

International Institute of Information Technology, Hyderabad School of Multi-disciplinary Computing

B. Tech. in Computer Science and Engineering

Table of Contents

Sl. No	Item
1	Vision and Mission Statements of the School of
1.	Multi-disciplinary Computing
2.	PEOs, POs, and PSOs
3.	Curriculum
1	Course Descriptions for Core, Breadth Electives and
4.	Bouquet Core courses



1. Vision and Mission for the School of Multi-disciplinary Computing

Vision

To be recognized as a globally reputed school by offering innovative academic programs and specializations in core computing, computing technologies, and computing in association with multiple disciplines, at all levels (UG, PG, Ph.D.) with state-of-the-art curricula, by promoting quality research in thrust areas, and blending research outcomes into teaching programs.

Mission Statements

MS1: To produce competent next-generation technology leaders, who can apply the science and engineering of computing to add immense value to their profession.

MS2: To implement a state-of-the-art curriculum in all the academic programs in line with the multidisciplinary societal and technological needs and encourage students to imbibe creativity, research, problem-solving skills, professional ethics, and human values.

MS3: To design and execute innovative multidisciplinary academic programs, specializations, and courses that combine computing and other domains organically, by involving all the stakeholders such as students, teachers, research scholars, experts from industry, academia, and alumni.

MS4: To conduct quality research in fundamental, applied, multidisciplinary, and futuristic domains and become a key player in the educational ecosystem within the country and abroad.

MS5: To create and sustain a strong suite of academic outreach programs catering to varied segments such as industry professionals, external students, and early career researchers.

MS6: To collaborate with other reputed institutions in India and abroad and implement best practices to achieve excellence.



2. PEOs, POs, and PSOs for the B.Tech in Computer Science and Engineering

Program Educational Objectives (PEOs)

After completing B.Tech in Computer Science and Engineering successfully, the graduates will be able to

PEO1: Demonstrate competency and creativity in different fields of Computer Science and Engineering and develop software systems and solutions for technological and societal applications.

PEO2: Demonstrate the requisite breadth and depth of knowledge in advanced areas of Computer Science and Engineering and also problem solving skills so as to excel in research-centric industry and academic environments.

PEO3: Exhibit communication skills and collaborative skills required to function effectively in a varied, dynamic and multi-member teams.

PEO4: Develop an aptitude for self-learning and life-long learning so as to keep abreast with rapidly evolving technologies.

PEO5: Practice ethics and human values in their profession.

PEO vs MS	MS1	MS2	MS3	MS4	MS5	MS6
PEO1	3	3	3	3	3	2
PEO2	3	3	3	3	3	3
PEO3	2	3	3	3	2	3
PEO4	3	3	3	3	3	3
PEO5	3	3	2	2	2	2

Mapping between PEOs and Mission Statements



Program Outcomes (POs)

PO1 Engineering knowledge: Use concepts from varied disciplines including Computer Science, Electronics, Mathematics, and the Sciences, to engineer and develop systems of varying scale.

PO2 Problem analysis: Identify, formulate and analyze complex engineering problems reaching substantial conclusions using first principles of Mathematics, Natural Sciences and Engineering Sciences.

PO3 Design/Development of solutions: Identify and bring to fore the necessary concepts from Computer Science and arrive at creative ways to solve problems that take into account the societal, cultural, and ethical considerations.

PO4 Conduct investigations of complex problems: Interpolate and extrapolate based on existing knowledge base and self-learning skills to investigate the dynamics of complex problems and find solutions.

PO5 Modern tool usage: Demonstrate requisite hands-on skills to work with a variety of software packages, libraries, programming languages, and software development environment tools useful in engineering large scale systems

PO6 The engineer and society: Make judicious use of resources and understand the impact of technology across the societal, ethical, environmental, and economic aspects.

PO7 Environment and sustainability: Find technological solutions by considering the environmental impact for sustainable development

PO8 Ethics: Practice principles of professional ethics and make informed decisions after a due impact analysis.

PO9 Individual and team work: Work efficiently in individual and team-oriented projects of varying size, cultural milieu, professional accomplishments, and technological backgrounds.

PO10 Communication: Effectively communicate and exchange ideas and solutions to any individual including peers, end-users, and other stakeholders.

PO11 Project management and Finance: Apply the principles of project management in general and software project management in particular with focus on issues such as the life cycle, scoping, costing, and development.

PO12 Life-long learning: Exhibit the aptitude for independent, continuous, and life-long learning required to meet their professional and career goals.



Program Specific Outcomes (PSOs)

PSO1 Exhibit specialized knowledge in some sub-areas of Computer Science and Engineering such as Theoretical Computer Science, Computer Systems, Artificial Intelligence, Cyber-physical Systems, Cyber-security and use this specialized knowledge base to solve advanced problems

PSO2 Perform gap analysis in terms of systems and technologies and prepare roadmaps for incorporating state-of-the-art technology into system analysis, design, implementation, and performance.

PSO3 Demonstrate research and development skills needed to define, scope, develop, and market futuristic software systems and products

PSO4 Demonstrate knowledge and skills at the required depth and breadth to excel in post-graduate and research programs

PO vs PEO	PEO1	PEO2	PEO3	PEO4	PEO5
PO1	3	3	2	2	2
PO2	3	3	2	2	2
PO3	3	3	2	2	2
PO4	3	3	2	2	2
PO5	3	3	2	3	2
PO6	2	2	2	3	3
PO7	2	2	2	2	3
PO8	2	2	2	2	3
PO9	2	2	3	2	3
PO10	2	2	3	2	2
PO11	2	2	3	2	2
PO12	2	2	2	3	2
PSO1	3	3	3	2	2
PSO2	3	3	3	2	2
PSO3	3	3	3	2	2
PSO4	3	3	3	2	2

Mapping between POs, PSOs and PEOs



Curriculum

The curriculum for CSE for the next ten years, 2018-2028, is prepared keeping in mind the pace of change in computing as a discipline. Computing now is affecting many domains including the natural sciences, materials, and humanities in a big way. It is therefore imperative that the curriculum offers a student the scope to acquire fundamental knowledge quickly and the ability to specialize in some subject domain. The curriculum design used this as the main goal.

Any curriculum design exercise is an expedition into the unknown future. One has to contend with making recommendations by projecting the present to the future. For instance, circa 2008, it is believed that Computer Graphics will develop in a big way given the thrust on gaming and animation. While some of that is true, the extent of that growth is not revolutionary. Similarly, circa 2008, statistical and data-driven artificial intelligence and machine learning were still nascent. Today, many departments are heavily invested in these areas.

Given the difficulty of this task, the committee took the view that a curriculum should be such that midterm corrections should be possible and smooth. Viewed differently, the basic structure of the curriculum need not change but some elements of the curriculum can be modified to adjust to a changing reality.

On the other hand, IIIT Hyderabad strives to impart education that does not simply train students get their first employment but rather trains students to be thought-leaders in their respective domains. Doing so requires a very hands-on curriculum and flexibility that allows individual students to focus on their strengths and interests to create a curriculum for themselves. At the same time, the curriculum will have a minimal core/foundation that helps students in making informed choices going forward.

Keeping the above observations in mind the curriculum is created as a set of 52 Computer Science credits in the first four semesters of which 46 credits are common for everybody. In these 46 credits, courses include programming, data structures, algorithms, circuits, computer systems, operating systems, computer networks, database systems, and software engineering. These courses can be viewed as offering the students from the basics of the discipline, prepare them for problem-solving, and include lots of hands-on projects. By the end of these courses, students will be able to create systems that are requirement and process driven, version-controlled, talk over the network, prepare SQL queries, and are usable. Indeed, IIIT Hyderabad has a history of using several such systems.

To create a good synergy between CS and ECE, within these credits, we also recommend hands-on courses in EECS. Through these courses, students will be able to build simple systems that can bring together several aspects of EECE such as sensors, converters, decision making/feedback, communication, and the like. The remaining 6 credits in the minimal CS is driven by two credit courses that prepare them for depth in a broad area. These courses therefore serve to introduce the required body of knowledge. These 6 credits are earned by choosing three courses out of a set of about 8 courses.

The curriculum across semesters five through eight is largely elective driven. Students are encouraged to do five courses (20 credits) in Computer Science apart from other open choice credits, science courses, and courses in humanities. The five courses in Computer Science can be from one subject area such as AI or be spread across several areas depending on the interests of the individual student. These courses can be also from other applied domains such as computational science or building science. We consider that



every student will do at least one course from the Theory bouquet and at least one course from the Systems bouquet.

It is therefore considered that 18 courses adding up to 72 credits plus a 4 credit B. Tech. Project is the minimal CS requirement. This layer of 76 credits is appropriate for students wanting to specialize in domains outside of CS. Most CSE students will be encouraged to do additional CSE credits.

The proposed curriculum, following the present curriculum, also allows for strange possibilities that a student can miss some subjects such as Compilers and still obtain the degree. We would see this as a strength of the curriculum rather than as a weakness. Students are encouraged to identify their interests and pursue those interests at a deep level of engagement. Student passing through our curriculum should be known for what they are good at rather than being pointed out what they are short of.

1.1 Salient Features

- 1. A minimal core that is compulsory, and a braoder definition of CS for enabling learning, projects and research in many emerging areas of computer science.
- 2. BTech (CSE) has 47.5% of the credits coming from the major subjects (broader CSE). Out of this, 23.75% of the credits are preplanned core courses, 2.5% of credits as compulsory projects, 5.0% of the credits as courses that student can flexibly schedule and 16.25% of the credits as electives.
- 3. More converging view of CS and ECE with courses that can be seen as EECS which are getting into the mainstream in both CSE and ECE.
- 4. Smaller credits in the first year so that (i) students can migrate to the new environments more easily (ii) trans-disciplinary programs get more space for introducing basic concepts in the early years.
- 5. The notion of breadth elective early into the course (in second year) so that students can taste different disciplines and take well informed choice on specialization.

The CSE curriculum is organized as three tiers. The first tier corresponds to Core CS that is common to all students. The second tier is the tier of bouquet courses which allows students to specialize in a broad area. The third tier of courses is the open tier that corresponds to open electives.

In the first tier, we have a set of introductory courses that span areas such as programming, data structures, algorithms, computer systems organization, operating systems, embedded systems with a hands-on project, software engineering including a project, and the like.

This first tier ends with six credits of half-courses that students

can take from a collection of about eight courses. These courses are organized into two groups: Theory, and Systems. The idea behind these two groups is to give students a chance to explore four domains of Computer Science so that eventually they may select one of these areas to specialize in.

The first four semesters of the curriculum cover courses in the first tier. Table 1 shows the placement of these courses over the four semesters.



Table 2: Courses in the First Four Semesters and Credits

Yea r	Sem	Categor y	Course Name	Credit s
1	1	Mat	Discrete Structures	4
1	1	PC	Computer Programming	5
1	1	Mat	Real Analysis	4
1	1	EC	Digital Systems and Microcontrollers	5
			Total	18
1	2	EC	Introduction to Embedded Systems (2 lecture + 3 lab per week)	3
1	2	PC	Data Structures and Algorithms	5
1	2	PC	Introduction to Software Systems (full semester, 3 hr lab, 1 hr lecture)	2
1	2	PC	Computer Systems Organization (little OS)	4
1	2	Mat	Linear Algebra (common with ECE)	4
			Total	18
2	3	Mat	Probability and Statistics	4
2	3	PC	Design and Analysis of Software Systems	4
2	3	PC	Algorithm Analysis and Design	4
2	3	PC	Computer Systems Engineering - 1 Equiv. to OS + CN	4
2	3	Sci	Science -1	4
2	3	EC	Embedded Systems Workshop	2
			Total	22
2	4	PE	Breadth Elective-1	2
2	4	PC	Introduction to Database Systems	2
2	4	PC	Automata Theory* (Flexicore), H1	2
2	4	PE	Breadth Elective – 2	2
2	4	PE	Breadth Elective – 3	2
2	4	PC	Intelligent Systems (like AI/ML)*Flexicore	4
2	4	Sci	Science 2	4
2	4	Hum	Intro to HS	4
			Total	22

The second tier of courses consist of a four-credit B. Tech. major project split over two semesters, the sixth and the seventh. Apart from this, students have to acquire 20 credits, four courses, from this tier. Courses in this tier are first-level and second-level electives in different areas of Computer Science. To allow students to specialize, we have arranged courses in this tier as five buckets. The buckets are: Theory, Systems, Artificial Intelligence (AI), EECS, and IT for Applications.

These buckets justify their existence in a natural manner. Presently, AI is experiencing a major revival in terms of courses, opportunities, applications, and impact. Therefore, we wish to equip interested students to specialize in this area with courses such as Computer Vision, Machine Learning, Information Retrieval, Data Mining, and the like. The Theory bucket will have courses such as Advanced Algorithms,



Optimization Methods, Modern Information Security, and Principles of Programming Languages. These courses have traditionally served to advance an interest in theoretical areas. The Systems bucket will build on the systems background from the first tier and include courses such as Distributed Systems, Compilers, Advanced Database Systems, Advanced Computer Networks, and Advanced Software Engineering.

The EECS bucket will be offered common with ECE students and will have courses such as Robotics, Sensor Networks, Internet-of-Things, Cyber-Physical Systems, and the like. As these areas are entering our daily lives, it is expected that students who specialize in this area will be able to position themselves as experts at the boundary of CS and ECE.

The IT for Applications bucket will have courses that use computing to advance other disciplines.

The third tier of courses are left to the open category. Students can take any course including those from Computer Science. These cover another 20 credits over five courses. Apart from these, the curriculum will have four science courses (2 core and 2 elective), four mathematics courses, and five (1 core and 4 elective) courses in Human sciences. The placement of courses from the fifth semester onward is shown in Table 3.

Choice Based Credit System at IIITH:

The curriculum aims to continue the implementation of Choice Based Credit System with a minimal core program followed by electives from across disciplines including mathematics, sciences, human sciences, engineering electives, and so on. The curriculum set aside close to 17% of the credits necessarily from courses outside of the program so as to allow scope for students to credit courses from the sciences, mathematics, human sciences, and engineering sciences. A total of 12% credits are set aside as open electives – students can use these credits to either go deeper in the program or to opt for courses outside of the program and broaden their outlook by opting for multi-disciplinary courses too.

All courses use a continuous evaluation model with a combination of homework assignments, quiz exams, mid-term, and final examinations. Students are required to stay clear of plagiarism in any of their work submitted for evaluation. Most elective courses include a course project or a term paper additionally. These course projects often require students to practice team-work, enhance their self-learning and communication skills, and impart essential project management skills. Some courses include a laboratory component with a scheduled laboratory session.

For the highly motivated students, the present curriculum continues to provide the Honors option which requires students to do additional credits including projects and advanced electives and work under the supervision of a faculty member.



	Comp	uter Science and E	ngineer	ring -	- Se	mest	er wi	se Pl	an (S	Subjec	t to mi	nor c	hange	es)
Yea r & Se	Course Code	Course Name	Full/H alf	Leo Tu Pra Ho	cture toria actica ours j week	s-L, -T, l-P per				Cre	edits earn	ed		
m	Coue		Sem	L	Т	Р	Ins Cor e	Pro g Cor e	Pro g Ele c	Math s	Scienc e	Hu m	Ope n Elec	Hons
						Mon	soon							
	MA5.10 1	Discrete Structures	Full	3	1	0	-	-	-	4	-	-	-	-
	CS0.101	Computer Programming	Full	3	1	3	-	5	-		-	-	-	-
	MA4.10 1	Real Analysis	Full	3	1	0	-	-	-	4	-	-	-	-
I-I	EC2.101	Digital Systems and Microcontrollers	Full	3	1	3	-	5	-	-	-	-	-	-
	OC2.101	Arts-1	Full	0	2	2	2	-	-	-	-	-	-	-
	OC1.101	Sports-1	Full	0	0	2	1	-	-	-	-	-	-	-
	OC3.101	Value Education-1	Full	2	1	0	2	-	-	-	-	-	-	-
		Sub Total		14	7	10	5	10	0	8	0	0	0	0
		Total												23
					T	Spr	ing							
	MA2.10 1	Linear Algebra	Full	3	1	0	-	-	-	4	-	-	-	-
	CS3.303	Introduction to IoT	Full	2	0	3	-	3	-	-	-	-	-	-
	CS1.201	Data Structures and Algorithms	Full	3	1	3	-	5	-	-	-	-	-	-
I-II	CS6.201	Introduction to Software Systems	Half	1	0	3	-	2	-	-	-	-	-	-
	CS2.201	Computer Systems Organization	Full	3	1	0	-	4	-	-	-	-	-	-
	OC1.102	Sports-2	Full	0	0	2	1	-	-	-	-	-	-	-
	OC2.102	Arts-2	Full	0	2	2	2	-	-	-	-	-	-	-
		Sub Total		12	5	13	3	14	0	4	0	0	0	0
		Total												21
			1		.	Mon	soon							
II-I	MA6.10 1	Probability and Statistics	Full	3	1	0	-	-	-	4	-	-	-	-
11-1	CS4.301	Data and Applications	Half 2	3	1	0	-	2	-	-	-	-	-	-
	CS1.302	Automata Theory	Half 1	3	1	0	-	2	-	-	-	-	-	-

THE DEST
INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

	CS1.301	Algorithm Analysis and Design	Full	3	1	0	-	4	-	-	-	-	-	-
	CS3.301	Operating Systems and Networks	Full	3	1	0	-	4	-	-	-	-	-	-
	SC1.110	Science -1	Full	3	1	0	-	-	-	-	4	-	-	-
	EC3.202	Embedded Systems Workshop	Half	1	0	2	-	3	-	-	-	-	-	-
	OC1.103	Sports-3	Full	0	0	2	1	-	-	-	-	-	-	-
		Sub Total		19	6	4	1	15	0	4	4	0	0	0
		Total												24
		1			1	Spr	ing							
	CSx.3nn	Theory/Foundation Breadth Elective – 1	Half	3	1	0	-	-	2	-	-	-	-	-
	CSx.3nn	Theory/Foundation Breadth Elective – 2	Half	3	1	0	-	-	2	-	-	-	-	-
	CSx.3nn	Theory/Foundation Breadth Elective – 3	Half	3	1	0	-	-	2	-	-	-	-	-
	CS6.301	Design and Analysis of Software Systems	Full	3	1	0	-	4	-	-	-	-	-	-
II- II	CS7.301	Machine, Data and Learning	Full	3	1	0	-	4	-	-	-	-	-	-
I	SC1.111	Science 2	Full	3	1	0	-	-	-	-	4	-	-	-
-	HS8.102	Intro to Human Sciences	Full	3	1	0	-	-	-	-	-	4	-	-
I	OC1.104	Sports-4	Full	0	0	2	1	-	-	-	-	-	-	-
I	OC3.102	Value Education-2	Full	2	1	0	2	-	-	-	-	-	-	-
I		Sub Total		23	8	2	3	8	6	0	4	4	0	0
ľ		Total												25
						Mon	soon							
I		Bouquet Course – 1	Full	3	1	0	-	-	4	-	-	-	-	-
		Science Elective – 1	Full	3	1	0	-	-	-	-	4	-	-	-
III- I		HS Elective-1	Full	3	1	0	-	-	-	-	-	4	-	-
1		Open Elective-1	Full	3	1	0	-	-	-	-	-	-	4	-
I		Sub Total		12	4	0	0	0	4	0	4	4	4	0
I		Total												16
						Spr	ing							
III-	CS9.301/ CS9.303	BTP-1 / Honours-2	Full	0	2	6	-	-	-	-	-	-	-	2
II		Bouquet Course – 2	Full	3	1	0	-	-	4	-	-	-	-	-
						0			4					
		Bouquet Course – 3	Full	3	1	0	-	-	4	-	-	-	-	-

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY H Y D E R A B A D

		Open Elective-2	Full	3	1	0	-	-	-	-	-	-	4	-
		Sub Total		12	6	6	0	0	8	0	0	4	4	2
		Total												18
						Mon	soon							
	CS9.401/ CS9.402	BTP-2 / Honours-3	Full	0	2	6	-	-	-	-	-	-	-	2
		Open Elective-3	Full	3	1	0	-	-	-	-	-	-	4	-
IV-		Bouquet Course – 4	Full	3	1	0	-	-	4	-	-	-		-
1 -		Open Elective-4	Full	3	1	0	-	-	-	-	-	-	4	-
		Ethics-1	Half	3	1	0	-	-	-	-	-	2	-	-
		Sub Total		12	6	6	0	0	4	0	0	2	8	2
		Total												16
		Bouquet Course – 5	Full	3	1	0	-	-	4	-	-	-	-	-
		HS Elective-3	Full	3	1	0	-	-	-	-	-	4	-	-
IV-		Open Elective-5	Full	3	1	0	-	-	-	-	-	-	4	-
II		Ethics-2	Half	3	1	0	-	-	-	-	-	2		-
		Science Elective –2	Full	3	1	0	-	-	-	-	4	-	-	-
		Sub Total		15	5	0	0	0	4	0	4	6	4	0
IV- I		Total												18

	Tota l	%
Institute Core	12	8
Programme Core	47	29
Programme Electives	26	16
Maths	16	10
Science	16	10
Humanities	20	13
Open Electives	20	13
BTP	4	3
Total	161	100



Graduation Requirements

B.Tech in Computer Science and Engineering (CSE)

In order to graduate with B.Tech in Computer Science and Engineering, a student must successfully complete **161** credits with **minimum CGPA of 5.5** and meet the following requirements.

