topics.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Course Outline; Basics	2
2	Linear Regression, Gradient Descent, Logistic Regression	6
3	Linear Support Vector Machines	5
4	Kernels, Non-Linear SVMs	4
5	Artificial Neural Networks, Decision Trees	4
6	Naive Bayes, Introduction to Bayesian Networks	4
7	K-Means, Gaussian Mixture Models, EM	5

8	Feature Selection, Model Selection, Regularization, Bias and Variance	3
9	Learning Theory	3
10	Markov Decision Processes	3
11	Applications and Advanced Topics	3
COURSE TOTAL (14 times 'L')		42

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities
------------	---

Module no.	Experiment description	No. of hours
1	Implementation of Linear Regression, Gradient Descent (Batch and Stochastic), SVMs	6
2	Naive Bayes for Text Classification, Decision Trees, Playing with WEKA datasets	6
3	K-Means, PCA for dimensionality Reduction	6
4	Course Project	10
COURSE TOTAL (14 times 'P')		28

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Kevin P. Murphy. "Machine Learning: A Probabilistic Perspective". MIT Press, 2012.
2. Christopher Bishop. "Pattern Recognition and Machine Learning", First Edition, Springer, 2006.
3. R. Duda, P. Hart and D. Stork. "Pattern Classification" Second Edition, John Wiley & Sons

INC., 2001.
4. Tom Mitchell. "Machine Learning" First Edition, McGraw-Hill, 1997.
5. Online Reading Material.

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	C/C++/Java. Perl. Free Download (e.g. Weka machine learning repository, UCI datasets). Matlab for doing assignments.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the Course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Note: The course is designed to be a mix of theory as well as hands on experience. A background in Linear Algebra and Basic Probability is required.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1. Department/Centre proposing the course	Computer Science and Engineering
2. Course Title (<i>< 45 characters</i>)	Learning Probabilistic Graphical Models
3. L-T-P structure	3-0-2
4. Credits	4
5. Course number	CSL 776
6. Status (<i>category for program</i>)	Postgraduate

7. Pre-requisites (<i>course no./title</i>)	MAL 250 (Introduction to Probability Theory and Stochastic Processes)
---	---

8. Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1 Overlap with any UG/PG course of the Dept./Centre	-
8.2 Overlap with any UG/PG course of other Dept./Centre	-
8.3 Supersedes any existing course	-

9.	Not allowed for (<i>Indicate program names</i>)
-----------	---

10. Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>
----------------------------------	---

11. Faculty who will teach the course	Parag Singla, Mausam, KK Biswas
12. Will the course require any visiting faculty?	No

13. Course objective (<i>about 50 words</i>):	This course is meant to be the first graduate level course in the area of Probabilistic Graphical Models (PGM). PGMs have emerged as a very important research field during last decade or so with wide range of applications including Computer Vision, Information Retrieval, Natural Language Processing, Biology and Robotics. This course aims to provide students with a comprehensive overview of PGMs. The course content will include introduction to Probabilistic Graphical Models, directed and undirected representations, inference and learning algorithms and practical applications. This course is also meant to provide the required background for pursuing research in this area.
--	--

14. Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>):	Basics: Introduction. Undirected and Directed Graphical Models. Bayesian Networks. Markov Networks. Exponential Family Models. Factor Graph Representation. Hidden Markov Models. Conditional Random Fields. Triangulation and Chordal Graphs. Other
--	--

	Special Cases: Chains, Trees. Inference: Variable Elimination (Sum Product and Max-Product). Junction Tree Algorithm. Forward Backward Algorithm (for HMMs). Loopy Belief Propagation. Markov Chain Monte Carlo. Metropolis Hastings. Importance Sampling. Gibbs Sampling. Variational Inference. Learning: Discriminative Vs. Generative Learning. Parameter Estimation in Bayesian and Markov Networks. Structure Learning. EM: Handling Missing Data. Applications in Vision, Web/IR, NLP and Biology. Advanced Topics: Statistical Relational Learning, Markov Logic Networks.
--	--