- Must successfully complete the **SAVE** (Sports, Arts, Value Education) credits in the 1st and 2nd years.
- Must successfully complete the **programme** Core.
- Must successfully complete at least 3 **Breadth Electives (Half courses)** in 4th semester and beyond.
- Must successfully complete **5 Bouquet electives** in the 3rd and 4th years (at least 1 each in semester and not more than 2 in any semester).
- Must successfully complete **1 Maths elective** in the 3rd and 4th years.
- Must successfully complete **2** Science electives in the 3rd and 4th years (not more than 1 in any semester).
- Must successfully complete **3 Humanities electives** in the 3rd and 4th years (not more than 6 credits in any semester).
- Must successfully complete **2 courses** (2 credits each) in Ethics in the 4th year.
- Must successfully complete **5 Open electives** in the 3rd and 4th years (at least 1 each in semester and not more than 2 in any semester).
- Must successfully complete **4 BTP credits via 2 credits each** in the 6th and 7th Semesters.



Course Descriptions for the core courses

Discrete Structures

Name of the Academic Program: B. Tech. in CSECourse code: MA5.101L-T-P: 3-1-0Credits: 4L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. <u>Prerequisite Course / Knowledge</u>:

Basic abstract algebra, High School Mathematics

2. <u>Course Outcomes (COs</u>)

After completion of this course successfully, the students will be able to

CO-1: Demonstrate critical thinking, analytical reasoning, and problem solving skills

CO-2: Apply appropriate mathematical and probabilistic concepts and operations to interpret data and to solve problems

CO-3: Identify a problem and analyze it in terms of its significant parts and the information needed to solve it **CO-4:** Formulate and evaluate possible solutions to problems, and select and defend the chosen solutions

CO-5: Construct graphs and charts, interpret them, and draw appropriate conclusions

CO-6: Apply the concepts of group theory, ring and field in various applications in computer science

3. <u>MappingofCourseOutcomes(COs)withProgramOutcomes(POs)andProgramSpecificOutcomes(PSOs) – Course Articulation Matrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PSO1	PSO2	PSO3	PSO4
CO1	1	2	2	2	2	2	2	2	3	2	2	2	1	2	2	3
CO2	3	3	3	2	2	2	2	2	2	2	1	2	1	2	2	3
CO3	2	2	2	2	2	2	2	2	2	2	1	2	1	1	2	3
CO4	2	2	2	3	2	2	2	2	2	2	2	2	1	1	2	3
CO5	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	3
CO6	2	2	2	2	2	2	2	2	3	2	2	3	1	1	2	3

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. **Detailed Syllabus:**

• Unit 1:Sets, relations, functions, permutations, combinations. Applications torelations. Logic, Propositional Equivalences, Predicates and Quantifiers Sets, Proof Techniques, Contradiction. Mathematical induction, pigeonhole principle.

Cardinality of sets, finite and infinite sets, countable and uncountable sets, Cantors numbering.



• Unit 2:Group, subgroup/normal subgroup, homorphism/automorphism/isomorphism/ eipmorphism, kernel, cosets, quotient group, product set in a group, center of a group, order/conjugate of an element, commutator.

Coding theory (Application to group theory).

- Unit 3: Ring, Field, Finite field over a prime. Applications to finite fields.
- Unit4: Recurrence relations, generating functions, numeric functions. Applications to recurrence relations.
- Unit 5: Basics of probability theory, birthday attacks. Applications on hash functions.
- Unit6:Graphs, Adjacency, Special Graphs, Isomorphic Graphs, Paths, Cycles and Circuits,

Connected Graphs, Eulerian Graphs, Hamiltonian Graphs and Planar Graphs.

Reference Books:

1. Thomas Koshy, "Discrete Mathematics with Applications", Elsevier Press, 2004.

2. C.L.LiuandD.P.Mohapatra,"ElementsofDiscreteMathematics:AComputerOrientedApproach," Tata McGraw-Hill Edition,2017.

3. D.S.Malik and M.K.Sen,"Discrete Mathematical Structures: theory and applications, "Thomson, 2004.

- 4. Joseph Gallian, "Contemporary Abstract Algebra", 9th Edition, CENGAGE publications, 2019.
- 5. Gross and Yellen, "Graph Theory and its Applications", 2nd Edition, Chapman and Hall, 2005.
- 6. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", Sixth Edition, 2006

5. <u>Teaching-Learning Strategies in brief</u>(4 to 5sentences):

This course supports the expected characteristics, capabilities and skills for computer science graduates in the following ways:

- Mastery of Computer Science technical foundations
- Recognition of common Computer Science themes and principles
- Recognition of interplay between theory and practice
- Effective problem solving and critical thinking skills

6. <u>Assessment methods and weightages in brief(4 to 5sentences):</u>

- Assignments:10%
- In-Class Tests:20%
- Mid Semester Examination:30%
- End Semester Examination:40%

Computer Programming

Name of the Academic Program	: B.Tech in Computer Science and Engineering
Course Code	: CS1.302
Title of the Course	: Computer Programming
L-T-P	: 3-1-3.
Credits	: 5
(L = Lecture Hours, T = Tutorial Ho	ours, P = Practical Hours)

1. Prerequisite Course / Knowledge:



Logical thinking and mathematical concepts at the level of a 10+2 standard student with a math major.

No prior programming experience or computing background is required.

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Explain the syntax of programming language constructs and their semantics and describe a program structure and its execution model. (Cognitive Level: **Understand**)

CO-2: Describe the steps in program editing, compilation and execution using tools such as Visual Studio Code, GCC compiler on a Linux/Windows/MAC operating system.

CO-3: Choose appropriate primitive data types and design new composite data types to model the relevant data in a given computation problem and also discover the algorithmic logic required to solve well-defined computational problems. (Cognitive Levels: **Apply** and **Analyze**)

CO-4: Compare and contrast the performance of different algorithmic approaches for simple computational problems with respect to time and memory. (Cognitive Levels: **Analyze** and **Evaluate**)

CO-5: Write programs involving basic dynamic data structures such as linked lists and use tools such as Valgrind to detect any memory leaks. (Cognitive Levels: **Apply** and **Analyze**)

CO-6: Use debugging tools such as GDB proficiently to rapidly isolate and remove subtle/complex bugs in programs. (Cognitive Levels: **Apply** and **Analyze**)

CO-7: Manage complex large projects using source code management tools such as GIT and build tools such as Make. (Cognitive Levels: **Apply** and **Analyze**)

CO-8: Assess and evaluate the solutions of their classmates through a peer review process (Cognitive Level: **Evaluate**)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
CO 1	3	-	3	-	3	-	-	-	-	-	-	-	2	3	3	3
CO 2	3	-	3	-	3	-	-	-	-	-	-	-	2	3	3	3
CO 3	3	3	3	2	3	-	-	-	-	-	-	-	2	2	3	3
CO 4	2	3	3	3	3	-	-	-	-	-	-	-	2	2	3	3

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix



CO 5	2	3	3	3	3	-	-	-	-	-	-	-	2	2	3	3
CO 6	1	1	1	1	3	-	-	-	-	-	-	-	2	3	3	3
CO 7	3	2	2	2	3	-	-	3	3	3	-	-	2	3	3	3
CO 8	1	2	2	2	2	-	-	3	3	3	-	-	2	2	3	3

4. Detailed Syllabus

- Unit 1:
 - $\circ~$ Basic computer organization, Von Neumann architecture and stored program concept
 - High level programming languages, assemble code, binary instructions, compilers and assemblers
 - Programming editing, compilation and execution cycle
- Unit 2:
 - Use of variables as reference to memory locations
 - Basic data types and their representation
 - Operators and precedence levels, expressions
 - Writing straight-line sequence of code
 - Standard I/O Libaries
- Unit 3:
 - Conditional Statements (if-then-else) and Loops (for, while, etc.)
 - o Arrays
 - Functions and parameter passing mechanisms
 - Standard libraries for string manipulation, disk file access etc.
 - Structures, Unions and Enumerations
- Unit 4:
 - Recursion
 - Program stack, scope and lifetime of variables
 - Pointers, heap memory, dynamic memory management, linked lists and memory leaks
- Unit 5:
 - Preprocessor directives
 - Source code management tools like GIT and use of GDB for program debugging
 - Multi-file programming and Makefiles

Reference Books:

- 1. K. N. King. 2008. C Programming: A Modern Approach, Second Edition. W.W. Norton & Company.
- 2. Nick Parlante. 2017. Essential C. Independently published.

5. Teaching-Learning Strategies in brief



Lectures are conducted in a highly interactive fashion. Programming problems are solved in-class along with students in a collaborative fashion. Sometimes two-three students are given an opportunity to present their programs to the class. At the end of every class, a small homework problem which helps in enhancing the concepts discussed in the class will be released. Students need not submit this homework. Tutorial sessions are used to teach the utilization of tools such as Visual Studio Code, GCC, GDB, GIT, Makefiles, perf, valgrind etc. Lab sessions are used to solve programming assignments and teaching assistants help students in developing program logic, debugging etc. on an individual basis. Faculty conducts office hours once in week. On the rest of the days, teaching assistants conduct office hours. This ensures continuous support to students. Key milestones are defined. Feedback from the students at those milestones are taken. The provided feedback is taken to fine tune the course and provide special support to students who are lagging behind. Five to six programming assignments are designed which gives an in-depth understanding of various concepts discussed in the class and their application to new problem scenarios along with proper analysis. Some problems involve evaluating, comparing and contrasting multiple solution approaches.

6. Assessment methods and weightages in brief

- 1. Programming Assignments (5 to 6) : 50 percent
- 2. Best 2 out of 3 Programming Lab Exam: 2 x 15 = 30 percent
- 3. Best 2 out of 3 Theory $\frac{\text{Exams}}{\text{Exams}}$: 2 x 10 = 20 percent

For programming assignments and lab exams, online judges such as DMOJ are used to provide immediate feedback to students. While some test cases are revealed, others are hidden. Partial marks are allocated for code peer-reviewing in programming assignments.

Real Analysis

Course Code	: IMA.303
Title of the Course	: Real Analysis
L-T-P	: 3-1-0.
Credits	: 4
(L= Lecture hours, T=Tu	torial hours, P=Practical hours)

1. Prerequisite Course / Knowledge:

Elementary knowledge of Calculus

Much of mathematics relies on our ability to be able to solve equations, if not in explicit exact forms, then at least in being able to establish the existence of solutions. To do this requires a knowledge of so-called "analysis", which in many respects is just Calculus in very general settings. The foundations for this work are commenced in Real Analysis, a course that develops this basic material in a systematic and rigorous manner in the context of real-valued functions of a real variable..

2. Course Outcomes (COs)

On successful completion of this course, students will be able to:

CO1. describe the fundamental properties of the real numbers that underpin the formal development of real analysis;



CO2. demonstrate the knowledge of an understanding of the theory of sequences and series

CO3. demonstrate skills in constructing rigorous mathematical arguments;

CO4. apply the theory in the course to solve a variety of problems at an appropriate level of difficulty;

CO5. demonstrate skills in communicating mathematics.

CO6: analyse how abstract ideas and regions methods in mathematical analysis can be applied to important practical problems.

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and</u> <u>Program Specific Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
CO 1	2	2	2	2	3	-	1	-	3	1	3	2	3	3	2	3
$\begin{array}{c} \text{CO}\\ 2 \end{array}$	2	2	2	1	3	-	1	-	2	2	2	3	3	2	1	1
CO 3	1	1	1	1	1	1	1	-	2	2	2	1	1	1	1	1
CO 4	3	2	2	1	2	2	1	-	2	2	2	1	3	3	3	2
CO 5	3	2	2	1	2	2	1	-	2	2	2	1	3	3	3	2
CO 6	1	1	1	1	1	2	1	-	1	1	1	1	1	1	1	1

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

Unit 1 Sequence of real No, Bounded and Unbounded Sets, Supremum, Infimum, Limit points of a set, Closed Set, Countable and uncountable sets. Sequences, Limit points of a Sequence. Limits Inferior and Superior, Convergent sequence, Non convergent sequence, Cauchy General Principle of Convergence, bounded and monotone sequence, Infinite Series, Positive Term Series, Convergence of series of real numbers, Necessary condition, Absolute convergence and power series, Convergence tests for series.

(9 hours)

Unit 2 Mean value theorems (Rolle's Theorem, Cauchy Mean Value Theorem, Lagrange's Mean Value Theorem), Indeterminate forms, Taylors Series, Partial derivatives. Integration as a limit of a sum, Some integrable functions, Fundamental theorem of Calculus, Mean Value Theorems of Integral calculus, Integration by parts, Change of variable in an integral, Second Mean value theorem, Multiple integrals,

(9 hours)

Unit 3: Vector, Vector operations, Products, Areas and Determinants in 2D, Gradients, Curl and Divergence, Volumes and Determinants in space. Differential equations of first order and first degree. Linear ordinary differential equations of higher order with constant coefficients. Elements of Partial Differential Equation (PDE).

(7.5 hours)

Unit 4: Analytic function of complex variable, CR Equation, harmonic functions, Laplace equation, applications (7.5 hours);

Unit 5 Integration of a function of a complex variable, M-L inequalities. Cauchy's Integral Theorem. Cauchy's Integral formula. Taylor's and Laurent Expansion, Poles and Essential Singularities, Residues, Cauchy's residue theorem, Simple contour integrals.

(9 hours)

• A project related to the above syllabus will be done by students to be submitted by the end of the semester.

References:

• Rudin, Walter, *Principles of Mathematical Analysis*, third edition, International Series in Pure and Applied Mathematics. McGraw-Hill Book Co., New YorkAuckland-D[•]usseldorf, 1976

•

Bartle, Robert G., *The Elements of Real Analysis*, second edition, John Wiley & Sons, New York-London-Sydney, 1976. (for Fourier Series)

•

Ross, Kenneth A., *Elementary Analysis. The Theory of Calculus*, second edition, in collaboration with Jorge M. L'opez, Undergraduate Texts in Mathematics, Springer, New York, 2013.

• Kreyszig, Erwin. (1983). Advanced engineering mathematics. New York : Wiley,

• Goldberg, Richard R., *Methods of Real Analysis*, second edition, John Wiley & Sons, Inc., New York-London-Sydney, 1976.

• Churchill, Ruel V. and Brown, James Ward, *Complex Variables and Applications*, fourth edition, McGraw-Hill Book Co., New York, 1984

• Ahlfors, Lars V., *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*, third edition. International Series in Pure and Applied Mathematics, McGraw-Hill Book Co., New York, 1978.

5.Teaching-Learning Strategies in brief:

Lectures in the classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning





6. Assessment methods and weightages in brief:

Assignments in theory: 10 marks, Quizzes in theory: 10 marks, Mid Semester Examination in theory: 20 marks, End Semester Examination in Theory: 30 marks, Assessment project: 30 marks

Digital Systems and Microcontrollers

Name of the Academic Program: B.Tech in ECE

Course Code	: S21EC2.101
Title of the Course	: Digital Systems and Microcontrollers (DSM)
L-T-P	: 3-1-3
Credits	: 5
(L= Lecture hours, T=Tute	orial hours, P=Practical hours)

1. Prerequisite Course / Knowledge:

Understanding of basic algebra concepts taught up to the 10+2 level

2. <u>Course Outcomes (COs)</u>:

After completion of this course successfully, the students will be able to.

CO-1: Solve problems pertaining to the application of Boolean algebra, number systems, and simplification of logic expressions using Karnaugh maps.

CO-2: Develop a simplified combinational circuit as a solution for a given problem.

CO-3: Analyze a real-world problem to develop a digital design solution using sequential circuits to solve the problem.

CO-4: Describe the working of a basic 8-bit von Neumann architecture processor.

CO-5: Develop skills for simulating circuits using basic components on online simulation tools (example, Tinker CAD).

CO-6: Design, implement and test a given logic circuit using basic electronic components such as breadboards, ICs etc.

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes</u> (PSOs) – Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	1	2	1	1	1	1	3	1	2	1	2	1	1
CO2	1	2	3	2	1	1	1	1	1	1	1	2	2	2	1	1
CO3	1	2	3	2	1	1	1	1	1	2	1	2	3	2	1	1
CO4	2	2	1	2	1	2	1	1	1	2	2	3	3	2	3	3
CO5	1	2	3	2	3	2	1	1	3	2	3	1	2	1	3	2
CO6	1	2	3	2	3	2	1	1	3	2	3	1	3	1	3	2
.1 1	0	(TT)		1.5		• •	N C	() E 1		1 15			<u>с</u> (т		1.5	

Note: 3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. <u>Detailed Syllabus:</u>



Unit 1: Number systems and interconversions (binary, decimal, hexadecimal), postulates of Boolean algebra, binary logic gates, binary functions

Unit 2: Simplification of binary expressions using K-maps, logic function implementation, combinational circuits

Unit 3: Latches and flip-flops, types of flip-flops, internal circuit design and operation

Unit 4: Sequential circuits, state diagrams, state tables, state equations, applications of sequential circuits Unit 5: Registers and counters, memory and processor architecture

Reference Books:

1. M. Morris R. Mano and Michael D. Ciletti (2013), Digital Design, 6th Ed, Pearson.

5. <u>Teaching-Learning Strategies in brief (4 to 5 sentences):</u>

The course instruction is delivered through lectures with examples of real-world application of electronic systems to foster student understanding and interest. The course is structured as a theory and laboratory course, such that the concepts and circuits introduced in the theory classes can be experimentally applied and understood by the students. Assignments are designed to encourage students to critically think about the concepts discussed in the class and to learn to independently solve problems.

6. <u>Assessment methods and weightages in brief (4 to 5 sentences):</u>

Continuous evaluations: Assignments – 10% MCQ Quizzes – 20% Lab reports – 20%

Comprehensive evaluations: Lab exam – 15% End semester exam in Theory – 35%

Value Education-1

Name of the Academic Program	: B. Tech. in ECE, BTech in CSE
Course Code	: OC3.101
Title of the Course	: VALUE EDUCATION - I
L-T-P	: 12-6-0 (Total number of hours)
Credits: 2	

(L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. Prerequisite Course / Knowledge: -NIL-

2.Course Outcomes (COs) :

After completion of this course successfully, the students will be able to:

CO-1: Apply the basic framework of universal human values to the self.



CO-2: Look at larger issues that (for many reasons) most are not exposed to: social, political, community, family, individual, etc. in a sensitized way.

CO-3: Understand themselves and their own roles within the bigger context. What are really, truly important to them? What are made important by others?

CO-4: Engage and connect with others and nurture the relationships.

CO-5: Think to shape and change the world, and not be mere technologists or scientists.

3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	-	-	-	-	-	3	2	3	2	-	-	-	-	-	-	-
CO2	-	-	-	-	-	3	3	3	3	-	-	-	-	-	-	-
CO3	-	-	-	-	-	3	3	3	2	-	-	-	-	-	-	-
CO4	-	-	-	-	-	2	3	3	3	-	-	-	-	-	-	-
CO5	-	-	-	-	-	3	3	3	2	-	-	-	-	-	-	-

Note: '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

Unit 1: Goal in life - short term and long term goals; Basic aspirations - Happiness and Prosperity; Role of education and human conduct; Self-exploration; Developing a holistic view

Unit 2: Gratitude and the need to acknowledge one's gratefulness; Understanding Self and Other;

Unit 3: Living in harmony at 4 levels: self-self, self-family, self-society, self-nature

Unit 4: Understanding needs of body and self; Right understanding of physical facilities and relationships; Understanding human relationships; Trust and Respect - the foundational values in relationships;

Unit5: Harmony in Society; The sense of safety, justice and peace in society; Nature and Sustainability; Self-reliance and Gandhian thought

Reference Books:

1. R.R. Gaur, R. Sangal, G. P. Bagaria. 2009. A Foundation course in Human Values and Professional Ethics. Excel books, New Delhi.

2. Randy Pausch. 2008. The Last Lecture. Hachette Books.

3. E. F. Schumacher. 1973. Small is beautiful: a study of economics as if people mattered. Blond & Briggs, Britain.

4. P. L. Dhar, R. R. Gaur. 1990. Science and Humanism. Commonwealth Publishers.

5.Teaching-Learning Strategies in brief (4 to 5 sentences):

This is a discussed based course. The instructor shares information on a topic and guides the discussion in the class by asking the right questions. By keeping the objectives in mind, the instructor adopts different techniques including smaller group discussions, role-play/skit, use of video clips or images to analyse and some activities to keep the students engaged in class throughout. Talks by experts who made a difference are also organised for the batch. Field trips to farms, orphanages, old-age homes,



villages and jails are arranged as part of the induction programme, in parallel to the classes in VE for the first year UG batch.

6.Assessment methods and weightages in brief (4 to 5 sentences):

This is a Pass/Fail course. The assessment methods include submissions of assignments and term papers. Critical thinking is expected from watching relevant short films or by reading assigned books. The classroom participation is also taken into consideration for evaluation. There are a few community-based activities and projects also. Participation in them is also important.(weightage for each kind of assessment may be given.)

Linear Algebra

Name of the Academic Program	: BTech in Computer Science
Course Code	:
Title of the Course	: Linear Algebra
L-T-P	: 3-1-0
Credits	: 4

Prerequisite Course / Knowledge:

This is one of the first math courses and only assumes school knowledge of maths.

Course Outcomes (COs):

After completion of this course successfully, the students will be able to...

CO-1: Explain the basic mathematical concepts like vector space, Basis, Linear Transformation, Rank Nullity Theorem, Matrix Representation of Linear Transformations, System of Equations, Determinants.

CO-2: Demonstrate familiarity with Eigenvalues, Eigenvectors, Orthogonality and Matrix Decomposition theorems.

CO-3: Synthesize proofs of theorems related to Matrices and Vector Spaces using clear mathematical and logical arguments.

CO-4: Apply principles of Spectral Decomposition and Singular Value Decompositions to real world problems in Image Compression, Principal Component Analysis etc.

CO-5: Design dimension reduction techniques with approximation guarantees using Best Fit Subspaces.

CO-6: Create mathematical models using principles of Linear Algebra and analyze them.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10PO11PO12PSO1PSO2PSO3PSO4

CO1	2	3	1	3	1	1	1	1	2	2	1	3	1	1	1	2
CO2	1	2	2	2	1	1	1	1	2	2	1	1	3	1	1	2
CO3	2	3	1	3	1	1	1	1	2	2	1	3	2	1	1	2



CO4	1	3	2	2	1	1	1	1	2	2	1	1	3	1	2	2
CO5	1	3	2	3	1	1	1	1	2	2	1	1	3	1	1	3
CO6	1	2	1	2	2	1	1	1	2	2	1	2	3	2	2	2
'3' fo	r 'Hi	igh-le	evel'	map	ping,	2 for	· 'Me	ediun	n-leve	el' ma	apping	g, 1 fc	or 'Lo	w'-lev	vel' m	napping

Detailed Syllabus:

Unit 1: Vector spaces, subspaces, Linear dependence, Span, Basis, Dimension, Finite dimension vector spaces Linear transformation, Range and Null space of linear transformation, Rank Nullity Theorem, Sylvester's Law, Matrix representation of a linear transformation for finite dimensional linear spaces, Matrix operations, change of basis, Rank of a Matrix, Range and Null Space of a matrix representing a linear transformation. Linear spaces with inner product [inner product example over space of functions: orthogonality and orthogonal functions in L_2.

Unit 2: System of Linear Equations, Row-echelon form, reduced row-echelon form. Gauss-Jordon elimination, Solution of linear systems using Gauss-Jordon elimination, matrix inversion by Gauss Jordon elimination, Understanding Range Space and Solution Space using Rank-Nullity Theorem.

Unit 3: Eigenvalues and Inner product: Eigenvalues & Eigenvectors, Norms, Inner Products and Projections, Applications like Analysis of Random Walks.

Unit 4: Advanced Topics: Spectral & Singular Value Decomposition Theorems, Applications of SVD and Best Fit Subspaces

Reference Books:

- 1. Linear Algebra, 2nd edition, K. Hoffman and R. Kunze. , Publishers, Edition, Year
- 2. Finite Dimensional Vector Spaces, P. Halmos., Publishers, Edition, Year
- 3. Introduction to Linear Algebra, Gilbert Strang., Publishers, Edition, Year
- 4. Linear Algebra Done Wrong, Sergei Treil., Publishers, Edition, Year

Teaching-Learning Strategies in brief (4 to 5 sentences):

Lectures will initially introduce the motivations, concepts, definitions along with simpler examples. This will be followed by assignments and quizzes that will make sure that the students have understood the concepts. These will be followed by deeper lectures and assignments which lead the students to the bigger questions in the area. These will also be supplemented with real world engineering problems so that they can apply the concepts learned by them.

Assessment methods and weightages in brief (4 to 5 sentences):

- In-class Quizes: 15%
- Assignments: 15%
- Class Test 1: 10%
- Class Test 2: 10%
- Mid Semester Exam: 20%
- End Semester Exam: 30%

Introduction to IoT

Name of the Academic Program	: B-Tech in Computer Science and Engineering
Course Code	: CS3.303
Title	: Introduction to IoT



L-T-P :2-0-3 Credits :3 (L=Lecture hours, T=Tutorial hours, P=Practicalhours)

1. <u>PrerequisiteCourse/Knowledge</u>

Basic knowledge of C/C++ programming, Digital Systems and Microcontrollers

2. <u>Course Outcomes(COs)</u>

After completion of this course successfully, the students will be able to

CO-1. Explain the definition of IoT and the various IoT architectures.

CO-2. Explain the types and characteristics of commonly used sensors, actuators and microcontrollers.

CO-3. Explain the communication and application layer IoT protocols.

CO-4. Explain the concepts of Cloud+Fog Computing, IoT Interoperability, data handling and analytics.

CO-5. Employ the Arduino Programming concepts to program microcontrollers.

CO-6. Employ the interfacing of sensors and actuators with microcontroller.

CO-7. Employ a few communication and application layer protocols.

CO-8. Employ an Interoperability standard called oneM2M.

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program</u> <u>Specific Outcomes (PSOs) – Course ArticulationMatrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	1	1	1	-	1	-	-	-	2	-	3	1	1	-	-
CO2	1	1	1	1	-	1	-	-	-	2	-	3	1	1	-	-
CO3	1	1	1	1	-	1	-	-	-	2	-	3	1	1	-	-
CO4	1	1	1	1	-	1	-	-	-	2	-	3	1	1	-	-
CO5	2	1	-	2	3	1	-	-	3	1	-	3	1	1	-	-
CO6	2	1	-	2	3	1	-	-	3	1	-	3	1	1	-	-
CO7	2	1	-	2	3	1	-	-	3	1	-	3	1	1	-	-
CO8	2	1	-	2	3	1	-	-	3	1	-	3	1	1	-	-

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs.

Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. <u>DetailedSyllabus</u>

Unit 1: Introduction – Definition, Architectures and Use Cases

Unit 2: Sensor and Actuators – Definition, features, classification, characteristics, physics of few basic and important sensors and actuators

Unit 3: Microcontroller and Programming -



- Basics of a controller, popular microcontrollers
- Microcontroller programming(Arduino/ESP32)
- Overview of different peripherals: ADC, DAC, Memory, GPIO, Timers
- Interfacing of Sensors and Actuators to microcontrollers: UART, SPI,I2C

Unit 4: Communication Protocols –

- Basics of communication network
- Overview of different communication technologies for IoT: LoRa WAN, Cellular (3G/4G/5G), WLAN, Bluetooth, Zigbee
- Overview of application/middleware protocols: MQTT, HTTP, CoAP
- Connecting the sensor node to the internet

Unit 5: Cloud + Fog Computing – Characteristics, Types of Cloud, Challenges in Cloud Computing, Fog Computing Architecture, Advantages of Fog Computing, Case Study

Unit 6: Interoperability -

- Concepts and Types of Interoperability
- Interoperability Standards and one M2M

Unit 7: Data Handling and Analytics -

• Handling - Definition, Data Types, Characteristics of Big Data, Data Flow (Generation, Acquisition, Storage, Analysis)

• Analytics - Definition, Types of Analytics (Descriptive, Diagnostic, Predictive,

Prescriptive), Qualitative and Quantitative Analysis

Reference Books:

1) Perry Lea, (2018) Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security, Packt Publishing

5. <u>Teaching-Learning Strategies in brief</u>

Weekly lectures cover the theory in the syllabus and the labs will deliver the hands-on experience in building IoT systems. The comprehensive quizzes and end semester exam will test the students on the relevant theory taught for IoT systems. The project will give the students an end-to-end IoT system development covering all the concepts learned in the labs.

6. Assessment methods and weightages inbrief

Type of Evaluation	Weightage (in %)
Quizzes	15
End Sem Exam	30
Labs	30



Project

25

Data Structures and Algorithms

Name of the Academic Program	: B.Tech in Computer Science and Engineering
Title of the Course	: Data Structures and Algorithms
L-T-P	: 3-1.5-3.
Credits	: 4
(L = Lecture Hours, T = Tutorial Hours.	P = Practical Hours)

1. Prerequisite Course / Knowledge:

CS1.302 - Computer Programming

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Explain the design and implementation details of fundamental data structures and sorting/searching algorithms. (Cognitive Level: Understand)

CO-2: Write programs involving fundamental data structures and sorting/searching algorithms (Cognitive Levels: Apply and Analyze)

CO-3: Compare and contrast the performance of different data structures and sorting/searching algorithms with respect to time and memory. (Cognitive Levels: Analyze and Evaluate)

CO-4: Discover the algorithmic logic and new composite data structures required to solve well-defined computational problems while following specified compute constraints. (Cognitive Levels: Apply and Analyze)

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CO2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CO3	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CO4	3	1	3	3	1	1	1	1	3	1	1	1	1	1	1	1

4. Detailed Syllabus

- Unit-1
 - Recap: Array, Pointers, Structures, Asymptotic Complexity
 - Abstract Data Types
- Unit-2: Linear Data Structures
 - Linked Lists
 - Stacks
 - o Queues
- Unit-3: Non-linear Data Structures
 - o Binary Trees and Search Trees
 - Hash Tables, Sets, Maps
- Unit-4: Sorting Algorithms



- Sorting Insertion
- Sorting Selection, Merge, Quicksort
- Heapsort
- Counting Sorts
- Radix Sort, External Sorting
- Sorting External, Selection Algorithms
- Selection Algorithms
- Unit-5: Graph Algorithms
 - Graphs Representation and Algorithms
 - Graphs Representation and Algorithms (DFS, Dijkstra, Bellman)
 - Graphs Representation and Algorithms (MST)
 - o Graphs Strongly Connected Components
- Unit-6: Advanced Data Structures
 - AVL Trees
 - Suffix Trees

Reference Books:

1. Data Structures and Algorithm Analysis in C (M.A. Weiss), Pearson

5. Teaching-Learning Strategies in brief

Lectures are conducted in a highly interactive fashion. The design and implementation of data structures and sorting/searching algorithms is done as an in-class coding exercise. Tutorial sessions are used to teach the utilization of tools such as Visual Studio Code, Git etc. Lab sessions are used to solve programming assignments and teaching assistants help students in developing program logic, debugging etc. on an individual basis. Faculty conducts office hours once in week. Additionally, teaching assistants conduct office hours. This ensures continuous support to students. Five to six programming assignments are designed which gives an in-depth understanding of various concepts discussed in the class and their application to new problem scenarios along with proper analysis. Some problems involve evaluating, comparing multiple solution approaches.

6. Assessment methods and weightages in brief

- 1. Programming Assignments (5): 40%
- 2. Programming Lab Exam: 15%
- 3. Best 2 out of 3 Theory Exams: 30%
- 4. Mini Project (4 members per team): 15%

For programming assignments and lab exams, online judges such as DMOJ are used to provide immediate feedback to students. While some test cases are revealed, others are hidden. Partial marks are allocated for code peer-reviewing in programming assignments. For mini project, a presentation followed by a code-execution demonstration is used for evaluation.

Introduction to Software Systems

Name of the Academic Program	: Bachelor of Technology in Computer Science and Engineering
Course Code	: CSE
Title of the Course	: Introduction to Software Systems
L-T-P	:1-0-3
Credits	:2
(L= Lecture hours, T=Tutorial hour	rs, P=Practical hours)

Prerequisite Course / Knowledge: Not applicable.

2. <u>Course Outcomes (COs)</u>(5to8fora3or4creditcourse):

After completion of this course successfully, the students will be able to...



CO-1: Demonstrate familiarity with various OS Concepts, Shell programming, Web Technologies, Database Systems, Python Programming, and software engineering principles.

CO-2: Explain the different types of tools and technologies that are suitable for solving different software problems

CO-3: Apply tools and technologies to implement simple software solutions

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program</u> <u>Specific Outcomes (PSOs)–Course Articulation Matrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2								2							
CO2	3		2		2				2	2		1				
CO3	3			1	3				3		1					

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs.Write'3'in the box for 'High-level' mapping, 2for'Medium-level mapping, 1for'Low'-level mapping

4. **Detailed Syllabus:**

Unit 1: Software and Systems overview - SHELL: OS concepts, Kernel, Memory, Shell basics, Advance Linux commands including file management and schedulers, Control flows, Regex, Awk,

Unit2: Developing web applications-Introduction to HTML, CSS, and Javascript concepts, Datatypes, variables, operators, conditions, loops, functions, function expressions, events, form controls, data structures, javascript libraries, AFrame, Three.js

Unit 3: Programming with Python – Functions, Exceptions, Error Handling, Sequences, scoping rules, closures, higher-order functions, mutability, object model and inheritance, modules and packages, variable args, decorators, usage of libraries including SOAP and REST API, Flask based server set up.

Unit4: SDLC and Databases–SDL Cconcepts, Version Control Systems, Editors, Bugtrackers, Basics of SQL, CRUD;

Reference Material/Books:

1. Mastering Linux Shell Scripting : A practical guide to Linux command-line, Bash scripting, and Shell programming, by Mokhtar Ebrahim, Andrew Mallett. 2ndEdition, 2018.ISBN-13 :978-1788990554

2. Learning Python: Powerful Object-Oriented Programming, by Mark Lutz. 5thEdition, 2013.ISBN-13: 978-1449355739

3. JavaScript: The Definitive Guide, by David Flanagan. 7thEdition, 2020.ISBN-13: 978-1491952023

4. Workbook/Gitbook created by the course instructors (<u>https://serciiit.gitbook.io/introduction-to-software-systems</u>/)

5. <u>Teaching-LearningStrategiesinbrief</u>(4to5sentences):

The course is delivered using problem based learning methodology. The major goal of the course is to introduce the students to various software and systems technologies and tools that can facilitate them to develop simple



software systems. To achieve this goal, the course is delivered as a combination of lectures and tutorial sessions that provide students with hands-on experience in understanding the problem and implementing solutions using the corresponding software technologies and tools.

6. <u>Assessmentmethodsandweightagesinbrief</u>(4to5sentences):

Mid Semester Exam – 15% End semester Exam – 20% Assignments (3) – 25 % Labs (4 tests) – 20% Others– 20% (In-class Activities, Surprise quiz/test)

Computer Systems Organization

Name of the Academic Program	: B.Tech in Computer Science and Engineering
Course Code	: CS2.201
Title of the Course	: Computer Systems Organization
L-T-P	: 3-1-0.
Credits	:4
(L = Lecture Hours, T = Tutorial Hou	rs, P = Practical Hours)

1. Prerequisite Course / Knowledge:

Digital logic circuits and design. Combinational and Sequential Circuits. Fundamentals of Programming.

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1:Explain the Von Neumann Model of Computing. Describe all the steps involved in the execution of a program: composition, compilation, assembly, linking, loading and hardware interpretation of the program instructions. (Cognitive Level: Understand)

CO-2:Describe the instruction set architecture design principles. Show how programming language constructs can be mapped to sequences of assembly language instructions. Analyze and assess any given ISA. (Cognitive Levels: Analyze and Evaluate)

CO-3: Describe processor design architectural approaches. Compare and contrast sequential designs with pipelined designs. Propose new architectural approaches to optimize on performance and hardware costs (Cognitive Levels: Apply, Analyze and Create)

CO-4:Describe the basic functionality of an operating system. Clearly explain the system call interface, its design and implementation. Build systems akin to a bash shell, file server etc. using system calls. (Cognitive Levels: Understand and Apply)

CO-5:Describe the basics of process control and management. (Cognitive Levels: Understand and Apply)



CO-6:Describe the principles of virtual memory management. Analyze various memory management schemes for process isolation and physical memory utilization across multiple processes (Cognitive Levels: Understand, Apply and Analyze)

3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
CO 1	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3
CO 2	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3
CO 3	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3
CO 4	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3
CO 5	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3
CO 6	3	3	2	2	2	-	-	-	2	2	1	1	3	2	3	3

4.Detailed Syllabus

- Unit 1:
 - o Basic computer organization, Von Neumann architecture and stored program concept
 - o High level programming languages, assemble code, binary instructions, compilers and assemblers
 - o Programming editing, compilation and execution cycle
- Unit 2:
 - o Instruction Set Architecture Design Principles
 - o CISC vs RISC ISAs
 - o Binary encoding of the instructions
 - o Mapping language constructs such as expressions, if-then-else statements, loops, functions to assembly code
 - o Machine representation of numbers
- Unit 3:
 - o Processor design fundamentals
 - o ALU Design
 - o Single Cycle and Multi Cycle Processor Design
 - o Pipelined Architectures
 - o Hazards in Pipelined Architectures and approaches to resolve them.
- Unit 4:
 - o Introduction to Operating Systems. Bootstrapping Process
 - o System Calls, their design, implementation and application.
 - 0
- Unit 5:
 - o Process Control and Management



- o Scheduling multiple processes on multiple cores.
- o Basics of scheduling mechanisms and policies.
- Unit 6:
 - o Physical vs Virtual Memory
 - o Process and memory isolation/protection mechanisms
 - o Virtual memory management
 - o Page replacement algorithms

Reference Books:

- 1. Computer Systems: A Programmer's Perspective. Randal Bryant and David O'Hallaron
- **2.** Computer Organization and Design. The Hardware/Software Interface. David A. Patterson and John L. Hennessy.
- 3. Operating Systems: Three Easy Pieces by Remzi H. Arpaci-Dusseau and Andrea C. Remzi H. Arpaci-Dusseau

5.Teaching-Learning Strategies in brief

Lectures are conducted in a highly interactive fashion. Use of various system tools such as compilers, assemblers, loaders, linkers, simulators etc. are demonstrated live in the class. Assignments include assembly language programming, digital system design exercises such as Arithmetic and Logic Unit Design, programming using system calls. Most of the ideas introduced in the class are emphasized through these assignments. Teaching Assistants and Faculty conduct office hours every day. Thus students have continuous access to resources to get their doubts clarified and seek any extra help that is required. Some times students are encouraged to come to the board and explain the novel design ideas they came up with while solving assignments or mini-projects.

6.Assessment methods and weightages in brief

- **1.** Programming Assignments (5 to 6) :25 percent
- 2. Two Quizes: 2 x 10 percent
- **3.** Mid Term: 20 percent
- 4. Final Exam: 35 percent

Probability and Statistics

Name of the Faculty	: Pawan Kumar
Course	: CSE
Name of the Academic P	rogram : B.Tech. in Computer Science and Engineering
Course Code	: MA6.101
Title of the Course	: Probability and Statistics
L-T-P	: 3-1-0.
Credits	: 4



(L= Lecture hours, T=Tutorial hours, P=Practical hours)**1. Prerequisite Course / Knowledge:**

Linear Algebra, Real Analysis

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to -

CO-1. Explain theaxioms of probability and rules, discrete and continuous random variables.

CO-2. Derive the density function of transformations of random variables and use these to generate data corresponding to various distributions.

CO-3: Derive marginal and conditional distributions of multivariate random variables and probability bounds.

CO-4. Discuss the classical and Bayesian inference theory and applications.

CO-5. Discuss the basic random processes and theirapplications.

CO-6. Outline a proof of stated theorem and writethelogically derived proof.

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program</u> <u>Specific Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
CO 1	1	3	1	1	-	-	-	-	2	2	-	3	3	-	-	3
CO 2	1	3	1	3	1	-	-	-	2	2	-	3	3	-	-	3
CO 3	1	3	1	3	1	-	-	-	2	2	-	3	3	-	-	3
CO 4	1	3	2	3	1	-	-	-	2	2	-	3	3	-	-	3
CO 5	2	3	2	3	1	-	-	-	2	2	-	3	3	-	-	3
CO 6	1	3	1	1	-	-	-	-	2	2	-	3	3	-	-	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

Unit 1: Discrete Random Variables:Probability mass function, expectation and variance. Bernoulli distribution, Geometric distribution, Binomial distribution. Derivation of mean and variance. Sampling with and without replacement. Hypergeometric distribution and Poisson distribution. Derivation of the Poisson distribution as limiting form of Binomial. Derivation of mean and variance. Bounding



probabilities, tail sum formula, Markov's inequality and Chebyshev's inequality. Probability generating functions and moment generating functions. (9 hours)

Unit 2: Continuous Random Variable: Probability density function, cumulative distribution function, expectation, mean and variance. Moment generating functions and uniqueness theorem. Chebyshev's inequality. The uniform distribution on (a, b), the normal distribution. Mean and variance of the normal distribution. The Cauchy distribution. The exponential distribution, moments, memoryless property, hazard function. Gamma distribution, moments, Chi-square distribution. (9 hours)

Unit 3: Multivariate Distributions: Cumulative distribution function method for finding the distribution of a function of random variable. The transformation rules. Discrete bivariate distributions, marginal and conditional distribution and multinomial distribution. Continuous bivariate distributions, marginal and conditional distributions, independence of random variables. Covariance and correlation. Mean and variance of linear combination of two random variables. The joint Moment generating function (MGF) and MGF of the sum. The bivariate normal distribution, marginal and conditional expectation and variance, joint MGF and marginal MGF. Linear combinations of independent random variables. Means and variances. Sequences of independent random variables and the weak law of large numbers. The central limit theorem, normal approximation to the binomial distribution. (9 hours)

Unit 4: Statistical Inference and Random Processes: Point Estimation, Interval Estimation, Hypothesis Testing, Linear Regression, Bayesian Inference. Point Processes, Poisson Process, Discrete Time Markov Chains. (9 hours)

References:

- Sheldon Ross, Introduction to Probability Models, Academic Press, 2010.
- Online resource: <u>https://www.probabilitycourse.com/</u>
- A first course in Stochastic Models, H. C. Tijms, Wiley, 2003.
- Introduction to Probability and Statistics for Engineers and Scientists, S. Ross, 5th Edition, Academic Press, 2014.

5.Teaching-Learning Strategies in brief:

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing one mini-project.