15. Lecture Outline <i>(with topics and number of lectures)</i>
--

Module no.	Topic	No. of hours
1	Course Outline; Basics	2
2	Directed Graphical Models: Bayesian Networks	4
3	Markov Networks, Factor Graph Representation, Exponential Family Models	4
4	Hidden Markov Models, Conditional Random Fields	3
5	Exact Inference: Variable Elimination	4
6	Exact Inference: Junction Tree Algorithm, Forward Backward Algorithm	4
7	Approximation Inference: MCMC, Metropolis Hastings, Importance Sampling, Gibbs Sampling	6
8	Variational Inference	3
9	Parameter Learning in Bayesian and Markov Networks, Structure Learning	5
10	Handling Missing Data	3
11	Applications and Advanced Topics	4
COURSE TOTAL (14 times 'L')		42

16. Brief description of tutorial activities

N.A.

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
1	Implementing Bayesian Networks, Markov Networks, Naïve Bayes	6
2	Gibbs Sampling, Loopy Belief Propagation, Junction Tree Algorithm	6
3	Working with Applications (Vision, IR, NLP, Biology)	6

4	Course Project	10
COURSE TOTAL (14 times 'P')		28

18. Suggested texts and reference materials
STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Daphne Koller and Nir Friedman. "Probabilistic Graphical Models: Principles and Techniques". MIT Press, 2009.
2. M. I. Jordan (ed). "Learning in Graphical Models". MIT Press. 1998. Collection of papers.
3. J. Pearl. "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference." Morgan Kaufmann. 1988.

19. Resources required for the course <i>(itemized & student access requirements, if any)</i>
--

19.1	Software	C/C++/Java. Perl. Matlab. Publicly Available Online Datasets (e.g. IR, Vision, Biology etc.).
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20. Design content of the course <i>(Percent of student time with examples, if possible)</i>

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Computer Vision
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 780
6.	Status (<i>category for program</i>)	Post Graduate and Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	Earned Credits 90
----	--	-------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	EEL806
8.3	Supersedes any existing course	CSL840

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Every sem1 st sem2 nd sem Either sem
-----	------------------------------	--

11.	Faculty who will teach the course	Prof Subhashis Banerjee
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words</i>): This course is meant to be an introduction to the concepts and applications of computer vision. Computer vision, in general deals with methods of extracting and interpreting	
-----	---	--

	information from images and videos.
--	-------------------------------------

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Camera models. Calibration, multi-views projective geometry and invariants. Feature detection, correspondence and tracking. 3D structure/motion estimation. Application of machine learning in object detection and recognition, category discovery, scene and activity interpretation.
------------	--

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1	Camera Models, Calibration and Projective Geometry	6
2	Multi Views Geometry	6
3	Feature Extraction, Correspondence and Tracking	6
4	3D Structure/Motion Estimation	6
5	Applications: Single View Reconstruction, Large Scale Reconstruction	6
6	Object Detection and Recognition	6
7	Scene and Activity Interpretation	6
8		
9		
10		
11		
COURSE TOTAL <i>(14 times 'L')</i>		42

16.	Brief description of tutorial activities
------------	---

N.A.

17. Brief description of laboratory activities

The course structure is 3-0-2, to reinforce some concepts it is desired that students implement and experiment through assignments/projects in the course.

Module no.	Experiment description	No. of hours
1	Camera Calibration	8
2	Affine and Metric Rectification	8
3	Project (Open)	12
4		
COURSE TOTAL (14 times 'P')		28

18. Suggested texts and reference materials

STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Richard Hatley and Andrew Zisserman, Multiple View Geometry, Cambridge University Press
2. David Forsyth and Jean Ponce, Computer Vision: A Modern Approach
3. Research Papers

19. Resources required for the course *(itemized & student access requirements, if any)*

19.1	Software	Matlab/OpenCV for doing assignments/project.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	Standard
19.4	Laboratory	Standard
19.5	Equipment	Standard
19.6	Classroom infrastructure	Data Projector

19.7	Site visits	N.A.
------	-------------	------

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>	
------------	---	--

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	30%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Note: The course is designed to be a mix of theory as well as hands on experience. A background in Linear Algebra and Basic Probability is required.