6. Assessment methods and weightages in brief:

Assignments in theory: 15 marks, Mid Semester Examination-1: 25 marks, Mid Semester Examination-2: 30 marks, End Semester Examination: 30 marks

Data and Applications

Name of the Academic Program: B.Tech. in Computer Science and Engineering Course Code : CS4.301



Title of the Course: Data and Applications [Half]L-T-P: 3-1-0.Credits: 2(L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. Prerequisite Course / Knowledge:

Data Structures

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to -

CO-1. State data requirements for an application.

CO-2. Develop a conceptual model (such as, Entity Relationship Model and Diagram) for a set of data requirements.

CO-3: Comprehend relational data model and integrity constraints, and relational database design with normalization.

CO-4. Map the conceptual model to a relational data model and create and populate its corresponding relational database

CO-5. Map user queries into correct relational algebra, Structured Query Language (SQL), and tuple relational calculus expressions/statements. And updates using SQL.

CO-6. Implement an application to access, query and update a relational database.

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and</u> <u>Program Specific Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
CO 1	3	3	3	3	1	2	1	2	3	3	3	3	3	3	1	1
CO 2	3	3	3	1	3	-	-	-	3	3	3	3	3	3	1	1
CO 3	3	2	2	1	2	-	-	-	1	1	1	3	3	3	2	3
CO 4	3	3	3	3	1	2	1	2	3	3	3	3	3	3	1	3
CO 5	3	3	3	3	1	2	1	2	3	3	3	3	3	3	1	3
CO 6	3	3	3	3	1	2	1	2	3	3	3	3	3	3	1	3


'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping **4. Detailed Syllabus:**

- Unit 1: Data, Database, Database System (3 hours)
- Unit 2: Data models, Conceptual Data Modeling, ER Models (5 hours)
- Unit 3: Relational Data Model, Relational Algebra, Tuple Relational Calculus (6 hours)

Unit 4: SQL, Constraints, Triggers, Database Connectivity, Applications (3 hours)

Unit 5: Normalization, Relational Database Design (4 hours)

• Four mini projects related to the above syllabus will be done by students.

References :

- Elmasri & Navathe, Fundamentals of Database Systems, 6th Edition, Pearson Education, 2013
- Abraham Silberschatz, Henry F.Korth, S.Sudarshan, Database system concepts, fifth edition, Mc Graw Hill, 2006

5.Teaching-Learning Strategies in brief:

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing four mini-projects.

<u>6. Assessment methods and weightages in brief</u>:

Assignments in theory: 10 marks, Quizzes in theory: 10 marks, Mid Semester Examination in theory: 20 marks , End Semester Examination in Theory: 30 marks, Assessment of four mini projects: 30 marks

Automata Theory

Name of the Academic Program: B.Tech. in Computer Science and Engineering Course Code : CS1.302

Title of the Course: Automata TheoryL-T-P: 3-1-0.Credits: 2(L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. Prerequisite Course / Knowledge: Data structures, Elementary Formal Logic <u>**2. Course</u></u> Outcomes (COs)**</u>

After completion of this course successfully, the students will be able to



CO-1. Develop an understanding of the core concepts of Automata theory such as Deterministic Finite Automata, Non-deterministic Finite Automata, Regular Languages, Context Free Languages, Push down Automata, the basics of Turing Machines

CO-2. Design grammars and automata for different languages

CO-3. Identify formal language classes and prove language membership properties

CO-4. Describe the limitations of the different computational models

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and</u> <u>Program Specific Outcomes (PSOs) – Course Articulation Matrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	1	1	1	1	-	-	2	2	1	2	3	1	1	3
CO2	2	2	3	1	2	-	-	-	2	2	1	2	3	1	1	3
CO3	2	2	3	1	1	-	-	-	2	2	1	2	3	1	1	3
CO4	1	2	2	1	1	1	-	-	2	2	2	1	2	1	1	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. **Detailed Syllabus:**

Unit 1: Introduction, Finite State Machines, Deterministic Finite Automata (DFA), Non deterministic Finite Automata (NFA), Equivalence of NFA and DFA, Regular Expressions, Regular Languages, Closure

properties of regular languages, Pumping Lemma, Grammars, Left and Right-linear grammars

Unit 2: Context Free Grammar (CFG), Chomsky Normal Form, Push Down Automata (PDA), Equivalence of CFG and PDA, Context Free Languages (CFL), Deterministic PDA and Deterministic CFL, Pumping Lemma for context free languages

Unit 3: Introduction to Turing machines, Total Turing Machines, Recursive languages, Recursively enumerable languages, The Halting problem.

References:

- M. Sipser, Introduction to the Theory of Computation, Third Edition, Cengage Learning 2012.
- J.E. Hopcroft, R. Motwani and J. Ullman, Introduction to Automata Theory,

Languages and Computation, Third Edition, Pearson, 2006.



5. <u>Teaching-Learning Strategies in brief:</u>

The lectures will be arranged in a manner that facilitates inter-student and faculty-student discussions. Additionally, the lectures will have small exercises that will ensure that the students actively participate in the learning activity and think out of the box. There will be more emphasis on ideas and reproduction of textbook material. There will be small homework problems that would help the student to re-engage with the essential components of the lecture. Assignments will test the student's ability to apply key concepts learnt, and also inform the faculty of the progress being made by the students in acquiring them.

6. Assessment methods and weightages in brief :

Homework: 25% Quiz 1: 20% Quiz 2: 20% Final exam: 35%

Algorithm Analysis and Design

Title of the course	: Algorithm Analysis and Design
Faculty Name	: Suryajith Chillara
Course Code	: CS1.301
Credits	: 4
L - T – P	: 3-0-0

(L - Lecture hours,

T-Tutorial hours, P - Practical hours)

Semester, Year:	Monsoon 2022
(Ex: Spring, 2022)	

Name of the Program

Pre-Requisites : Discrete Mathematics, and Data Structures and Algorithms Course Outcomes :

After completion of this course successfully, the students will be able to...

B.Tech

CO-1: Demonstrate the ability to fully understand the analysis of various known algorithms.

CO-2: Identify problems where various algorithm design paradigms can possibly be applied.

CO-3: Understand the notions of computational intractability and learn how to cope with hardness.

CO-4: Understand the notion of approximation and randomized algorithms. If time permits, intro to quantum algorithms.



Detailed syllabus:

- **1.** Basic graph algorithms
- 2. Greedy algorithms
- **3.** Divide and Conquer
- **4.** Dynamic Programming
- **5.** Network flows
- **6.** NP and computational intractibility
- 7. Intro to Approximation and Randomized algorithms
- 8. Intro to Quantum algorithms

Assessment method and Grading scheme:

- Deep quizzes 1 and 2: 10 + 10 = 20%
- Mid-semester exam = 20%
- End-semester exam = 30%
- In-class quizzes (unannounced) = 15%
- Assignments = 15%

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	Р О 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	PO 10	PO 11	PO 12	PS O 1	PS O 2	PS O 3	PS O 4
CO 1	1	2	2	2	1	1	1	2	1	1	1	2	3	1	1	3
CO 2	2	3	2	2	1	1	1	2	2	2	1	3	3	2	2	3
CO 3	2	3	1	2	1	1	1	2	2	2	1	3	3	2	2	3
CO 4	2	3	1	2	2	1	1	2	2	1	1	3	3	1	1	3

Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs). Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'- level' mapping

Teaching-Learning Strategies in brief (4-5 sentences) :

The course lectures will include activities that promote the understanding of the lecture content by using small examples that students work out during the class itself and promote active and participatory learning.



A good part of the lecture will involve problem solving and finding solutions to problems rather than expositing known material. In class tests that are held periodically are useful as summative assessments. Homework assignments are designed to reiterate the material covered in class lectures

and also solve problems that are based on simple extensions of concepts described in the lectures.

Operating Systems and Networks

Name of the Academic Program	: B.Tech. in CSE
Course Code	: CS3.301
Title of the Course	: Operating Systems and Networks
L-T-P	: 3-1-1.
Credits	: 4
(L= Lecture hours, T=Tutorial hour	s, P=Practical hours)

1.Prerequisite Course / Knowledge:

Programming languages, Digital Logic Design, Computer Organization

2.Course Outcomes (COs)

After completion of this course successfully, the students will be able to,

CO-1. Extend the concepts of layering and modularity to build new software systems

CO-2. Develop appropriate scheduling/synchronization/memory management/ virtual memory/protection module for a new task-specific operating system.

CO-3: Implement an application on the top of given operating system in an efficient manner based on process and thread framework available in the given operating system.

CO-4. Architect the given system on the top of operating systems by exploiting the system calls of the given operating system services as far as possible.

CO-5. Develop a network-based application by exploiting networking related system calls.

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> Outcomes (PSOs) – Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1	2	-	-	-	3	2	2	3	3	2	1	2
CO2	3	3	3	2	2	-	1	-	2	2	2	3	2	1	2	2
CO3	3	3	3	2	2	2	1	-	3	2	2	3	3	2	1	2
CO4	2	2	3	2	2	3	2	-	3	2	2	2	3	3	2	3

THINK UNIT
INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

CO5	3	2	1	1	2	-	-	-	3	2	2	3	3	2	1	2

Note '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

Unit 1: Introduction, Process and thread management (9 hours);

Unit 2: CPU scheduling, Process Synchronization, Deadlocks (12 hours);

Unit 3: Memory management, Virtual memory (9 hours);

Unit 4: File systems, Protection and Security (6 hours);

Unit 5: Networking (9 hours);

• Five mini projects related to the above syllabus will be done by students in the laboratory

Reference Books:

1. Silberschatz, A, Galvin, P, Gagne, G. Operating system concepts, Addison-Wesley, 2018

2. Computer Networks (5th Edition) Andrew S. Tanenbaum, David J. Wetherall Prentice Hall, 2013

3. Wlliam Stallings, Operating Systems, Prentice-Hall, 2018.

4. Charles Crowley, Operating systems: A design oriented approach, Tata McGraw-Hill, 2017.

5. Tanenbaum, A., Modern Operating Systems, Prentice-Hall, Second Edition (latest edition, 2015.

5. Teaching-Learning Strategies in brief

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing 5 mini-projects in laboratory by the students

6.Assessment methods and weightages in brief

Two Class Room tests: 10 marks; Mid Semester Examination in theory: 20 marks, End Semester Examination in Theory: 40 marks, Assessment of 5 mini projects in Laboratory: 30 marks

Science-1

Name of the Academic Program: B. Tech. (CSE)Course Code:Title of the Course: Science IL-T-P: 3-1-0(L= Lecture hours, T=Tutorial hours, P=Practical hours)Credits: 4

1.Prerequisite Course / Knowledge: NA



<u>2.Course Outcomes (COs)</u> (5 to 8 for a 3 or 4 credit course): Outcomes of the Second Half (Introduction to Biology):

After completion of this course successfully, the students will be able to

CO-1: Analyse the aims, methodology of science and technology, and their impact on society

CO-2: Explain Special Theory of Relativity and compute its consequences for typical scenarios of relevance

CO-3: Demonstrate familiarity with Lagrangian and Hamiltonian formulations of mechanics, by formulating the

equations of motion from basic principles for mechanical systems

CO-4: Explain connections between thermodynamics and statistical mechanics and their use in modern chemical computations

C0-5: Infer the stability of molecules using the concepts of hybridization and molecular orbital theory

CO-6: Recognize the role of symmetry in nature

CO-7: Demonstrate problem solving skills upto a level that allows application to research topic of their interest

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	1	2	2	2	2	1	1	2	1	2	2	1	1	1
CO2	2	2	2	2	1	1	2	1	1	1	1	2	0	3	2	1
CO3	2	2	1	2	1	1	2	1	1	1	1	2	0	3	2	1
CO4	2	2	1	2	1	1	2	1	1	1	1	2	1	3	2	1
CO5	2	2	1	2	1	1	2	1	1	1	1	2	1	3	2	1
CO6	2	2	1	2	1	1	2	1	1	1	1	2	2	2	2	2
CO7	2	2	1	2	1	1	2	1	1	1	1	2	1	1	1	2

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

Unit 1: Mathematical modeling in sciences, (i) geometry and linear algebra, (ii) change and calculus and (iii) chance and probability. Simple models can have complicated behavior: logistic map demonstrates deterministic chaos

Unit-2: Forms in nature. Scales of length, time and energy in nature.

Unit 2: Special theory of relativity: postulates, Lorentz Transformation, Length Contraction, Time dilation, Doppler effect, relative velocity determination, twin paradox, relativistic momentum and energy. Space time graphs, and relativity of simultaneity.



Unit 3: Review of Newtonian Mechanics and its difficulties / failures. Introduction to Lagranian and Hamiltonian formulations, and application to mechanical problems.

Unit 5: Need for Quantum Mechanics. Schrodinger equation for time-dependent and time-independent scenarios. Application to atoms and molecules; provide qualitative picture of orbital hybridization to explain the molecular structures

Unit 6: Review of Thermodynamics and introduction Statistical Mechanics and applications to problems of relevance. Lasers. Fermi statistics, band-gap in semiconductors.

Reference Books:

- 1. "Concepts in Modern Physics" A Beizer (6th Edition)
- 2. "Classical dynamics of particles and systems" by Stephan Thornton and Jerry Marion (5th editioon)
- 3. "Physical Chemistry" Atkins and de Paula (8th Edition)

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

The objective of the course is to give the CSE/ECE students a good understanding of the concepts in Modern Physics and modern chemistry. To familiarize the students with available web-based resources, and problem solving (whenever possible with scientific programming).

6.Assessment methods and weightages in brief (4 to 5 sentences):

Assignments – (20%), Class notes (10%) Preannounced and surpise In-class quizzes (25%), End semester exam (35%)

Embedded Systems Workshop

Name of the Academic Program	: B.Tech. in Computer Science and Engineering
Course Code	: EC3.202
Title of the Course	: Embedded Systems Workshop
L-T-P	: 2-0-3-4
Credits	: 4
(L=Lecture hours, T=Tutorial hours, P=	Practical hours)

<u>1. Prerequisite Course /Knowledge:</u>

10+2 level physics

CS0.101: Computer Programming CS3.303: Introduction to IoT

2. Course Outcomes(COs)



After completion of this course successfully, the students will be able to

(Create) CO-1: Develop and implement an IoT-based solution for a real-life problem (Evaluate) CO-2: Assess system designs from IoT application point of view

(Understand) CO-3: Explain the working on microcontrollers, peripherals and its programming (Analyze) CO-4: Compare and select the sensors and actuators based on the system requirement (Analyze) CO-5: Compare different communication protocols for use in IoT systems

(Apply) CO-6: Employ techniques pertaining to the security, privacy and inoperability of IoT data (Analyze) CO-7: Examine various available solutions for data storage and cloud computing

(Create) CO-8: Design and fabricate a functional PCB and mechanical enclosure for their IoT project

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and</u> <u>Program Specific Outcomes (PSOs) – Course ArticulationMatrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	2	3	2	2	3	2	2	3	1	1	3	1	3	3	2
CO2	2	3	3	3	2	2	1	2	2	1	3	2	1	3	2	1
CO3	2	3	1	1	1	1	2	1	1	3	2	1	3	1	3	3
CO4	3	3	1	1	1	3	3	1	1	3	1	1	3	1	2	3
CO5	2	1	2	3	3	2	1	1	3	1	2	2	1	2	2	1
CO6	2	3	2	2	2	3	2	3	1	2	2	3	3	3	2	2
CO7	3	2	2	3	1	2	2	2	1	2	2	2	3	3	2	2
CO8	1	2	3	1	3	2	2	1	3	2	3	1	1	2	1	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

Detailed Syllabus:

1. Sensing/Actuators and interfacing

- 1. Sensor/Actuator selection (using datasheets)
- 2. Physics of sensors and actuators related to projects
- 3. Interfacing: Serial interfaces, Analog out, SPI, UART, I2C, "propriety" such asDHT22

2. Controller, Embedded Systems and Peripherals-

1. Platform selection – ATMEL328, ESP32, STM8 Architecture; timers, interrupts, AVR, SAMRarchitectures



- 2. Embedded Systems: power management, interrupts, memory management, leaks, OTA firmware update, reliability, onboard debugging
- **3.** Peripherals: RTC, ADC channels, resolution, onboard memory, power, external/internal watchdog

3. Communications, Networking and IoT Architecture

1. Different IoT communication protocols: Comparison ofZigbee/WiFi/BLE/4G/5G/eSim/LoRaWAN

2. Data Protocols:MQTT/HTTPS/CoAP

4. Data Storage and Computation

- 1. Cloud storage and computing
- 2. Data retrieval optimization
- 3. IoT standards for interoperability: Implementation usingoneM2M

5. PCB and Enclosure Design

- 6. Data privacy and security
- 7. Dashboard and visualization

1. Software/ Approaches : UI/UX and Time Series Data Visualization; Front-end and backend technologies

8. Documentation

- 1. User document and developer's documentation
- 2. Best practices for writing the two documents
- 3. Referring style manual. For example, *Microsoft/Chicago manual of style*

Reference:

- 1. Raj Kamal, Internet of Things, McGraw Hill, 2018
- 2. P. Lea, Internet of Things for Architects, 2018
- 3. O. Hersent, D. Boswarthick, O. Elloumi, The Internet of Things, Wiley, 2016
- 4. D. Norris, The Internet of Things, McGraw Hill, 2015
- 5. A. Bahga and V. Madisetti, Internet of Things, University Press, 2016

<u>5.</u> <u>Teaching-Learning Strategies inbrief</u>:

Lectures will be integrating ICT into classroom teaching, active learning by students, and projectbased learning by doing an IoT-based project.



6. Assessment methods and weightages in brief:

Quiz	10%
EndSem	30%
Project	60%

Sports-3

Design and Analysis of Software Systems

Name of the Academic Program: Bachelor of Technology in Computer Science and Engineering

Course Code: Title of the Course: Design & Analysis of Software Systems

L-T-P: Credits: 4

(L=Lecture hours, T=Tutorial hours, P=Practical hours)

1.Prerequisite Course / Knowledge: Intro to Software Systems

2.Course Outcomes (COs) (5 to 8 for a 3 or 4 credit course):

After completion of this course successfully, the students will be able to...

CO-1: Understand the process of building software, through a live project

CO-2:Inculcate software engineering knowledge, skills, and technologies needed to build software

CO-3: Understand the structured approach and disciplined process (iterative) to develop software

CO-4: Learn the steps in building a reasonably complex piece of usable that is maintainable

CO-5: Enhance written and oral communication skills, needed for software engineering

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes</u> (PSOs) – Course Articulation Matrix

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
CO 1	2		2	1					2		1	1		1		
CO 2	3		1		2						1		1	3		
CO 3	2	1	1								1		1	2		1



CO 4	2	2	2			3			2	1	
CO 5							3				

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

The course will be run as units, following typical agile development sprints

- **1.** Introduction
 - a. Introduction to Software Engineering
 - b. Development Lifecycle, Process Model Traditional Vs Agile processes.
 - **c.** Project and Team Management Project organization concepts (roles, tasks, work products),
- 2. Requirements
 - a. Analysis and Specification),
 - b. Estimation, Release Planning, Organizational activities (communication, status meetings).
- 3. Design
 - a. Modelling (UML), Architecture and Design,
 - b. System Decomposition, Software Architectural styles, Documenting Architectures,
- 4. Testing
 - **a.** Quality Assurance Unit, Integration, System and Acceptance Testing, Introduction to various testing techniques (e.g. Stress testing),
- 5. Design Patterns
 - a. Design patterns, UI design
 - **b.** Software Development for startups



Reference Books:

Software Engineering - A Practitioner's Approach, 10th Edition, Roger Pressman.

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

The proposed course provides an introduction to software engineering concepts and techniques to undergraduate students using project based methodology. Students work in a small teams to deliver a software system that are proposed by real industrial clients. The course content and project introduces various software technologies, process and project management skills that are needed for the delivery of software in a team setting.

<u>6.Assessment methods and weightages in brief</u> (4 to 5 sentences):

Component	Percentage (%)
Project	40
Client Feedback (R1 1% + R2 3%)	4
Coding Assignments (4)	20
Quizzes (Q1 + Q2, no midterm)	12
Class submissions (3 Questions)	4
Class Assignments	8
End Exam/Research Paper	12
TOTAL	100



CS Program Outcomes (POs)

PO1 Engineering knowledge: Use concepts from varied disciplines including Computer Science, Electronics, Mathematics, and the Sciences, to engineer and develop systems of varying scale.

PO2 Problem analysis: Identify, formulate and analyze complex engineering problems reaching substantial conclusions using first principles of Mathematics, Natural Sciences and Engineering Sciences.

PO3 Design/Development of solutions: Identify and bring to fore the necessary concepts from Computer Science and arrive at creative ways to solve problems that take into account the societal, cultural, and ethical considerations.

PO4 Conduct investigations of complex problems: Interpolate and extrapolate based on existing knowledge base and self-learning skills to investigate the dynamics of complex problems and find solutions.

PO5 Modern tool usage: Demonstrate requisite hands-on skills to work with a variety of software packages, libraries, programming languages, and software development environment tools useful in engineering large scale systems

PO6 The engineer and society: Make judicious use of resources and understand the impact of technology across the societal, ethical, environmental, and economicaspects.

PO7 Environment and sustainability: Find technological solutions by considering the environmental impact for sustainable development

PO8 Ethics: Practice principles of professional ethics and make informed decisions after a due impact analysis.

PO9Individual and team work:Work efficiently in individual and team-oriented projects of varying size, cultural milieu, professional accomplishments, and technological backgrounds.