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1	Department/Centre proposing the course	Department of Computer Science and Engineering
2	Course Title (< 45 characters)	ADVANCED TOPICS IN EMBEDDED COMPUTING
3	L-T-P structure	3-0-2
4	Credits	4
5	Course number	CSL 788
3	Status (category for program)	PE for CS5, MCS, VDTT

7	Pre-requisites (course no/title)	(CSL211 or EEL308 or MEL432) and (CSL373 or EEL358)
----------	--	---

8	Status vis-à-vis other courses (give course number/title)	
81	Overlap with any UG/PG course of the Dept/Centre	NIL
82	Overlap with any UG/PG course of another Dept/Centre	EEL705
83	Supersedes any existing course	NA

9	Not allowed for (indicate program names)	
----------	--	--

10	Frequency of offering	Every alternative semester
-----------	------------------------------	----------------------------

11	Faculty who will teach the course 1. Kolin Paul, S R Sarangi, Anshul Kumar	
12	Will the course require any visiting faculty?	No

13	Course objective <i>(about 50 words; motivation and aims of this course):</i> Modern Embedded Computing systems are used for feature rich highly integrated handheld mobile devices. The embedded platforms use sophisticated processors like Cortex and Atom to deliver performance. Topics ranging from using these embedded platforms to system initialization and debug will be covered. Program optimizations to ensure real time deadlines are met in typical media applications will be covered. The course will be systems oriented encompassing a mix of topics in Processor architecture and Real Time Operating systems.
14	Course contents <i>(about 100 words; this is a topics wise listing that will appear in the Courses of Study) (Include laboratory/design activities):</i> Embedded Platforms , Embedded processor architectures, System initialization, Embedded operating systems (linux) , DSP and graphics acceleration, Interfaces, Device Drivers, Network, Security, Debug support, Performance tuning. The course would involve substantial programming assignments on embedded platforms.

15	Lecture Outline <i>(with topics and number of lectures)</i>	
Module no	Topic	No of hours
1	Embedded Platforms	3
2	Embedded processor architectures	9
3	System initialization	3
4	Embedded operating systems (linux)	9
5	DSP and graphics acceleration	3
3	Network	3
7	Security	3
8	Performance tuning	3
9	Device Drivers	3
10	Advanced Issues	3
11		

12		
COURSE TOTAL (14 times 'L')		42

13	Brief description of tutorial activities

17 Brief description of laboratory activities		
Moduleno	Experiment description	No of hours
1	IDE in the linux environment Hello World	3
2	Boot Process	5
3	DSP	3
4	Multithreaded Programming	3
5	Real Time Performance	3
3	ICE and Debugging	3
7	Performance Tuning	3
8	Code Optimizations	5
9		
10		
COURSE TOTAL (14 times 'P')		28

18	Suggested texts and reference materials 1. Modern Embedded Computing: Designing Connected, Pervasive, Media-Rich Systems ISBN-10: 0123914903

19	Resources required for the course (itemized & student access requirements, if any)	
191	Software	Open Source
192	Hardware	
193	Teaching aides (videos, etc)	
194	Laboratory	
195	Equipment	Hardware kits
193	Classroom infrastructure	
197	Site visits	

20	Design content of the course (Percent of student time with examples, if possible)	
201	Design-type problems	25
202	Open-ended problems	05
203	Project-type activity	45
204	Open-ended laboratory work	15
205	Others (Critique Research Papers)	10

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Department of Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	PRINCIPLES OF MULTIPROCESSOR SYSTEMS
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 818
6.	Status (<i>category for program</i>)	Programme Elective for CSL, SIY, CSY, CSZ, ANZ

7.	Pre-requisites (course no./title)	CSL 211: Computer Architecture CSL 356: Analysis and Design of Algorithms CSL 373: Operating Systems
-----------	---	--

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	10% overlap with CSL 718 (advanced architectures) and CSL 730 (parallel programming)
8.2	Overlap with any UG/PG course of another Dept./Centre	No
8.3	Supersedes any existing course	NA

9.	Not allowed for (indicate program names)	NA
-----------	--	----

10.	Frequency of offering	Every alternate semester
------------	------------------------------	--------------------------

11.	Faculty who will teach the course 1. Dr. Smruti Ranjan Sarangi 2. Prof. S. Arun Kumar	
------------	--	--

	3. Prof. Sanjiva Prasad	
12.	Will the course require any visiting faculty?	No

13.	Course objective (<i>about 50 words; motivation and aims of this course</i>): This course will give graduate students a thorough understanding of the major concepts underlying the software and hardware aspects of shared memory multiprocessor systems. It will also expose them to the latest technologies in this rapidly emerging area.
14.	Course contents (<i>about 100 words; this is a topics wise listing that will appear in the Courses of Study</i>) (<i>Include laboratory/design activities</i>): (1) Mutual Exclusion (2) Coherence and Consistency (3) Register Constructions (4) Power of Synchronization Operations (5) Locks and Monitors (6) Concurrent queues (7) Futures and Work-Stealing (8) Barriers (9) Basics of Transactional Memory (TM) (10) Regular Hardware TMs (11) Unbounded Hardware TMs (12) Software TMs

15.	Lecture Outline (<i>with topics and number of lectures</i>)	
	Module no.	Topic
		No. of hours
	1	Introduction
	2	Mutual Exclusion
	3	Coherence and Consistency
	4	Register Constructions
	5	Power of Synchronization Operations
	6	Universal Construction
	7	Locks
	8	Monitors
	9	Concurrent Queues
	10	Futures and Work Stealing
	11	Barriers
	12	Basics of Transactional Memory
	13	Regular Hardware Transactional Memory
	14	Unbounded Hardware Transactional Memory
	15	Opacity and Fo-consensus
	16	Software Transactional Memory
	COURSE TOTAL (<i>14 times 'L'</i>)	
		42

16.	Brief description of tutorial activities
	NA

17. Brief description of laboratory activities		
Module no.	Experiment description	No. of hours
1	Design of mutual exclusion primitives	5
2	Implementation of non-blocking data structures	5
3	A parallel future based work scheduler	6
4	Writing programs using software transactional memory frameworks	6
5	Implementation of a minimal STM	6
COURSE TOTAL (14 times 'P')		28
18.	Suggested texts and reference materials 1. Art of Multiprocessor Programming, Herlihy and Shavit, 1st edition, 2009. Morgan Kaufmann Publications 2. Principles of Transactional Memory, Guearroui and Kapalka, 1st edition, 2011. Morgan and Claypool Publications	
19.	Resources required for the course (itemized & student access requirements, if any)	
19.1	Software	C++ and Java compilers
19.2	Hardware	Parallel computer with at least 16 cores. They are already available in the CSE and SIT deptts.
19.3	Teaching aides (videos, etc.)	No
19.4	Laboratory	No
19.5	Equipment	No
19.6	Classroom infrastructure	None
19.7	Site visits	None
20.	Design content of the course(Percent of student time with examples, if possible)	
20.1	Design-type problems	20.00%
20.2	Open-ended problems	
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	Department of Computer Science and Engineering
2.	Course Title (< 45 characters)	ADVANCED DISRIBUTED SYSTEMS
3.	L-T-P structure	3-0-2
4.	Credits	4
5.	Course number	CSL 819
6.	Status (category for program)	PE for CS5, CSZ, CSY, SIY, ANZ, MCS

7.	Pre-requisites (course no./title)	CSL 373 (Operating systems), CSL 374 (Computer networks), CSL 380 (Introduction to parallel and distributed computing)
----	---	--