PO10Communication: Effectively communicate and exchange ideas and solutions to any individual including peers, end-users, and other stakeholders.

PO11 Project management and Finance: Apply the principles of project management in general and software project management in particular with focus on issues such as the life cycle, scoping, costing, and development.

PO12 Life-long learning: Exhibit the aptitude for independent, continuous, and life-long learning required to meet their professional and career goals.

Program Specific Outcomes (PSOs)

PSO1Exhibit specialized knowledge in some sub-areas of Computer Science and Engineering such as Theoretical Computer Science, Computer Systems, Artificial Intelligence, Cyber-physical Systems, Cybersecurityand use this specialized knowledge base to solve advanced problems

PSO2Perform gap analysis in terms of systems and technologies and prepare roadmaps for incorporating state-of-the-art technology into system analysis, design, implementation, and performance.

PSO3 Demonstrate research and development skills needed to define, scope, develop, and market futuristic software systems and products



PSO4Demonstrate knowledge and skills at the required depth and breadth to excel in post-graduate and research programs

Machine, Data and Learning

Name of the Academic Program	: B.Tech. in Computer Science and Engineering
Course Code	:
Title of the Course	: Machine, Data and Learning
L-T-P	: 3-1-0
Credits	:4
(I – I acture hours T–Tutorial hou	D-Drastical hours)

(L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. Prerequisite Course / Knowledge:

Data Structures, Computer Programming

2. <u>Course Outcomes (COs</u>)

After completion of this course successfully, the students will be able to:

CO-1. Understand basic ML concepts such as Underfitting, Overfitting and Bias-Variance tradeoff CO-2. Gain hands-on experience of applying these concepts to example problems

CO-3. Understand local search techniques with focus on Genetic algorithms CO-4. Understand the basics of Probability and Utility theory

CO-5. Usage of these concepts in the context of formal models such as Decision theoretic models and Bayesian networks CO-6. Understand Decision tree learning and notion of Information Gain

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program</u> Specific Outcomes (PSOs) – Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO_4
CO 1	3	3	3	2	2	3	I	-	3	2	2	2	1	2	2	1
CO 2	2	2	3	1	1	3	I	Ι	3	2	1	2	1	2	2	1
CO 3	2	1	3	1	1	3	-	-	3	2	1	2	1	2	2	1
CO 4	2	2	3	2	2	3	-	-	3	2	2	2	1	2	2	1
CO 5	3	3	3	2	1	3	-	-	3	2	1	2	1	2	2	1
СО	3	2	3	2	1	3	-	-	3	2	1	2	1	2	2	1



6								

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

Unit 1: Overview of AI and ML

Unit 2: Basic ML concepts including Data and generalization, Overfitting, Underfitting, Bias-variance tradeoff

Unit 3: Local Search Techniques, Genetic Algorithms Unit 5: Basics of Probability and Utility Theory Unit 6: Decision Theory, Markov Decision Process, Modeling observation errors

Unit 7: Decision Tree Learning, Construct decision trees from examples, Notion of information gain Unit 8: Bayesian networks

References :

- Python ML by Example by Yuxi (Hayden) Liu, Packt Publishing, 2017
- Stuart Russell and Peter Norvig, Artificial Intelligence A Modern Approach, Pearson Education Inc., 2009

5. Teaching-Learning Strategies in brief:

The course lectures will cover the core concepts while assignments will provide ample scope to implement and understand many of the concepts in more detail. Learning of theoretical concepts and problem solving will be enabled via quizzes, mid and final exams.

6. Assessment methods and weightages in brief :

Assignments	: 35 marks,
Quizzes	: 15 marks,
Mid Exam	: 20 marks,
End Exam	: 30 marks

Science-2

NAME OF FACULTY: Marimuthu Krishnan + Nita Parekh Name of the Academic Program:B. Tech. (CSE) Course Code: SC1.111 Title of the Course: Science II L-T-P: 3-1-0 (L= Lecture hours, T=Tutorial hours, P=Practical hours) Credits: 4

1.Prerequisite Course / Knowledge:NA

<u>2.Course Outcomes (COs)</u> (5 to 8 for a 3 or 4 credit course):

The course is divided into two halves:

First Half: Computing in Sciences

Second Half: Introduction to Biology



Outcomes of the First Half (Computing in Sciences):

After completion of the first half of this course successfully, the students will be able to

CO-1: Outline the uses of Monte Carlo to evaluate multidimensional integrals that appear in theoretical natural sciences

CO-2: Describe numerical algorithms and pseudocodes to solve ordinary and partial differential equations that appear in theoretical natural sciences

CO-3: Apply computational methods to find numerical solutions to scientific problems

Outcomes of the Second Half (Introduction to Biology):

After completion of this course successfully, the students will be able to

CO-1: Familiarize themselves with basic terms and terminology in biology, various biological entities and their function, DNA, RNA, proteins, and enzymes, cell and its functionality,

CO-2: appreciate that biology is very quantitative and how sequence analysis using algorithms can help in understanding the evolution, function of genes and proteins

CO-3: carry out a mini-project to learn how to go from sequence to structure, function and disease association

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes</u> (PSOs) – Course Articulation Matrix

For the First Half (Computing in Sciences):

	РО 1	PO 2	РО 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
CO 1	3	3	1	2	3	2	2	2	2	2	2	2	3	3	3	3
CO 2	3	3	1	2	3	2	2	2	2	2	2	2	3	3	3	3
CO 3	3	3	1	2	3	2	2	2	2	2	2	2	3	3	3	3

<u>.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes</u> (PSOs) – Course Articulation Matrix

Forthe Second Half (Introduction to Biology):

РО	PO	РО	РО	РО	РО	PO	PO	PO	PO	PO	PO	PS	PS	PS	PS
1	2	3	4	5	6	7	8	9	10	11	12	O1	O2	O3	O4



CO 1	1	1										
CO 2	1	1		1						1	1	1
CO 3	1	1	1	1		1	1	1			1	1

4.Detailed Syllabus:

Syllabus of the First Half (Computing in Sciences):

Unit 1: Monte Carlo method: Its application in solving large dimensional integrals seen in statistical mechanics and quantum mechanics

Unit 2: Solving linear systems: Huckel molecular orbital approximation for band structure in metallic bonding

Unit 3: Algebra of matrices: Singular-Value Decomposition (SVD), Hessian matrix in normal mode analysis, and spectral decomposition

Unit 4:Differential equations in sciences: Prey predator model, dynamics from Newton Laws, molecular dynamics simulation

Unit 5:Stochastic differential equations: Diffusion, bistability of cellular processes

Unit 6:Partial Differential equations in sciences: Heat equation and wave equation

Syllabus of the Second Half (Introduction to Biology):

Unit 1: Introduction: Classification of Living Organisms, Origin of Life and Evolution, Biomolecules – Nucleotides, Amino Acids, Proteins, Enzymes

Unit 2: Cell Biology: Structure and Function - Prokaryotic and Eukaryotic Cells, Cell Cycle – Cell division – Mitosis, Meiosis, DNA Replication, Transition, Translation – Central dogma, DNA amplification, sequencing, cloning, restriction enzymes

Unit 3: Genetics: Mendelian Genetics – Genetic Disorders, Mendelian Inheritance Principles, Non-Mendelian Inheritance, Clinical Perspective

Unit 4:Macromolecules: DNA, Proteins – Structure, Function, Analysis, Carbohydrates – Features, Structure, Metabolism, Kreb cycle

Unit 5:Biological data analysis: Biological Data – sequence, structure, expression, etc., Sequence Data Analysis – alignment, database search, phylogeny, Applications

Reference Books:

1. Molecular Biology of the Cell by Alberts, Johnson, Lewis, Morgan, Raff, Roberts, Walter

2.Lehninger Principles of Biochemistry by David L. Nelson and Michael M. Cox



3.Reading the Story in DNA: A Beginners Guide to Molecular Evolution by Lindell Bromham

4. An Introduction to Computational Physics by Tao Pang

5. Molecular Modelling – Principles and Applications by A. R. Leach

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

The objective of the course is to give the CSE students a flavour of biological sciences and scientific computing. To familiarize the students with available web-based resources (databases and tools) for biological sequence analysis and extract meaningful information. Whenever possible, after a theory lecture to follow up with analysis of real sequence data. Give the student small programming tasks in biological data analysis to be able to appreciate the role of computing in biological data analysis. Applications of computational and mathematical models in natural sciences are also discussed.

<u>6.Assessment methods and weightages in brief (4 to 5 sentences):</u>

Assignments – (10%), Class Quizzes + Mid-term evaluation (20%), Final exam (20%)

Intro to Human Sciences

Name of the Academic Programs	: B.Tech. in CSE, B.Tech in ECE
Title of the Course	: Introduction to Human Sciences
Course code	: HS8.102
L-T-P	: 3-1-0
Credits	: 4

1. Prerequisite Course / Knowledge: Nil

2. Course Outcomes (COs)

After completion of this course successfully students will be able to:

CO1: Discuss the origin and development of key disciplines in the human sciences CO2: Identify some of the fundamental questions that shape and drive inquiry in human sciences

CO3: Demonstrate knowledge of concepts related to theorizing about reflection, society, and culture

CO4: Analyze crucial normative elements and descriptive frameworks in human sciences inquiry

CO5: Develop skills to formulate nuances involved in problems concerning humans and societies

CO6: Write clear and well thought out short essays on topics in humanities and social sciences

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix



	PO1	PÓ2	PÔ3	PÔ4	PÔ5	PO6	PÓ7	PO8	PO9	PO10	PO11	PO12	P\$O1	PSO2	PSO3	PSO4
CO1	1	2	3	1	1	3	2	3	1	1	-	3	1	1	2	3
Ċ02	2	2	3	3	2	3	2	3	1	3	1	3	1	1	2	3
ĊO3	2	2	2	3	1	3	2	3	1	2	1	1	1	1	2	3
CO4	1	2	2	3	1	2	2	3	2	2	-	2	1	2	1	3
Ċ05	2	2	3	3	1	2	3	3	1	1	1	3	1	2	2	2
Ċ06	2	2	3	3	1	3	3	3	2	2	1	2	1	1	3	2

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Syllabus:

The course will be divided into four modules, each of which will introduce students to a particular discipline in the human sciences. The various disciplines that constitute human sciences are:

- 1. Philosophy
- 2. Psychology
- 3. Literature
- 4. History
- 5. Sociology
- 6. Anthropology

Each module will offer a systematic worldview, tools of enquiry to study and analytical frameworks to make sense of topics taken up for discussion. Detailed list of topics under a module will be provided by the faculty teaching that module when the lectures begin. The overarching theme for the topics are the fundamentals of human sciences so that students grasp what humans sciences are all about.

Reference books:

Readings for each of the modules will be given with the commencement of the lectures. There is no single textbook as such for all four modules.

5.Teaching-Learning Strategies in brief:

Each module will have one faculty giving six lectures of 90 mins each Through discipline specific modes of understanding and everyday examples, class lectures will enable students to connect and ponder about themselves, the society and cultures that surrounds them. The teaching-learning strategy emphasises the merits of avoiding simplistic solutions to complex problems and instead ask meaningful questions that enrich debates about how we produce, distribute, consume, reflect, represent, and govern ourselves. Lectures impress upon students the need to critically reflect on issues that are impacted by technology, the historical and social context of the world they live in, the literary and philosophical ideas that permeate human thought and psychological principles of human behaviour.

6. Assessment methods and weightages in brief:

This is mainly a writing-driven course, and the evaluation questions are carefully designed to make students think independently. Students are assessed for abilities like critically assessing issues, questioning



assumptions, clarifying distinctions, and bringing out nuances. In assignments and exams, students are expected to demonstrate these abilities by presenting their views clearly and systematically. Students will be evaluated for each of the four modules and the pattern of evaluation will be decided by the respective faculty.

Evaluation pattern can include weekly assignments, quizzes and term papers. Each module will carry 25 % of total marks. The End Semester exam carries 25% of marks.

Value Education-2

Name of the Academic Program	: B. Tech. in ECE, BTech in CSE
Course Code	: OC3.101
Title of the Course	: VALUE EDUCATION - 2
L-T-P	: 12-6-0 (<mark>Total hours</mark>)
Credits	:2
(L= Lecture hours, T=Tutorial hours	s, P=Practical hours)

1.Prerequisite Course / Knowledge: -NIL-

2. Course Outcomes (COs) :

After completion of this course successfully, the students will be able to:

CO-1: Apply the basic framework of universal human values to understand oneself

CO-2: Explain the relation of self with family, society and nature

CO-3: Explain the concept of living in harmony at all the levels

CO-4: Demonstrate the right understanding of relationships and Right utilization of physical facilities

CO-5: Realise the long-term goal of being happy and prosperous

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	-	-	-	-	-	3	2	3	2	-	-	-	-	-	-	-
CO2	-	-	-	-	-	3	3	3	3	-	-	-	-	-	-	-
CO3	-	-	-	-	-	3	3	3	2	-	-	-	-	-	-	-
CO4	-	-	-	-	-	2	3	3	3	-	-	-	-	-	-	-
CO5	-	-	-	-	-	3	3	3	2	-	-	-	-	-	-	-



Note: '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

<u>4.Detailed Syllabus</u>:

Unit 1: Revisiting goal in life - short term and long term goals; Basic aspirations - Happiness and Prosperity; Role of education and human conduct; Self-exploration; Developing a holistic view Unit 2: Self-reflection and reflecting on relationships; understanding value-based life Unit 3: Living in harmony at 4 levels: self-self, self-family, self-society, self-nature Unit 4: Harmony in Society; Broadening one's perceptions;

Unit5: Nature and Sustainability; Our role in protecting Nature;

Reference Books:

1. R.R. Gaur, R. Sangal, G. P. Bagaria. 2009. A Foundation course in Human Values and Professional Ethics. Excel books, New Delhi.

2. Randy Pausch. 2008. The Last Lecture. Hachette Books.

3. E. F. Schumacher. 1973. Small is beautiful: a study of economics as if people mattered. Blond & Briggs, Britain.

4. P. L. Dhar, R. R. Gaur. 1990. Science and Humanism. Commonwealth Publishers.

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

This is a discussed based course. The instructor shares information on a topic and guides the discussion in the class by asking the right questions. By keeping the objectives in mind, the instructor adopts different techniques including smaller group discussions, role-play/skit, use of video clips/films or images to analyse and some activities to keep the students engaged in class throughout. Talks by experts who made a difference are also organised for the batch.

6.Assessment methods and weightages in brief (4 to 5 sentences):

This is a Pass/Fail course. The assessment methods include submissions of assignments and term papers. Critical thinking is expected from watching relevant short films or by reading assigned books. The classroom participation is also taken into consideration for evaluation. There are a few community-based activities and projects also. Participation in them is also important.

Weightage for each kind of assessment may be given

Basics of Ethics

Name of the Academic Programs	: B.Tech. in CSE, B.Tech in ECE
Title of the Course	: Basics of Ethics
Course code	: HS0.203
L-T-P	: 3-1-0
Credits	: 2



1. Prerequisite Course / Knowledge: Nil

2. Course Outcomes (COs)

After completion of this course successfully students will be able to:

CO1: Explain the philosophical nature of the basic concepts and principles of ethics

CO2: Analyze ethical arguments for logical validity, soundness, and informal fallacies

CO3: Demonstrate the knowledge of conceptual challenges involved in normative inquiry in the ethical domain

CO4: Develop skills to formulate fundamental nuances in ethical justification and explanations

CO5. Identify the various kinds of normative elements that constitute ethical frameworks

CO6. Discuss the major tenets of normative ethical theories and their scope of application

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	PÔ1	PO2	PÔ3	PÓ4	PÔ5	PO6	PÓ7	PO8	PO9	PO10	PO11	PO12	P\$O1	PSO2	PSO3	PSO4
Ċ01	1	2	3	1	1	3	2	3	1	1	-	3	1	1	2	3
ĊO2	2	2	3	3	2	3	2	3	1	3	1	3	1	1	2	3
CO3	2	2	2	3	1	3	2	3	1	2	1	1	1	1	2	3
CO4	1	2	2	3	1	2	2	3	2	2	-	2	1	2	1	3
Ċ05	2	2	3	3	1	2	3	3	1	1	1	3	1	2	2	2
Ċ06	2	2	3	3	1	3	3	3	2	2	1	2	1	1	3	2

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

- Unit I Introduction (3 hours): Distinction between conventional and critical ethics, philosophical tools for argument analysis, intuition, evidence, justification, and explanation.
- Unit II Skepticism (4.5 hours): Intrinsic vs Instrumental value, challenge of egoism, problem of cultural relativity and subjectivism, error theory and nihilism, distinction between being ethical and seeming ethical.
- Unit III Goodness (3.5 hours): the problem of defining 'good', naturalistic fallacy and the open question argument, implications of the experience machine thought experiment.
- Unit IV Responsibility (3.5 hours): challenge of attributing moral responsibility to agents, the control, competence and epistemic conditions of responsibility, moral luck.



Unit V – Normative theories (5 hours): Consequentialism, deontology, and virtue ethics

Reference books:

- 1) Shafer-Landau, R. 2019. Living Ethics: An Introduction with Readings. Oxford University Press.
- 2) Shafer-Landau, R. 2013. Ethical Theory: An Anthology 2nd Edition. Wiley-Blackwell.
- 3) Stich, S. and Donaldson. T. 2019. *Philosophy: Asking Questions, Seeking Answers*. Oxford University Press.

5.Teaching-Learning Strategies in brief:

The general teaching strategy employed is the use of moral dilemmas and conceptual puzzles to introduce course topics. Lectures make use of this strategy to impress upon students the need to critically reflect on ethical issues and the relevance of doing a careful, philosophical investigation of those issues. Student interaction at this stage is aimed at bringing out conflicting ethical intuitions. This is followed up by introducing proper vocabulary to map out the problems involved in normative moral assessment. Using case studies and toy examples, ethical principles and methods of inquiry are taught so that students develop effective reasoning skills to engage with any real-world ethical matter. Student interaction and discussion at this stage is aimed to give flesh to the intuitions identified in the previous stage. The teaching-learning strategy emphasises the merits of avoiding simplistic solutions to complex ethical problems and instead ask meaningful questions that enrich moral debates.

6. Assessment methods and weightages in brief:

This is mainly a writing-driven course, and the exercise questions are carefully designed to make students think independently in ethical contexts. Students are assessed for abilities like logically dissecting issues, questioning assumptions, clarifying distinctions, and bringing out nuances. In assignments and exams, students are expected to demonstrate these abilities by presenting their views clearly, assessing competing positions systematically, anticipating possible objections to a reasoned conclusion and composing cogent responses to those objections. The assessment components and their weightages are as follows. Assignments: 60 marks, class participation: 10 marks, Mid semester exam: 10 marks, End semester exam: 20 marks.

Course descriptions of Elective Courses

Title of the Course: Program VerificationCredits: 2,Lectures-Tutorials-Practicals:3-1-0 (hours/week)Faculuty Name:Venkatesh Choppella

<u>1. Course structure</u>

2.Prerequisite courses



- 1. Computer Programming
- 2. Discrete Mathematics

3 Course outcomes

A student graduating from the Program Verification course should be able to perform each of the following sample tasks:

- 1. CO1: Specify simple computational problems Use logic and func- tional notation to precisely specify a problem in terms of an input- output relation. This includes problems related to elementary data structures and sequential algorithms.
- 2. CO2: Model sequential algorithms as iterative systems Model se- quential algorithms for searching and sorting and basic graph algo- rithms as simple iterative systems.
- 3. CO3 Prove sequential algorithms correct Write down a complete proof of total correctness of sequential algorithms for searching and sorting.
- 4. CO4: Use verification tools Develop facility to use software tools (Dafny, or Z3 etc.) for expressing and verifying correctness of algorithms

4 Mapping of Course Outcomes to Programme and Programme Specific Outcomes

Table 1: Mapping of Course Outcomes to programme and programme specific outcomes

Programme Outcome (PO/PSO)	CO1	C02	C03	C04
PO1	3	3	3	3
Engg. Knowledge				
P02	3	3	1	1
Problem Analysis				
PO3	3	3	3	3
Design/Develop				
PO4	2	2	2	2
Complex Problems				
PO5	1	1	1	3
Modern tool usage				
PO6	1	2	3	2
Engr. & Society				
PO7	1	1	1	1



Environment & Sustainability				
PO8	1	2	2	1
Ethics				
PO9	1	1	1	2
Team work				
P10	3	3	3	3
Communication				
P11	1	1	1	1
Project Mgmt & Finance				
P12	3	2	2	2
Life learning				
PS01	3	3	3	3
Specialised knowledge				
PS02	3	3	3	3
Roadmap for technologies				
PS03	3	3	3	3
Research &				

Continued on next page

Continued from previous page				
Programme Outcome (PO/PSO)	CO1	C02	C03	C04 Development Skills
PS04	3	3	3	3

Potential for PG study

5 Syllabus

Propositional Logic: Syntax of propositional logic. Evaluation of propo- sitional logic formulas. Analytical tableaux. Deduction Systems.



First Order Logic: Quantifiers, syntax of first order logic. Evaluation. Deduction Systems. Applications to program specification. Writing proofs.

Sequential Program models Discrete Flows, fixed points, convergence, limit maps, sequential problem solving, bound functions and invariants. Semantics of programming constructs.

Hoare Logic Programs as predicate transformers. Hoare rules for assign- ment, sequential composition, conditional composition and iteration. Proofs in Hoare Logic.