8.	Status vis-à-vis other courses (give course number/title)	
8.1	Overlap with any UG/PG course of the Dept./Centre	CSL 830 (Distributed Computing) – 10% overlap, CSL 847(Distributed Algorithms) – 15% overlap
8.2	Overlap with any UG/PG course of another Dept./Centre	NA
8.3	Supersedes any existing course	NA

9.	Not allowed for (indicate program names)	
----	--	--

10.	Frequency of offering	Every Alternate Semester
-----	------------------------------	--------------------------

11.	Faculty who will teach the course 1. Smruti Ranjan Sarangi 2. Sanjiva Prasad 3. Sorav Bansal	
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words; motivation and aims of this course):</i> The course has broadly three objectives – understanding of the basic paradigms in large scale distributed systems, basic distributed algorithms, and a description of real life case studies.
14.	Course contents <i>(about 100 words; this is a topics wise listing that will appear in the Courses of Study) (Include laboratory/design activities):</i> 1) Epidemic/Gossip based algorithms 2) Peer to peer networks 3) Distributed hash tables 4) Synchronization 5) Mutual exclusion 6) Leader election 7) Distributed fault tolerance 8) Large scale storage systems 9) Distributed file systems 10) Design of social networking systems

15.	Lecture Outline <i>(with topics and number of lectures)</i>	
Module no.	Topic	No. of hours
1	Introduction	3
2	Epidemic based algorithms	3
3	Gossip based algorithms	2
4	Peer to peer systems	7
5	Synchronization and logical clocks	3
6	Consistency and replication	3
7	Mutual exclusion and leader election	2
8	Fault Tolerance	5
9	Distributed storage systems	4
10	Distributed file systems	3
11	Publish subscribe systems	2
12	Design of social networking platforms	5
Total		42

16.	Brief description of tutorial activities
	None

17. Brief description of laboratory activities		
Module no.	Experiment description	No. of hours
1	Implementation of a distributed hash table	8
2	Implementation of a distributed algorithm. Examples: Finding the minimum spanning tree, leader election, mutual exclusion	8
3	Implement a distributed file system	6
4	Implementation of a publish subscribe system	6
5		
6		
7		
8		
9		
10		
COURSE TOTAL (14 times 'P')		28
18.	Suggested texts and reference materials 1. Distributed Systems (Tanenbaum and Marten V. Steen) 2. Distributed Algorithms (Gerard Tel) 3. Distributed Algorithms (Nancy Lynch) 4. Introduction to Reliable and Secure Distributed Programming (Rachid Guerraoui)	
19.	Resources required for the course (itemized & student access requirements, if any)	
19.1	Software	Open source software
19.2	Hardware	24 core servers, present in the CSE department
19.3	Teaching aides (videos, etc.)	
19.4	Laboratory	
19.5	Equipment	
19.6	Classroom infrastructure	
19.7	Site visits	
20.	Design content of the course (Percent of student time with examples, if possible)	
20.1	Design-type problems	30
20.2	Open-ended problems	30
20.3	Project-type activity	20
20.4	Open-ended laboratory work	20
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Special Topics in Cryptography
3.	L-T-P structure	3-0-0
4.	Credits	3
5.	Course number	CSL 872
6.	Status (<i>category for program</i>)	Post Graduate

7.	Pre Requisites (<i>course no./title</i>)	CSL 759 or MAL 786
----	--	---------------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	No
8.2	Overlap with any UG/PG course of other Dept./Centre	No
8.3	Supersedes any existing course	No

9.	Not allowed for (<i>Indicate program names</i>)	N.A
----	---	-----

10.	Frequency of offering	Every alternate semester, following CSL 759
-----	------------------------------	---

11.	Faculty who will teach the course	ShwetaAgrawal RageshJaiswal
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> This course is meant to be a Special Topics course in Cryptography. The objective of this course is to develop an in depth knowledge and understanding of new results and current trends in the field. A set of advanced topics would be covered. The course is also aimed to impart skills for carrying out research in the area of Cryptography.
------------	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> Contents may vary based on the instructor's interests within the broader area of Cryptography. Examples include CCA secure encryption, multiparty computation, leakage resilient cryptography, broadcast encryption, fully homomorphic encryption, obfuscation, functional encryption, zero knowledge, private information retrieval, byzantine agreement, cryptography against extreme attacks etc.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i> Exact Outline will depend on the instructor and the specific content of the course being offered.
------------	---

16.	Brief description of tutorial activities
------------	---

N.A.

17.	Brief description of laboratory activities N.A.
------------	---

18.	Suggested texts and reference materials STYLE: Author name and initials, Title, Edition, Publisher, Year.
------------	---

1. Foundations of Cryptography I and II by Oded Goldreich. Cambridge University Press, 2001. ISBN 0-521-79172-3. 2. Online Reading Material.

19.	Resources required for the course <i>(itemized & student access requirements, if any)</i>
------------	--

19.1	Software	N.A
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	N.A.
19.4	Laboratory	N.A.
19.5	Equipment	N.A.
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the Course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	50%
20.2	Open-ended problems	50%
20.3	Project-type activity	
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department/Centre proposing the course	CSE Dept
2.	Course Title (<i>< 45 characters</i>)	SPECIAL MODULE IN FINANCIAL ALGORITHMS
3.	L-T-P structure	2-0-0
4.	Credits	2-0-0
5.	Course number	CSV879
6.	Status (<i>category for program</i>)	Open Elective
7.	Pre-requisites (<i>course no./title</i>)	MAL 250 or MAL 140

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	About 20% overlap with
8.2	Overlap with any UG/PG course of other Dept./Centre	MAL 732 & MAL 733
8.3	Supercedes any existing course	No

9.	Not allowed for (<i>indicate program names</i>)	Nil
-----------	---	-----

10.	Frequency of offering	~~Every Sem~~ / ~~1st Sem~~ / ~~2nd Sem~~ / Either Sem
------------	------------------------------	---

11.	Faculty who will teach the course Dr. Prasoon Tiwari	
12.	Will the course require any visiting faculty?	Yes, Dr. Prasoon Tiwari B65 SFS Sheikh Sarai Phase-I New Delhi 110017

13.	Course objective (<i>about 50 words</i>): The module will start with model independent results in replication of derivatives, and then move on to pricing of derivatives using the binomial method followed by Black-Scholes formula derivation via stochastic calculus. Various Financial Algorithms for Statistical arbitrage, structured products and volatility derivatives will be discussed and the module will conclude with introduction to martingale methods.
------------	--

14.	Course contents (<i>about 100 words</i>) (<i>Include laboratory/design activities</i>): The indicative contents will be from a mix of the following 1. Model Independent Results (10%): 2. Vanilla/Exotic Options and Binomial Pricing (15%) 3. Stochastic Calculus (10%) 4. Black-Scholes (10%)
------------	--

	5. Statistical Arbitrage (10%) 6. Structured Products (10%) 7. VIX (10%) 8. Martingales (10%) 9. Trading Strategies (15%) Grading Scheme The class will have a project where students will be asked to come up with a trading strategy for a market scenario of their choice. For this project, students will organize as team of 2 or more. Students will make presentations on the strategy. The following is the percentage of grading • Final 65% • Trading Strategy Presentation 20% • Class Participation 15%
--	---

15. Lecture Outline *(with topics and number of lectures)*

Module no.	Topic	No. of hours
1	Introduction	1
2	Model Independent Derivative Replication	3
3	Binomial Pricing of Vanilla and Exotic Derivatives	5
4	Brownian Motion	1
5	Stochastic Calculus	3
6	Black-Scholes Formula	3
7	Statistical Arbitrage	3
8	Structured Products	2
9	VIX	2
10	Martingale Methods	3
11	Trading Strategy Presentation	2
COURSE TOTAL (28 times 'L')		28

16. Brief description of tutorial activities

Students will be asked to come up with a market scenario and a trading strategy for that market scenario.