6 Texts and References

Textbooks

MLCS *Mathematical Logic for Computer Science*, 3rd Edition. Murdoch Ben-Ari. Springer, 2013. This is the main text for the first half of the course. Reserve copies in the library.

MICS A Mathematical Introduction to Computer Science. Kasturi Viswanath.

Universities Press, 2008. This is the main text for the second half of the course. Reserve copies in the library.

Other References

LiCS Logic in Computer Science. Huth and Ryan. Cambridge University Press.

7 Teaching and Learning strategies

Lectures will cover the theoretical foundations of program verification: propo- sitional and first order logic and state space models of programs. Lecture material will also include working with modern theorem provers and proof assistants in order for students to give a hands-on feeling of *working* with logic. Question-answer discussion will accompany each class. Assignments will challenge the student to master proof and modeling techniques. Summative assessments will be through a quiz, mid-semester and a final exam. Reading assignments will precede each lecture. Homework (programming) assignments will mostly involve the use of technologies related to verification of programs.

<u>8 Assessment (Tentative)</u>

Item	Weight (%)
Homeworks	35
Quiz	15
Mid-semester exam	20
Final exam	30

Appendix: Programme and Programme Specific Out- comes

Programme Outcomes (POs)



- PO1 :: Engineering knowledge Use concepts from varied disciplines in- cluding Computer Science, Electronics, Mathematics, and the Sciences, to engineer and develop systems of varying scale.
- PO2 Problem analysis Identify, formulate and analyze complex engineer- ing problems reaching substantial conclusions using first principles of Mathematics, Natural Sciences and Engineering Sciences.
- PO3 Design/Development of solutions Identify and bring to fore the necessary concepts from Computer Science and arrive at creative ways to solve problems that take into account the societal, cultural, and ethical considerations.
- PO4 Conduct investigations of complex problems Interpolate and ex- trapolate based on existing knowledge base and self-learning skills to investigate the dynamics of complex problems and find solutions.
- PO5 Modern tool usage Demonstrate requisite hands-on skills to work with a variety of software packages, libraries, programming languages, and software development environment tools useful in engineering large scale systems
- PO6 The engineer and society Make judicious use of resources and un- derstand the impact of technology across the societal, ethical, environ- mental, and economic aspects.
- PO7 Environment and sustainability Find technological solutions by considering the environmental impact for sustainable development
- PO8 Ethics Practice principles of professional ethics and make informed decisions after a due impact analysis.
- PO9 Individual and team work Work efficiently in individual and team- oriented projects of varying size, cultural milieu, professional accom- plishments, and technological backgrounds.
- PO10 Communication Effectively communicate and exchange ideas and solutions to any individual including peers, end-users, and other stake- holders.
- PO11 Project management and Finance Apply the principles of project management in general and software project management in particu- lar with focus on issues such as the life cycle, scoping, costing, and development.
- PO12 Life-long learning Exhibit the aptitude for independent, continu- ous, and life-long learning required to meet their professional and career goals.

Programme Specific Outcomes (PSOs)

- PSO1 Exhibit specialized knowledge in some sub-areas of Computer Science and Engineering such as Theoretical Computer Science, Computer Sys- tems, Artificial Intelligence, Cyber-physical Systems, Cyber-security and use this specialized knowledge base to solve advanced problems
- PSO2 Perform gap analysis in terms of systems and technologies and pre- pare roadmaps for incorporating state-of-the-art technology into sys- tem analysis, design, implementation, and performance.
- PSO3 Demonstrate research and development skills needed to define, scope, develop, and market futuristic software systems and products.
- PSO4 Demonstrate knowledge and skills at the required depth and breadth to excel in post-graduate and research programs.



<u>Title of the Course</u>	: Introduction to Information Security
Faculty Name	: Ashok Kumar Das
Name of the Academic Program	: B.Tech. In CSE / M.Tech. in CSE/CSIS
Course Code	: CS8.401
L-T-P	: 3-1-0
Credits	:2

(Half semester course)

(L=Lecture hours, T=Tutorial hours, P=Practical hours)

<u>1.Prerequisite Course / Knowledge:</u>

Discrete Structures, Programming Languages

<u>2.Course Outcomes (COs)</u>:

After completion of this course successfully, the students will be able to

CO-1: Demonstrate problem solving skills related to security

CO-2: Demonstrate critical thinking skills

CO-3: Demonstrate security protocols practically

CO-4: Demonstrate knowledge of B lock chain technology and its security aspects

CO-5: Demonstrate knowledge of Design and analysis of Internet of Things (IoT)-related security protocols

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> Outcomes (PSOs) – Course Articulation Matrix

	Р 01	P O2	Р О3	P O4	P O5	Р Об	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C 01	1	1	2	2	3	2	2	2	2	2	2	2	2	2	2	3
C O2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3	3
C O3	1	1	2	2	3	2	2	2	1	2	2	2	2	2	3	3
C O4	1	1	2	2	3	2	2	2	1	1	2	2	2	2	3	3
C O5	1	1	2	2	3	2	3	2	1	1	2	2	2	2	2	3



Note '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

- Unit 1: Basics of Cryptography: Cryptographic goals and objectives; Types of attacks, passive and active attacks;Introduction to Number Theory; Complexity Theoretic Connections; Overview of symmetric and public key cryptography
- Unit 2: Basics of System Security: Overview of intrusion detection: Types of intruders, intrusion

detection and prevention mechanisms; Overview of software vulnerabilities: Overview of phishing,

Buffer Overflow (BOF), heap overflow, and SQL injection attacks

- Unit 3: Basics of Network Security: Overview of encrypting communication channels
- Unit 4: Introduction to Internet of Things (IoT) security: IoT architecture; various IoT applications; security requirements, security attacks, threat model for the IoT ecosystem; taxonomy of security protocols
- Unit 5: Introduction to Blockchain technology: Various applications of Blockchain of Things (BCoT); centralized versus decentralized models; types of blockchain; brief overview of various consensus algorithms; block formation and addition in a blockchain

Reference Books:

1. William Stallings, "Cryptography and Network Security: Principles and Practices," Pearson Education, 6th Edition, 2014.

- 2. Bernard Menezes, "Network Security and Cryptography," Cengage Learning, 2010.
- 3. Behrouz A. Forouzan, "Cryptography and Network Security," Special Indian Edition, 2010.

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

* Design of efficient and secure symmetric/public key cryptosystems

- * Design of efficient intrusion detection systems
- * Understanding various system related attacks and their remedies
- * Understanding security aspects of IoT-related applications
- * Understanding Blockchain technology and its usage in various real-life applications

<u>6. Assessment methods and weightages in brief (4 to 5 sentences):</u>

- In-Class Tests: 20%
- Assignments: 20%
- Mid Semester Examination: 20%
- End Semester Examination: 40%



Title of the Course	: Introduction to Quantum Information and Computation
Faculty Name	: Indranil Chakrabarty
Course Code	: CS4.401
LTP	: 2-1-0.
Credits	: 2
(L=Lecture hours, T=Tutor	ial hours, P=Practical hours)

Name of the Academic Program: B.Tech. in Computer Science and Engineering

Prerequisite Course / Knowledge:

Knowledge of Advanced Linear Algebra, Quantum Mechanics, Classical information Theory

Course Outcomes (COs):

After completion of this course successfully, the students will be able to..

CO-1. Explain the basic idea of Qubits (Quantum States), Pure and Mixed States, Quantum Measurements, Entanglement, Quantum Gates and the idea of extension of Entropy from Classical to Quantum. Learning Dirac Algebra to solve problems of Quantum Computing and Information

CO-2. Demonstrate familiarity with process like Quantum Measurement, Information processing tasks like Teleportation, Superdense Coding, Entanglement Swapping, s Quantum Circuits.

CO-3: Synthesize proofs of theorems related to Quantum Entropy using mathematical and logical arguments.

CO-4. Design Quantum Circuits with Universal Gates,

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	Р О 7	P O 8	P O 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C 01	1	1	1	2	1	-	-	-	-	-	-	3	3	3	3	3
C O2	2	2	1	3	1	-	-	-	-	-	-	3	3	2	3	3



C O3	1	2	-	3	-	-	-	-	_	-	-	2	3	2	2	3
C O4	1	2	-	3	2	-	-	-	-	-	2	2	3	3	3	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

Detailed Syllabus:

Unit 1. Introduction and Overview: Transition from Classical to Quantum (2L)

Unit 2. Foundations of Quantum Theory I: States, Ensembles, Qubits, Pure and Mixed

states, Multi-qubit states, Tensor Products, Unitary transformations, Spectral

Decomposition theorem, Singular value Decomposition, Generalized Measurement,

Projective Measurement, POVM (4L)

Unit 3. Quantum Entropy and Entanglement: Quantum Entropy, EPR Paradox, Schmidt

Decomposition. (2L)

Unit 4. Basic Quantum Information Processing Protocols: Teleportation, Super Dense

Coding, Entanglement Swapping. (2L)

Unit 5 Quantum Computation: Introduction to quantum computing, Pauli Gates, Hadamard

Gates, Universal Gates, Quantum algorithms. (2L)

Reference Books:

<u>Preferred Textbooks</u>: 1. Quantum Computation and Quantum Information –M. A. Nielsen, I. L. Chuang. Cambridge University Press.

Other Books: 1. Quantum Computer Science: An Introduction --- N. D. Mermin, Cambridge University Press. 2. Quantum Computing: From Linear Algebra to Physical Realizations---M. Nakahara, T. Ohmi, Taylor and Franchis Group. 3. Lectures on Quantum Information (Physics Textbook)---D. Brub, G. Leuchs, WILEYVCH.

Teaching-Learning Strategies in brief (4 to 5 sentences):



First of all there will be lectures which will introduce the motivations, concepts, definitions along with simpler examples. After that there are going to be assignments and quizzes that will make sure that the students have understood the concepts. These will be followed by deeper lectures and assignments as the area is interdisciplinary and new. These will also be supplemented with innovative problems so that they can apply the concepts learned by them.

Assessment methods and weightages in brief (4 to 5 sentences):

Mid semester exam	- 20%	
End SemExam	- 30%	
Assignment	- 15%	
Quizz		- 15%
Project	-20%	

Name of the Program	: Introduction to Algorithm Engineering
Course Code	:
Credits	: 2

L - T - P

(L - Lecture hours, T-Tutorial hours,

:

P - Practical hours)

Semester, Year : Spring 2022

(Ex: Spring, 2022)

Pre-Requisites : first course on algorithms, programming, computer architecture/ organization

Course Outcomes :

(list about 5 to 6 outcomes for a full 4 credit course)

The action verbs to be used for writing the course outcomes can be found on slide 22 in the following presentation. You may remove this line and the following link after the course outcomes are formulated.

https://iiitaphyd-my.sharepoint.com/:b:/r/personal/dyacad_iiit_ac_in/Documents/NBA-2020-21/Reference%20Documents/Curriculum%20Design%20in%20NBA%20Framework%20and%20 Course%20design%20for%20all%20faculty%20IIIT%20Hyderabad%207th%20july%202021.pdf? csf=1&web=1&e=387W1k

At the end of the course, a student will be able to:

CO – 1: Demonstrate familiarity and scope of algorithm engineering

CO - 2: Explain the significance of algorithm engineering and analyze the practical performance of algorithms in connection to the nature of input



CO - 3: Apply algorithm engineering principles to implement a variety of graph and semi-numerical algorithms

Course Topics

(Please list the order in which they will be covered, and preferably arrange these as five to six modules.)

- 1. Introduction to algorithm engineering, its scope, and its importance -1
- 2. Cache-Aware Design: Algorithms and Techniques 1
- 3. Cache-Oblivious Design: Algorithms and Techniques -1
- 4. A Primer on Parallel Algorithms 3

:

- 5. Graph connectivity -2
- 6. Eccentricity and Diameter -2
- 7. Centrality Measures on Graphs -2

Preferred Text Books:

Reference Books : Reference papers that are used for some of the course topics will be posted as they are discussed in class.

E-book Links : Book being developed by the instructor available at http://cstar.iiit.ac.in/~kkishore/pgae.pdf

Grading Plan : Since the course is a half-course, we will have one quiz evaluation and one final evaluation.

Type of Evaluation	Weightage (in %)
Quiz-1	20%
Mid Sem Exam	
Quiz-2	
End Sem Exam	30%
Assignments	25%
Project	25%
Term Paper	
Other Evaluation	

For Office Use Only (starts on a new page)



Mapping of Course Outcomes to Program Objectives: (1 – Lowest, 2—Medium, 3 – Highest, or a '-' dash mark if not at all relevant). Program outcomes are posted at

https://iiitaphyd-my.sharepoint.com/:w:/r/personal/dyacad_iiit_ac_in/Documents/NBA-2020-21/Course%20Content/IIIT-CSE-ECE.docx?d=w111f0effcaea41b3a4d1e8a3fbc6332d&csf=1&web=1&e=z1Khby

	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	PO 9	P O 10	PO 11	PO 12	PS O 1	PS O 2	PS O 3	PS O 4
C O 1	1	2	2	2	1	1	1	1	1	2	1	1	2	1	1	2
C O 2	1	2	2	2	1	1	1	1	2	2	1	1	2	1	1	2
C O 3	1	2	3	3	1	1	1	1	2	2	1	1	2	1	1	2

Teaching-Learning Strategies in brief (4-5 sentences) :

The course will have hands-on exercises that help students understand the mechanisms available for algorithm engineering. The course project also equips them to explore an existing algorithm and a problem in depth and gain useful practical knowledge. The material used in the course is not part of standard textbook as yet, so lecture slides and reference papers will be made available for reading.

Note: This course description format comes into effect from Spring 2022.

Title of the Course: Introduction to Coding TheoryName of the Academic Program: B. Tech in ECE, B. Tech in CSECourse Code: EC5.205L-T-P: 1.5-0.5-0Credits:2(L= Lecture hours, T=Tutorial hours, P=Practical hours)

1. Prerequisite Course / Knowledge:

Linear Algebra



2. <u>Course Outcomes (COs)</u> :

After completion of this course successfully, the students will be able to:

CO-1: Explain the importance of redundancy and block codes as well as their parameters

CO-2: Discuss the characteristics of linear codes including generator matrix, parity-check matrix and dual code

CO-3: Apply encoding and decoding algorithms to linear codes

CO-4: Analyze the dependence between various parameters of the codes

CO-5: Deduce the additive, multiplicative and vector space structure of finite fields

CO-6: Construct BCH and Reed Solomon codes, given the specifications of the problem.

3.<u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	Р О1	P O2	Р О3	P O4	P O 5	P O 6	P O 7	P O 8	Р О9	PO 10	PO 1 1	PO 12	PS O1	PS O 2	PS O 3	PS O 4
C 01	3	3	2	-	-	-	-	-	-	-	-	2	1	3	1	-
C 02	3	3	2	-	-	-	-	-	-	-	-	-	1	3	1	-
C 03	3	3	2	-	-	-	-	-	-	-	-	-	1	3	1	-
C 04	3	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-
C 05	3	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-
C 06	3	3	2	-	-	-	-	-	-	-	-	-	-	3	-	-

4. <u>Detailed Syllabus</u>:

Unit 1: Noisy channels, block codes, encoding and decoding, maximum-likelihood decoding, minimum- distance decoding, error detection and correction.


Unit 2: Minimum distance, generator and parity-check matrices, dual codes, standard array decoding,

syndrome decoding. Repetition codes, Hamming codes.

Unit 3: Hamming bound, Singleton bound, Gilbert-Varshamov bound, Plotkin bound.

Unit 4: Definitions, prime fields, construction of prime power fields via irreducible polynomials, existence of primitive elements, minimal polynomials.

Unit 5: Bose-Choudhury-Hocquenghem (BCH) codes, Reed-Solomon codes. Applications of Reed-Solomon codes in digital communications and storage.

Reference Books:

- 1. R. Roth, Introduction to Coding Theory, Cambridge University Press, 2007.
- 2. W.C. Huffman and V. Pless, Fundamentals of Error Correcting Codes, Cambridge University Press, 2003.
- 3. S. Lin and D.J. Costello, Error Control Coding, Pearson, 2011
- 4. R.E. Blahut, Algebraic Codes for Data Transmission, Cambridge University Press, 2003

5. <u>Teaching-Learning Strategies in brief (4 to 5 sentences):</u>

The course has lectures supported by tutorials. In tutorials, problems related to the concepts presented in the class are solved by teaching assistants. Exams are conducted periodically so that students can actively engage with the course material. Viva is conducted at the end of the course to assess how students are able to apply concepts learnt in the class to new problems. A project is given towards the end of the course, which requires the students to present a research paper in the area of coding theory in detail.

Type of Evaluation	Weightage (in %)
2 Mid Semester Exams	2 x 10 = 20%
Assignments	30%
Viva	10%
Project	20%
End Semester Exam	20%

6. Assessment methods and weightages in brief (4 to 5 sentences):



Title of the Course	: Software Programming for Performance
Faculty Name	: Deepak Gangadharan
Course Code	: CS3.302
L-T-P	: 3-1-0
Credits	: 2(Half semester course)

(L=Lecture hours, T=Tutorial hours,

P=Practical hours)

Name of the Academic Program: B-Tech in Computer Science and Engineering

<u>1. Prerequisite Course/Knowledge</u>

Basics of Algorithm Analysis, Computer Architecture

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO-1. Explain the algorithmic optimizations necessary to improve the performance of a software on a uniprocessor.

CO-2. Analyze cache dependent performance of algorithms

CO-3. Employ cache-aware (such as tiling)/cache oblivious (such as recursive multiplication) optimizations to improve program performance

CO-4. Analyze the software performance improvement using SIMD Array Processing and Vector Processing Architectures

CO-5. Explain different concurrency platforms such as Pthreads, Threading Building Blocks. CO-6. Develop multicore programs using OpenMP pragmas

CO-7. Explain the basics of GPU architecture

<u>3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	РО 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
CO 1	1	1	-	1	-	-	-	-	-	2	-	3	1	1	1	1
CO 2	3	3	-	3	1	-	-	-	-	1	-	3	3	3	1	3
CO 3	3	1	-	2	-	-	-	-	3	1	-	3	3	1	1	1



CO 4	3	3	-	3	1	-	-	-	-	1	-	3	3	3	1	3
CO 5	1	1	-	1	-	-	-	-	-	2	-	3	1	1	1	1
CO 6	3	2	-	2	3	-	-	1	3	1	-	3	3	2	2	3
CO 7	1	1	-	1	-	-	-	-	-	2	-	3	1	1	1	1

Note 3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus

Unit 1:Algorithmic optimizations – Introduction to optimization of matrix multiplication: Language dependent performance, Loop ordering, compiler optimization, loop parallelization, tiling, vectorization

Unit 2:Memory Hierarchy aware Optimizations – Review on Caches, Conflict misses, Ideal Cache Model and cache misses, Cache analysis of matrix multiplication, Tiling, Recursive Matrix Multiplication

Unit 3:Using SIMD units – Flynn's Taxonomy, Data Parallelism, SIMD Array Processing, Vector Processing – Vector Registers, Vector Functional Units, Memory Banking, Basic Vector Code Performance, Vector Chaining, Multiple Memory Ports, Masked Vector Instructions

Unit 4:Programming Multi-cores – Shared Memory Hardware, Concurrency Platforms – Pthreads, Threading Building Blocks, OpenMP – Creating Threads, Synchronization: critical, barrier, Parallel loops, Data Sharing, Memory model

Unit 5: Acceleration using Hardware Accelerators (GPU)

Reference Books:

No specific text book, but the material would be taken from different books such as:

- 1. Cormen, Thomas H., et al. Introduction to algorithms.
- 2. Hennessy, John L., and David A. Patterson. Computer architecture: a quantitative approach.

5.Teaching-LearningStrategiesinbrief

Weekly lectures cover the topics in the syllabus. Tutorials cover how to use some tools for measuring performance of software implementations. There are couple of assignments that will provide the students experience in programming some functions and improve the performance employing the techniques learned in theory. Firstly they would learn how to improve cache performance and then exploit parallelism in code by employing multicore programming using OpenMP.

Type of Evaluation	Weightage (in %)
Quizzes	40
Assignments	30
Project (End semester)	30

6. Assessment methods and weightages in brief



Comment: Please revisit the Assessment and provide weightage for end semester exam for at least 30% marks

Title of the Course : <u>Digital Signal Analysis</u>

Course Code:

L-T—P: 2-1-0

Credits: 2

Name of the Academic Program B. Tech. in CSE

Prerequisite Course / Knowledge:

No prerequisite as it is a core course for CLD program.

Course Outcomes (COs):

After completion of this course successfully, the students will be able to..

CO-1 : Introduce the fundamentals of digital signal representation and processing to undergraduate students of CLD/CS/CSD.

CO-2: Introduce the advantage of a transformed domain representation.

CO-3: Application of basic signal processing to speech signals.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	P O 1	P O 2	P O 3	P O 4	РО 5	P O 6	РО 7	P O 8	РО 9	PO 10	PO 11	PO 12	PS 01	PS O2	PS O3	PS O4
C O 1	3	2	1	1	1	1	1	1	2	1	1	2	-	3	-	-
C O 2	3	2	1	1	1	1	1	1	2	1	1	2	-	3	-	-
C O 3	3	2	2	1	1	1	1	1	2	1	1	2	-	3	-	-

Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs). Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

Mapping with PSOs, where applicable.

Detailed Syllabus:

Unit 1: Basics of Fourier series and transform, sampling and quantisation, different types of signals and systems.

Unit 2: Z-transform, FIR and IIR systems. Introduction to digital filter design.

Unit 3: Application of concepts using speech signals.



Reference Books:

- 1. Digital signal processing by John G. Proakis and Dimitris K Manolakis.
- 2. Digital signal processing by Alan V. Oppenheim and Ronald W. Schafer.
- 3. Introduction to Digital Speech Processing by Lawrence R. Rabiner and Ronald W. Schafer, now Publishers Inc. Hanover, USA, 2007

Teaching-Learning Strategies in brief (4 to 5 sentences):

It is a mathematical oriented signal processing course, so regular problem solving assignments are given to understand the concepts. Surprise class tests are conducted based on assignments to test the seriousness in assignment solving. As a part of teaching practical examples like speech signal is used for demonstration of mathematical concepts learned.

Assessment methods and weightages in brief (4 to 5 sentences):

Assignments	- 20%
Quiz	- 30%
End exam	- 50%

Data Visualization

Title of the Course Faculty Name Name of the Prog Course Code Credits L - T - P	: INTRODUCTION TO BRAIN AND COGNITION : KAVITA VEMURI& VISHNU SREEKUMAR ram : BTech CSE : CS9.301 : 2 :2-0-0	
(L - Lecture nours, 1-1	utoriai nours,	
P - Practical hours)		
Semester, Year	: Spring 2022 (H2)	
(Ex: Spring, 2022)		
Pre-Requisites	: NONE	
Course Outcomes	:	

(list about 5 to 6 outcomes for a full 4 credit course)

After completion of this course successfully, the students will beable to:

CO-1:develop understanding and familiarity with seminal research findings in brain and cognition.

CO-2:read, interpret, critique, and evaluate research explaining brain/mind/behavior.

CO-3:critically think about the relationship between diverse fields such neuroscience, cognitive psychology, and cognitive science



CO-4:critical understanding and evaluation of the experiments, methods and practices for empirical and computational investigation of cognition utilizing various instruments by different teams in Cognitive Science Lab in order to make informed decision about the Lab to work for further research in the Dual Degree Program

Course Topics

(Please list the order in which they will be covered, and preferably arrange these as five to six modules.)

Module 1: Introduction

Brain Anatomy basics; Spatial and temporal aspects of the Brain and Cognition; Methods of Investigation of the Brain and Cognition

Module 2: Vision

Visual Perception; Recognizing Objects; Attention

:

Module 3: Memory

Acquisition; Relation between Acquisition and Retrieval; Memory of Complex Events

Module 4: Knowledge

Concepts; Language

:

:

:

:

Module 5: Thinking

Problem Solving and Intelligence; Conscious and Unconscious Thought

Preferred Textbooks

- Daniel Reisberg (2019). Cognition: exploring the science of the mind. 7th Edition. W. W. Norton & Company, NY, USA
- V. Srinivasa Chakravarthy (2019). Demystifying the Brain: A Computational Approach. Springer, Singapore (1st Edition).

Reference Books

- Eric Kandel, James H. Schwartz, and Thomas Jessell (2012). Principles of Neural Science. McGraw Hill Education (5th Edition).
- John R. Anderson (2009). Cognitive Psychology and its Implications. Worth Publishers (7th Edition).

E-book Links

Grading Plan

(The table is only indicative)

Type of Evaluation	Weightage (in %)
Quizzes (3 out of 4: each 20%)	
	60%
End Som Exom	30%
Term Paper	10%



Mapping of Course Outcomes to Program Objectives: (1 – Lowest, 2—Medium, 3 – Highest, or a '-' dash mark if not at all relevant). Program outcomes are posted at

	РО 1	PO 2	PO 3	РО 4	РО 5	РО 6	РО 7	РО 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
CO1	1	1	2	1	1	2	1	1	1	1	1	2	1	1	1	1
CO2	3	1	2	3	1	3	3	1	1	1	1	2	2	2	3	2
CO3	2	1	2	2	1	3	2	2	1	1	1	2	1	2	3	1
CO4	1	1	1	1	1	1	1	1	2	3	1	2	1	1	1	1

Teaching-Learning Strategies in brief (4-5 sentences) :

The IBC course is primarily lecture and discussion-based learning course. Students will be introduced to undergraduate-level introductory topics and issues in brain and cognition. Reading material will be assigned. Students will be required to engage in discussions, and to write a term paper on related topics. Students will be encouraged to relate the theory topicsto everyday experiences and will be asked to evaluate the event/phenomenon/ processes critically and scientifically. They will be encouraged to interact with various research teams in Cognitive Science Lab to familiarize themselves with the research projects so that they can start thinking about a future lab to conduct their research work.

Title of the Course	: Computer Graphics					
Course Code	: CS7.302					
L-T-P	:3-1-0					
Credits	:2					
(L= Lecture hours, T=Tutorial hours,						

P=Practical hours)

Name of the Academic Program: BTech in Computer Science & Engineering

<u>1.Prerequisite Course / Knowledge</u>:.....

<u>2.Course Outcomes (COs)</u> (5 to 8 for a 3 or 4 credit course):

After completion of this course successfully, the students will be able to.

CO-1 Introduce the 3D shape representation and modelling for Computer Graphics applications. **CO-2** Introduce Graphics libraries for development of graphics applications.

CO-3 Introduce Graphics Pipeline for rendering of 3D objects.

CO-4 Explain Graphics concepts/algorithms for fast and realistic rendering of 3D objects including lighting, texture, shadow as well as using the GPU based acceleration data structures like k-d trees.



1. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	Р О1	P O2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	PO 10	PO 11	PO 12	PS 01	PS O2	PS 03	PS O4
C 0 1	3	3	1	2	2	1	1	1	1	1	1	2	3	2	1	2
C O 2	2	1	2	1	3	1	1	1	3	1	1	2	3	2	2	3
C O 3	3	3	3	3	3	2	1	2	3	2	1	3	3	3	2	3
C O 4	3	3	3	3	3	2	1	2	3	2	1	3	3	3	2	3

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

2. Detailed Syllabus:

Unit 1: Basic Concepts in Computer Graphics : Revision to basics of Computer Graphics : Primitives, Geometric Transformations, Hierarchical Modeling and Viewing Transforms, Clipping Points and Lines, Polygon Filling, Visibility, intro to WebGL.

Unit 2: Advance Concepts in Computer Graphics: View Frustum Culling, Depth Buffering, Lighting Shading & Shadows Mapping, Texture Mapping, BRDF, raytracing.

Reference Books:

1. "Computer Graphics with OpenGL: 3rd Edition", by Donald Hearn and Pauline Baker. Published in India by Pearson Education (Prentice Hall).

2 Reference: "Computer Graphics: Principles and Practice: 3rd Edition", by James D. Foley, Andries van Dam, Steven Feiner and John Hughes. Published by Addison-Wesley

1. Teaching-Learning Strategies in brief (4 to 5 sentences):

The course lectures will include interactive graphics content for effectively conveying the basic concepts as

well as small activities to promote the understanding of the lecture content. Significant focus will be on problem solving aspect and concepts will be introduced in the context of relevant practical problems



related to 3D modelling and rendering of virtual objects/world. Tutorials will further try to bridge the gap between theoretical understanding and practical aspects of problem solving. Assignments are designed to to solve problems that are based on simple extensions of concepts described in the lectures.

2. Assessment methods and weightages in brief (4 to 5 sentences): Homeworks/Assignments: 55%

Quiz1: 7%

Quiz 2: 8%

End Exam: 30%

Title of the Course Faculty Name Course Code	: Performance modeling of computer systems : Tejas Bodas :
Credits	:2
L - T - P :	
(L - Lecture hours, T-Tutoria	l hours,
P - Practical hours)	:2-0-0
Semester, Year	: Spring 2022
(Ex: Spring, 2022)	
Name of the Program	: CSE and or ECE

Pre-Requisites : MA6.101 Probability and Statistics Course Outcomes :

Course outcomes (CO's): After completion of the course, the students will able to

- 1. Explain and identify the role of performance modeling in different computer systems such as data networks, server farms and cloud computing platforms.
- 2. Apply Markov chains to model and a variety of computer systems and analyze their performance metrics like response time, waiting time or job loss probability.
- 3. Derive expressions for the average delay or average number of jobs waiting for service in a variety of queueing systems.
- 4. Design and analyze the performance of multi-server queueing systems that have applications to cloud computing
- 5. Analyze and understand the impact of scheduling policies like FIFO, LIFO, processor sharing and random routing on the performance of queues.
- 6. Identify causes for performance degradation (large latency problem) in queueing systems and offer easy scalable solutions



Course Topics : Following is the tentative list of topics to be covered in this course in about 12 lectures. (Each lecture is of 90 mins.)

Module 1: (2 lectures)

- 1. Motivation to Performance modeling (Modeling = Design + analysis)
- 2. Probability refresher
- 3. Basics of Stochastic processes

Module 2: (2 lectures)

- 4. Discrete time Markov chains
- 5. Continuous time Markov chains

Module 3: Elementary Queues (2 lectures)

- 6. M/M/1 queue
- 7. Loss queues
- 8. Little's law and PASTA property

Module 4; Server-farms and networks (3 lectures)

- 9. Multi-server queues
- 10. Network of queues
- 11. load balancing systems
- 12. Applications to data centers, cloud computing and distributed systems

Module 5: Scheduling and resource allocation in computer systems (3 lectures)

- 13. M/G/1 queues
- 14. Performance analysis of FIFO, round-robin, processor sharing, LCFS
- 15. SMART scheduling policies

Preferred Text Books : Performance modeling and design of computer systems (Cambridge press) by Mor Harchol-Balter (Professor, CMU)

Reference Books : 1) **Probabilistic modeling by Isi Mitrani**

2) Queueing Systems (vol 1 and 2) by Klienrock

E-book Links : NA

Grading Plan :

(The table is only indicative)



Type of Evaluation	Weightage (in %)
Quiz-1	15
Mid SemExam	30
Quiz-2	15
End Sem Exam	40
Assignments	0
Project	0
Term Paper	0
Other Evaluation	0

Mapping of Course Outcomes to Program Objectives: (1 – Lowest, 2—Medium, 3 – Highest, or a '-' dash mark if not at all relevant). Program outcomes are posted at

https://iiitaphyd-my.sharepoint.com/:w:/r/personal/dyacad_iiit_ac_in/Documents/NBA-2020-21/Course%20Content/IIIT-CSE-ECE.docx?d=w111f0effcaea41b3a4d1e8a3fbc6332d&csf=1&web=1&e=z1Khby

	Р 01	Р О2	Р О3	Р О4	Р 05	Р Об	Р 07	Р 08	Р 09	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS 04
C 01	3	3	3	2	1	1	2	1	1	1	1	2	3	2	3	3
C O2	3	3	3	2	2	1	2	1	1	1	1	2	3	2	3	3
C O3	3	3	3	2	2	1	2	1	1	1	1	2	3	2	3	3
C 04	3	3	3	2	1	1	2	1	1	1	1	2	3	2	3	3
C 05	3	3	3	2	1	1	2	1	1	1	1	2	3	2	3	3
C0 6	3	3	3	2	2	1	2	1	1	1	1	2	3	2	3	3
•																

Teaching-Learning Strategies in brief (4-5 sentences) :

16. The course is planned to be a fine balance between theory and practice.



- 17. Traditionally, this course has been a theory intensive course with little emphasis on practical applications. We will however flip this around.
- 18. We will introduce theoretical mathematical concepts on a need to know basis or as and when required.
- 19. The emphasis will be to look at plenty of practical examples of queueing systems that we encounter not just in our daily lives but also see in advanced computing systems.
- 20. The goal is not only to design queueing systems that offer better performance guarantees but also to be able to analyze such systems so as to fine tune or control them.
- 21. The 12 lectures are meant to be very interactive, there would be lot of discussion and exchange of ideas on the design aspect of queueing systems.
- 22. As for the analysis, ample practice problems and practice assignments would be provided to gain analytical expertise.

Note: This course description format comes into effect from Spring 2022.

Title of the course	: Modern Complexity Theory
Faculty Name	: Girish Varma
Course Code	:
L-T-P	: 3-1-0
Credits	:4

Prerequisite Course / Knowledge:

Should have taken Introduction to Algorithms, and Formal Languages, or equivalent courses

Course Outcomes (COs):

After completion of this course successfully, the students will be able to..

CO-1: Understand different models of computation including Turing Machines, Boolean Circuits and complexity measures of time, space, depth.

CO-2: Demonstrate familiarity with various complexity classes including P, NP, PSPACE, NC and problems like Halting Problem, 3SAT.

CO-3: Design reductions between problems to show hardness of solving a problem in a complexity class.

CO-4: Synthesize proofs of upper and lower bounds of resources required for solving a computational problem using clear mathematical and logical arguments.

CO-5: Apply principles of NP-Completeness and NP-Hardness to avoid intractability in design of computational problems.

CO-6: Create mathematical models and complexity measures for novel computational models.



	PO	РО	PS	PS	PS	PS										
	1	2	3	4	5	6	7	8	9	10	11	12	0	0	0	0
													1	2	3	4
CO1	2	2	2	3	1	1	1	1	2	2	1	1	3	1	1	1
CO2	1	2	2	2	1	1	1	1	2	2	1	1	3	1	1	2
CO3	1	3	2	3	1	1	1	1	2	2	1	1	2	1	1	2
CO4	1	3	2	2	1	1	1	1	2	2	1	1	3	1	2	2
CO5	1	3	2	3	1	1	1	1	2	2	1	1	3	1	1	3
CO6	2	3	2	3	1	1	2	1	2	3	1	1	2	2	3	3

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

'3' for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping.

Detailed Syllabus:

Unit 1: Models of Computation and Impossibility Results: Turing Machines, Circuits, Encoding of Problems, Halting Problem, Shannon's Counting Lower bound.

Unit 2: Complexity Measures and Classes: Time, Space, Depth measures of complexity, Time, Space hierarchy theorems, Nondeterminism, Savitch's theorem, P, NP, P/poly, PSPACE, EXP, L, NL. **Unit 3: Completeness and** Hardness Reductions: 3SAT, Cook-Levin Theorem, NP-Complete, NL- Complete, Hardness reductions for common problems like VertexCover, Independent Set, Knapsack etc.

Unit 4: Advanced Topics: Definitions and relationships between PH, RP, BPP, NC including theorems like Karp-Lipton, Adleman's theorem, Derandomization Techniques.

Reference Books:

- S. Arora and B. Barak (2000), Computational Complexity: A Modern Approach, Cambridge University Press.
- C. Moore & S. Mertens (2011), The Nature of Computation, Oxford University Press.
- M. Sipser (2014), Introduction to Theory of Computation, Cengange Learning.
- C. Papadimitriou (1994), Computational Complexity, Addison Wesley Longman.

Teaching-Learning Strategies in brief (4 to 5 sentences):

Lectures will initially introduce the motivations, concepts, definitions along with simpler examples.

This will be followed by assignments and quizzes that will make sure that the students have understood the concepts. These will be followed by deeper lectures and assignments which lead



the students to the bigger questions in the area. The students will be given an advanced topic and will be required to summarize it in a presentation or a term paper. This will encorage self-

exploration and lead the student to do research on fundamental questions.

Assessment methods and weightages in brief (4 to 5 sentences):

- Light In-class Quizes: 20%
- Assignments: 20%
- Deep Quiz 1: 10%
- Deep Quiz 2: 10%
- Mid and End Exam: 30%
- Student Presentation and Scribe notes: 10%

Title of the Course: Principles of Programming Languages (PoPL)

Faculty Name: Venkatesh Chopella

1 Course structure

Name CSC 415 Principles of Programming Languages

Credits 4, Lectures-Tutorials-Practicals=3-1-0 (hours/week)

2 **Prerequisite courses**

- 1. Computer Programming
- 2. Discrete Mathematics (with some exposure to writing proofs)
- 3. Automata Theory

3 Course outcomes

A student graduating from a PoPL course should be able to perform each of the following sample tasks:

- 1. CO1: Document Abstract Syntax Document and critique the ab- stract syntax of industrial scale programming language like C or Java.
- 2. CO2: Design domain specific languages Design a small, domain pur- pose languages like a language for propositional logic and implement them.
- 3. CO3 Design object small, oriented language Design a small object-oriented language implement it either using an interpreter or by em- bedding it into a base language.
- 4. CO4: Compare languages Compare and analyse the semantic express- ibility (in terms of first class values) between imperative languages like C and functional languages Racket, and object oriented languages like Java and Python.



- 5. CO5: Specify application interfaces Specify the structure of a soft- ware application like a spreadsheet or a word processor in terms of its interface as a language of user operations and its internal structure as an abstract machine.
- 4 Mapping to Programme and Programme Specific Outcomes

Table 1: Mapping of Course Outcomes to programme and programme specific outcomes

Programme Outcome (PO/PSO)	CO1	C02	C03	C04	C05
PO1	3	3	3	3	3
Engg. Knowledge					
P02	3	3	3	3	3
Problem Analysis					
PO3	1	3	3	2	2
Design/Develop					
PO4	2	3	3	3	3
Complex Problems					
PO5	3	3	3	2	2
Modern tool usage					
PO6	1	2	2	2	2
Engr. & Society					
PO7	1	1	1	1	1
Environment & Sustainability					
PO8	2	1	1	1	1
Ethics					
PO9	2	3	3	2	2
Team work					
P10	3	3	3	3	3
Communication					

Programme Outcome (PO/PSO)

CO1 C02 C03 C04 C05

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY H Y D E R A B A D

P11	1	1	1	1	1
Project Mgmt & Finance					
P12	3	2	2	2	3
Life learning					
PS01	3	3	3	3	2
Specialised knowledge					
PS02	3	3	3	3	3
Roadmap for technologies					
PS03	2	3	3	3	3
Research & Development Skills					
PS04	3	3	3	3	2

Potential for PG study

5 Syllabus

Functional Programming: Abstract vs. Concrete Syntax. Racket syn- tax, Functions, recursion, syntactic extensions, higher-order functions, map, reduce and other combinators.

Operational Semantics: Abstract Reduction Systems, , Transition Sys- tems, Reduction, Simplification and Evaluation relations. Judgements and Rules. Rule Induction.

- Scope: Identifiers, Scope and extent, Lexical scope, Environments, 'Dy- namic scope' and parameters. Closures
- State: Stores and imperative constructs, explicit and implicit store refer- ences, objects, invariants and safety, interfaces and constructors, in- heritance, Parameter passing. Call-by-value, call-by-name and lazy evaluation.

Control: Tail calls, Contexts, continuations, continuation passing style, ex- ceptions, threads.

Types: Types syntax, type safety theorems. Type inference

Special Topics (if time permits): Monads, Concurrency.

6 Texts and References

Textbook

EOPL *Essentials of Programming Languages* 3rd Edition. Friedman and Wand. This is the main text for the course. Available on Amazon.in.

References

HtDP How to Design Programs. Felleisen et al. Available online.



SICP Structure and Interpretation of Programs. Abelson and Sussman. Available <u>online</u>. Accompanying video lectures also available <u>online</u>.

TRaAT Term Rewriting and All That. Baader and Nipkow. Chapters 1 and 2.

PLAI Programming Languages: Application and Interpretation 2nd Edi- tion. Available online.

SSICS Simply Scheme: Introducing Computer Science. Brian Harvey and Matthew Wright. Available online.

RG Racket Guide. Available as part of Racket language documentation.

7 Teaching and Learning strategies

Lectures will cover the theoretical aspects of operational semantics but will have plenty of examples explaining interpreters of programming langugaes visually and interactively. Question-answer discussion will accompany each class. Quizzes each week will test student's attention diligence, and con- cept recall, understanding and application. Summative assessments will be through a mid-semester and a final exam or project. Reading assignments will precede each lecture. Homework (programming) assignments will mostly involve implementation of interpreters discussed in the class and the text book. Tutorials will walk-through abstract syntax tree annotation, compo- nents of the interpreter implementation, and inductive proofs of properties in operational semantics.

8 Assessment (Tentative)

Item	Weight (%)
Quizzes (1 per week)	25
HW	25
Mid-semester exam	20
Final exam/Project	30

Appendix: Programme and Programme Specific Out- comes

Programme Outcomes

- PO1 :: Engineering knowledge Use concepts from varied disciplines in- cluding Computer Science, Electronics, Mathematics, and the Sciences, to engineer and develop systems of varying scale.
- PO2 Problem analysis Identify, formulate and analyze complex engineer- ing problems reaching substantial conclusions using first principles of Mathematics, Natural Sciences and Engineering Sciences.
- PO3 Design/Development of solutions Identify and bring to fore the necessary concepts from Computer Science and arrive at creative ways to solve problems that take into account the societal, cultural, and ethical considerations.
- PO4 Conduct investigations of complex problems Interpolate and ex- trapolate based on existing knowledge base and self-learning skills to investigate the dynamics of complex problems and find solutions.



- PO5 Modern tool usage Demonstrate requisite hands-on skills to work with a variety of software packages, libraries, programming languages, and software development environment tools useful in engineering large scale systems
- PO6 The engineer and society Make judicious use of resources and un- derstand the impact of technology across the societal, ethical, environ- mental, and economic aspects.
- PO7 Environment and sustainability Find technological solutions by considering the environmental impact for sustainable development
- PO8 Ethics Practice principles of professional ethics and make informed decisions after a due impact analysis.
- PO9 Individual and team work Work efficiently in individual and team- oriented projects of varying size, cultural milieu, professional accom- plishments, and technological backgrounds.
- PO10 Communication Effectively communicate and exchange ideas and solutions to any individual including peers, end-users, and other stake- holders.
- PO11 Project management and Finance Apply the principles of project management in general and software project management in particu- lar with focus on issues such as the life cycle, scoping, costing, and development.
- PO12 Life-long learning Exhibit the aptitude for independent, continu- ous, and life-long learning required to meet their professional and career goals.