17. Brief description of laboratory activities

Module no.	Experiment description	No. of hours
	No specific lab activities for this project.	

COURSE TOTAL (14 times 'P')		

18. Suggested texts and reference materials

Text:

- Options, Futures, and Other Derivatives by John C. Hull.
- Class handouts.

Other useful books:

- Option Markets by John C. Cox and M. Rubinstein.

19. Resources required for the course (*itemized & student access requirements, if any*)

19.1	Software	
19.2	Hardware	Computers
19.3	Teaching aides (videos, etc.)	
19.4	Laboratory	Standard software development lab will do.
19.5	Equipment	
19.6	Classroom infrastructure	Projector for PowerPoint Slides (i will bring my own laptop)
19.7	Site visits	

20. Design content of the course (*Percent of student time with examples, if possible*)

20.1	Design-type problems	15% Students will define a market scenario and design a trading strategy for it. This will be presented in class.
20.2	Others (please specify)	85% Lectures

Date:

(Signature of the Head of the Department)

COURSE TEMPLATE

1.	Department /Centre proposing the course	Computer Science and Engineering
2.	Course Title (<i>< 45 characters</i>)	Special Module on Visual Computing
3.	L-T-P structure	1-0-0
4.	Credits	1
5.	Course number	CSV 890
6.	Status (<i>category for program</i>)	Post Graduate and Senior Under Graduate

7.	Pre Requisites (<i>course no./title</i>)	Earned Credits 120
----	--	--------------------

8.	Status vis-à-vis other courses (<i>give course number/title</i>)	
8.1	Overlap with any UG/PG course of the Dept./Centre	
8.2	Overlap with any UG/PG course of other Dept./Centre	
8.3	Supersedes any existing course	

9.	Not allowed for (<i>indicate program names</i>)	
----	---	--

10.	Frequency of offering	Every sem 1 st sem 2 nd sem <u>Either sem</u>
-----	------------------------------	---

11.	Faculty who will teach the course	Prof Subhashis Banerjee Prof Prem Kalra Prof Subodh Kumar
12.	Will the course require any visiting faculty?	No

13.	Course objective <i>(about 50 words):</i> This course is meant to be a module with emphasis on some research topics in the broad area of visual computing.
------------	--

14.	Course contents <i>(about 100 words) (Include laboratory/design activities):</i> The course will be a seminar-based course where the instructor would present topics in a selected theme in the area of visual computing through research papers. Students will also be expected to participate in the seminar.
------------	---

15.	Lecture Outline <i>(with topics and number of lectures)</i>
------------	--

Module no.	Topic	No. of hours
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
COURSE TOTAL <i>(14 times 'L')</i>		

16.	Brief description of tutorial activities
------------	---

N.A.

17. Brief description of laboratory activities

The course structure is 1-0-0, however to reinforce some concepts it may be desired that students implement and experiment through assignments/projects in the course.

Module no.	Experiment description	No. of hours
1		
2		
3		
4		
COURSE TOTAL (14 times 'P')		

18. Suggested texts and reference materials

STYLE: Author name and initials, Title, Edition, Publisher, Year.

1. Research Papers**19. Resources required for the course** (*itemized & student access requirements, if any*)

19.1	Software	Matlab/OpenCV/OpenGL for doing assignments/project.
19.2	Hardware	Personal Laptop or Computer Access
19.3	Teaching aides (videos, etc.)	Standard
19.4	Laboratory	Standard
19.5	Equipment	Standard
19.6	Classroom infrastructure	Data Projector
19.7	Site visits	N.A.

20.	Design content of the course <i>(Percent of student time with examples, if possible)</i>
------------	---

20.1	Design-type problems	
20.2	Open-ended problems	20%
20.3	Project-type activity	20%
20.4	Open-ended laboratory work	
20.5	Others (please specify)	

Note: The course is designed to be a mix of theory as well as hands on experience. A background in Linear Algebra and Basic Probability is required.

Date:

(Signature of the Head of the Department)