Programme Specific Outcomes (PSOs)

- PSO1 Exhibit specialized knowledge in some sub-areas of Computer Science and Engineering such as Theoretical Computer Science, Computer Sys- tems, Artificial Intelligence, Cyber-physical Systems, Cyber-security and use this specialized knowledge base to solve advanced problems
- PSO2 Perform gap analysis in terms of systems and technologies and pre- pare roadmaps for incorporating state-of-the-art technology into sys- tem analysis, design, implementation, and performance.
- PSO3 Demonstrate research and development skills needed to define, scope, develop, and market futuristic software systems and products.
- PSO4 Demonstrate knowledge and skills at the required depth and breadth to excel in post-graduate and research programs.

Title of the Course	: Principles of Information Security										
Faculty Name	: Ankit Gangwal										
Course Code:	: CSE418										
Credits:	: 4										
L-T-P:	: 3-1-0										
(L= Lecture hours, T=Tutorial hours,											

P=Practical hours)



Name of the Academic Program: B.Tech. in Computer Science and Engineering

<u>1. Prerequisite Course / Knowledge</u>:

Basic principles of algorithms.

2. Course Outcomes (COs) :

After completion of this course successfully, the students will be able to..

CO-1 Discuss mathematical concepts of cryptographic primitives

CO-2 Describe fundamental concepts and algorithms of cryptography, including encryption/decryption and hash functions

CO-3 Summarize different authentication techniques and describe programs like PGP & S/MIME

- CO-4 Discuss network security principles, applications, and practices
- CO-5 Analyse protocols for various system security objectives using cryptographic tools
- CO-6 Evaluate the role of different security mechanisms like passwords, access control mechanisms, firewalls, etc.

<u>3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	Р О1	Р О2	Р О3	Р О4	Р О5	Р Об	Р О7	Р 08	Р О9	PO 10	PO 11	PO 12	PS 01	PS O2	PS 03	PS 04
C 01	3	2	1	1									2	1		2
C O2	3	2	2	2									3	2	2	3
C O3	1	1	2	1									2	1	1	2
C 04	2	2	2	2									2	1	2	2
C 05	2	3	2	3									3	2	2	2
C 06	1	1	2	1									1	1	2	1

Note: '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:



- Unit 1: Introduction: Security Trends, Security attacks, Security services, Security Mechanisms, A Model for Network Security Model, Classical Encryption Techniques, Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography.
- Unit 2: Block Ciphers and Data Encryption Standard: Block Cipher Principles, Data Encryption Standard, Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles, Advanced Encryption Standard, Evaluation Criteria of AES, AES Cipher, Multiple encryption and Triple DES, Block Cipher Modes of Operation, RC4.
- Unit 3: Public-key Encryption and Hash Functions: Principles of Public Key Cryptosystems, RSA Algorithm, Key Management, Message Authentication and Hash Functions, Authentication Requirements, Authentication Functions, Message Authentication, Hash Functions, Security of Hash Functions and MACs, Digital Signatures, Authentication Protocols, Digital Signature Standard.
- Unit 4: Network Security Applications: Kerberos, X.509Authentication Service, Public Key Infrastructure, Pretty Good Privacy, S/MIME, IP Security Overview, IP Security architecture, Authentication Header, Encapsulating Security Payload, Combining Security associations, Key Management.
- Unit 5: System Security: Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction, Intruders, Intrusion Detection, Password Management, Malicious Software, Firewalls, Trusted Systems

Reference Books:

- 1. W. Stallings, Cryptography and Network Security Principles & Practices, 4th edition, Prentice Hall, 2005
- 2. J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC Press, 2007
- 3. B. Schneier, Applied Cryptography, 2nd edition, John Wiley & Sons, Inc, 2001
- 4. Research papers

<u>5. Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

Lectures by integrating ICT into classroom teaching; tutorials involving problem solving; being a fundamental course, it requires critical thinking and active learning by the students to solve problems.

<u>6. Assessment methods and weightages in brief (4 to 5 sentences):</u>

Assignments	30 marks
Mid Semester Examination	30 marks
End Semester Examination	40 marks

TITLE OF THE COURS	SE:: Optimization Methods
FACULTY NAME	: Dr. Naresh Manwani
Course Code	: CS1.404
CREDITS	: 4 Credits
L-T-P	: 3-1-0



TYPE-WHEN: Spring 2022**PRE-REQUISITE**: Strict Prerequisites: NIL

EXPECTED BACKGROUND:

To follow this course, some level of familiarity with linear algebra (specially, vectors and matrices) is expected. In addition, student is expected to know the fundamentals of algorithms and some of the popular problems (eg. shortest path.)

OBJECTIVE:

- 1. To enable students to formulate and solve problems in an optimization framework.
- 2. To expose a set of powerful tools and techniques to the students. To demonstrate how these tools (i.e. optimization methods) can be used in practice.
- 3. To visualize the optimization algorithms and know the numerical and practical issues in their implementation.
- 4. To relate the optimization methods to applications in diverse areas.

COURSE TOPICS :

- 1. CO-1: Linear Programming, Geometric Interpretation, SimplexMethod, Duality, primal dual method, Interior point methods, Ellipsoidal methods, Computational Issues.
- 5. CO-2: Integer programming, LP relaxation, Examples from combinatorial optimization. Shortest paths, network flows and matchings.
- 6. CO-3: Convex sets and functions. Need for constrained methods in solving constrained problems.
- 7. CO-4: Unconstrained optimization, Optimality conditions, Gradient Descent, Newton Method, Quasi- Newton Methods, Trust Region Methods. Conjugate Gradient Methods. Least Squares Problems.
- 8. CO-5: Constrained Optimization, Optimality Conditions and Duality. Convex Programming Problem. Quadratic Programming. Dual Methods, Penalty and Barrier Methods, Interior Point Methods.
- 9. CO-6: Linear Equations, Solutions based Matrix Factorization, Singular Value Decomposition,
- 10. CO-7: Additional topics (if time permits) related to
 - 1. Specific Algorithms (eg. Cutting plane algorithms, Stochastic gradients)
 - 2. Applications in Approximate Algorithms
 - 3. Computational issues in large scale optimization
 - 4. Heuristic methods for optimization

PREFERRED TEXT BOOKS:

- 1. S. Boyd and L Vandenberghe, ``Convex Optimization", Cambridge University Press (Online Copy available at: <u>http://www.stanford.edu/~boyd/cvxbook/</u>).
- 11. L Vandenberghe, Lecture Notes for Applied Numerical Computing, (Online available at: <u>http://www.ee.ucla.edu/~vandenbe/103/reader.pdf</u>).
- 12. Edwin K. P. Chong, Stanislaw H. Żak, Introduction to Optimization, Fourth Edition, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons.



REFERENCE BOOKS:

- 1. M T Heath, ``Scientific Computing", TMH (Most of First six chapters)
- 13. C H Papadimitriou and K Steiglitz, Combinatorial Optimization: Algorithms and Complexity" (Most of First seven chapters), Dover.
- 14. D Bertisimas and J N Tsitsiklis, ``Introduction to Linear Optimization", Athena Scientific.
- 15. J Matousek and B. Gartner, "Understanding and Using Linear Programming", Springer, 2007.

OUTCOME:

This course will help in sharpen the problem solving skills of students. Students will have experience informally stating problems with the associated constraints, and solving them with computer friendly algorithms.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	Р О 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PS O1	PS O2	PS O3	PS O4
CO1	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO6	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3
CO7	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3	3

GRADING PLAN:

Type of Evaluation	Weightage (in %)
Small Quizzes (10 quizzes)	10%
Mid-Sem Exams (2)	30%
End Sem Exam	20%
Assignments	25%
Term Paper/Project	10%
Scribe	5%



Title of the Course: Advanced AlgorithmsName of the Faculty: Kishore Kothapalli,Course Code:L-T-P:3-1-0Credits:4Name of the Academic Program: B. Tech. in CSE

Prerequisite Course / Knowledge:

Should have taken Introduction to Algorithms, and Formal Languages, orequivalent courses

Course Outcomes (COs):

After completion of this course successfully, the students will be able to..

CO-1 :Demonstrate familiarity with using randomness in computing

CO-2: Apply principles of randomized algorithm design and analyze them for correctness and efficiency

CO-3: Synthesize randomized algorithms with either zero-error or one sided error for a variety of problems

CO-4: Explain the significance of parallelism to modern day computing and problem-solving needs

CO-5: Apply principles and paradigms of parallel algorithm design and analyze parallel algorithms for correctness and efficiency

CO-6: Create efficient parallel algorithms for a variety of semi-numerical problems and problems on graphs

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	P 0 1	P 0 2	P 0 3	P 0 4	P 0 5	P 0 6	P 0 7	P 0 8	P 0 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C 01	1	2	2	2	1	1	1	2	1	1	1	2	3	1	1	3
C 02	2	3	2	2	1	1	1	2	2	2	1	3	3	1	1	3
C 03	2	2	2	2	1	1	1	2	1	1	1	2	3	1	1	3
C 04	1	3	2	2	1	1	1	2	2	2	1	3	3	1	1	3



C 05	2	2	2	2	1	1	1	2	2	2	1	3	3	1	1	3
C 06	2	2	2	2	1	2	1	2	1	1	1	2	3	1	1	3

Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs). Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

Mapping with PSOs, where applicable.

Detailed Syllabus:

Unit 1: Randomness in computing: Tail inequalities and applications, fingerprinting, proofs of

existence, expander graphs

Unit 2: randomized rounding, approximate counting

Unit 3: Parallelism in computing: Models of PRAM, Basic algorithms for prefix, search, sort, merge,

Unit 4: Parallel algorithms for lists, graphs, and symmetry breaking

Reference Books:

R. Motwani and P. Raghavan (1995), Randomized Algorithms, Cambridge University Press.USA.

- 1. J. JaJa (1992), Introduction to Parallel Algorithms, Addison-Wesley, USA.
- 2. G. Tel (2000).Distributed Algorithms,2nd Edition, Cambridge University Press. USA.

Teaching-Learning Strategies in brief (4 to 5 sentences):

The course lectures will include activities that promote the understanding of the lecture content by using small examples that students work out during the class itself and promote active and participatory learning. A good part of the lecture will involve problem solving and finding solutions to problems rather than expositing known material. In class tests that are held periodically are useful as summative assessments. Homework assignments are designed to reiterate the material covered in class lectures and also solve problems that are based on simple extensions of concepts described in the lectures.

Assessment methods and weightages in brief (4 to 5 sentences):

- Homeworks: 20%
- In-class Objective Tests: 20%
- Quiz1: 15%
- Quiz 2: 15%
- End Exam: 30%



Title of the Course	: Distributed Systems
Name of the Faculty	: Kishore Kothapalli,
Course Code:	
L-T-P	:3-1-0
Credits	:4
(L= Lecture hours, T=7	Futorial hours ,

P=Practical hours)

Name of the Academic Program B. Tech. in Computer Science and Engineering

<u>1.Prerequisite Course / Knowledge:</u>

An understanding of operating systems, networks, and algorithms

2.Course Outcomes (COs):

After completion of this course successfully, the students will be able to..

CO-1 : Explain the challenges faced by distributed systems in terms of lack of global time, synchrony, faults, programming support, etc.

CO-2 :Employ standard distributed programming frameworks to write distributed programs for problem solving

CO-3 : Explain the properties and design principles of various real-world and practical distributed systems

CO-4 :Interpret the impact of faults in distributed systems in the context of important problems such as distributed agreement, distributed consensus, and distributed transaction processing

CO-5 :Analyze distributed algorithms for graphs with respect to correctness, round complexity, and message complexity.

CO-6 : Analyze the limitations of distributed systems and assess the operational scope of large scale distributed systems

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	P O 12	PS O1	PS O2	PS O3	PS O4
C 01	1	1	1	1	1	1	1	1	1	2	1	2	3	3	3	3
C O2	1	2	2	2	3	2	1	1	2	1	2	2	3	3	3	3



C O ²	1		1	2	2	1	2	1	1	1	2	1	2	3	3	3	3
C	1		2	2	2	1	2	1	1	2	2	1	2	3	3	3	3
O4																	
C O5	2		2	2	2	1	1	1	1	2	1	2	2	3	3	3	3
C O 6	2	2	2	2	2	1	2	1	1	2	2	2	2	3	3	3	3

Note: '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

- Unit 1
 - Introduction
 - Communication models
 - Time and Synchronization
 - Practice: MPI/Map-Reduce
- Unit 2
 - Distributed file systems
 - Consensus, Agreement, Locking
 - Practice: GFS, Chubby
- Unit 3
 - Distributed Database systems
 - Practice: NoSQL, MongoDB
- Unit 4
 - Limitations of distributed computing
 - Self-Stabilization
 - CAP Theorem
- Unit 5
 - Distributed algorithms for graphs
 - Advanced Topics such as Blockchain, Distributed Storage, and Distributed Program Verification

Reference Books:

1. A.D. Kshemkalyani, M. Singhal, (2011) <u>Distributed Computing: Principles, Algorithms, and Systems,</u> ISBN: 9780521189842, paperback edition, Cambridge University Press, USA.

2. N. Lynch, 1996. Distributed Algorithms, Morgan Kauffman, USA, Chapter 5.



3. Other significant papers from conferences such as OSDI, USENIX, NSDI, for material that is not part of textbooks

5.Teaching-Learning Strategies in brief:

Lectures of the class use the active learning methodology and allow students to learn concepts thoroughly in class along with practising small examples. Homeworks assigned as part of the course are useful to impart knowledge of using practical distributed programming tools and libraries. To promote team work, some of the homeworks are done in a team of two students. The overall learning from the course is enhanced by doing a substantial practice-based project – usually in a team of two students. The course will also have a summative assessment in the form of a final/end-semester exam.

6. Assessment methods and weightages in brief :

- i. In-class Quiz Exams (Cumulative over several): 15%
- ii. Homeworks: 20%
- iii. Project: 25%
- iv. End Semester Examination: 40%

Title of the Course	: Data Systems
Name of the Faculty	: Krishna Reddy Polepalli
Course Code	: CS4.401
L-T-P	: 3-1-1.
Credits	: 4
(L=Lecture hours, T	=Tutorial hours,

P=Practical hours)

Name of the Academic Program: B.Tech. in Computer Science and Engineering

1. Prerequisite Course / Knowledge:

Basic principles of Operating systems, Structured Query Language, Relational Data Model, Data structures, Programming language, Algorithms,

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to..

CO-1. Develop the tree-based and hash-based indexing algorithms to improve efficiency of the retrieval

CO-2. Tune the optimizer module of DBMS to meet the performance demands of diverse

applications, including distributed applications.



CO-3: Design the recovery sub-system of any given information system

CO-4. Design archival strategy for any given information system

CO-5. Develop a concurrency control algorithm for any given database system

CO-6. Develop a framework for building a large scale big data system.

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C O1	2	2	2	2	3	-	1	-	3	1	3	2	3	3	2	3
C O2	3	3	3	1	3	-	1	-	2	2	2	3	3	2	1	1
C O3	3	2	2	1	2	2	1	-	2	2	2	1	3	3	3	2
C O4	3	2	2	1	2	2	1	-	2	2	2	1	3	3	3	2
C O5	3	2	2	1	2	2	1	-	2	2	2	1	3	3	3	2
C 06	3	3	3	3	2	2	1	-	2	2	3	2	3	3	3	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

- Unit 1: Introduction, Data storage, Representing data elements (9 hours)
- Unit 2: Index structures, Multidimensional indexes (7.5 hours);
- Unit 3: Query execution, The query compiler (9 hours)
- Unit 4: Coping with system failures, Concurrency control (7.5 hours);

Unit 5: Transaction management, NoSQL and big data systems (9 hours)

• Five mini projects related to the above syllabus will be done by students in the laboratory



References:

- Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, Database System Implementation, Pearson Education, 2003
- Elmasri&Navathe, Fundamentals of Database Systems, 6th Edition, Pearson Education, 2013
- Raghu Ramakrishnan and Johannes Gehrke ,Database Management Systems, Third edition, Mc Graw Hill, 2017
- Abraham Silberschatz, Henry F.Korth, S.Sudarshan, Database system concepts, fifth edition, Mc Graw Hill, 2006
- Research papers

5.Teaching-Learning Strategies in brief:

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing 5 mini-projects in laboratory by the students

6. Assessment methods and weightages in brief :

Assignments in theory: 10 marks, Quizzes in theory: 10 marks, Mid Semester Examination in theory: 20 marks , End Semester Examination in Theory: 30 marks, Assessment of 5 mini projects in Laboratory: 30 marks

Title of the Course	: Compilers
Faculty Name	: Suresh Purini
Course Code	: CS1.403
L-T-P	: 3-1-0.
Credits	: 4
(L = Lecture Hours, T)	$\Gamma = Tutorial Hours,$

P = Practical Hours)

Name of the Academic Program: B.Tech in Computer Science and Engineering

1. Prerequisite Course / Knowledge:

Computer Programming. Data structures and algorithms. Computer Systems Organization. Operating Systems. Automata Theory.



2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Explain the principles and practices underlying production quality compilers such as GCC and LLVM (Cognitive Level: Understand)

CO-2: Modify open source compilers such as GCC and LLVM to support new languages and processor architectures; and write custom analysis and transformation passes. (Cognitive Leve: Apply)

CO-3: Identify problems or sub-problems in real world projects which can be solved by building custom compilers and interpreters of varying scale and complexity. (Cognitive Levels: Analyze, Evaluate and Create)

CO-4: Employ software engineering principles and practices to design, develop and manage complex software engineering tasks. Examples include object oriented design and programming, choosing appropriate design patterns, good support for debugging the system with ease and, develop comprehensive test suite with good coverage. (Cognitive Levels: Analyze, Evaluate and Create)

CO-5: Use software management tools such as GIT, build systems such as Make/Ant etc. Write proper software design documents and end-user manuals (Cognitive Levels: Apply)

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	РО 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C O1	3	3	3	3	3	-	2	-	-	-	-	2	3	3	3	3
C O2	3	3	3	3	3	-	-	-	-	-	-	2	3	3	3	3
C O3	3	3	3	3	3	-	-	-	-	-	-	2	3	3	3	3
C O4	2	3	3	3	3	-	-	-	3	3	-	2	3	3	3	3
C O5	2	2	3	3	3	-	-	3	3	3	3	2	3	3	3	3

4. Detailed Syllabus



• Unit 1: Syntax Analysis

- o Micro and macro syntax specification using regular expressions and context free grammars
- o Lexical Analysis
- o Top-down (LL(1)) and bottom-up (LR(1), LALR(1)) parsing
- Unit 2: Semantic Analysis and IR Generation
 - o Abstract Syntax Tree (AST) construction
 - o Static and Dynamically typed language
 - o Type Checking
- Unit 3: Intermediate Representations and their Generation
 - o Intermediate representations such as three address tuples, stack code
 - o AST to linear intermediate representation generation
 - o Basic blocks and control flow graphs
 - o Static Single Assignment Form (SSA)
 - o LLVM IR case study
- Unit 4: Machine Independent Optimizations
 - o Local and regional optimizations using value numbering optimization as a case study
 - o Global optimizations like constant propagation and dead code elimination
 - o Data flow analysis theory and practice. Examples include Available expressions analysis and live variable analysis.
 - o Compiler phase sequencing problem

• Unit 5: Code Generation and Register Allocation

- o Runtime environment for C-like programming languages
- o Scope and lifetime of variables. Parameter passing mechanisms.
- Generating machine code with virtual registers from machine independent linear intermediate representation.
- o Local and global register allocation. Mapping virtual registers to physical registers.
- o Basics of instruction scheduling

Reference Books:

1. Keith Cooper and Linda Torczon. 2011. Engineering a Compiler, Second Edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. 2006. Compilers: Principles, Techniques, and Tools (2nd Edition). Pearson.

5. Teaching-Learning Strategies in brief

The most important component of this course is the project in which students design a C like imperative programming language. Write a manual for their programming language specifying syntactic and semantic rules along with example programs written in their own language. Over the course, as students are introduced to principles and practices involved in designing various compiler modules, they build the corresponding modules for their programming language. At the end of the course, students will be able to



run the example programs they have written by compiling them with the compiler built by them. The target language for the compiler is usually LLVM IR.

Through the mini homeworks, theoretical ideas introduced in the class are reinforced. Students get continuous support through tutorial sessions, office hours conducted by teaching assistants and the concerned faculty.

6. Assessment methods and weightages in brief

- 1. Mini Homeworks (7 to 8) : 15 percent
- 2. Course Project
 - a. Syntax Analysis: 10 percent
 - b. AST Construction: 10 percent
 - c. Semantic Analysis: 10 percent
 - d. LLVM IR Generation: 10 percent
- 3. Mid Term Quiz: 15 percent
- 4. Final Theory Exam: 30 percent

Title of the Course	: Advanced Computer Networks
Faculty Name	: Ankit Gangwal
Name of the Academic Program	: B.Tech. in Computer Science and Engineering
Course Code	: CSE435
Credits	: 4
L-T-P	: 3-1-0

(L= Lecture hours, T=Tutorial hours, P=Practical hours)

<u>1.Prerequisite Course / Knowledge:</u>

Basic principles of computer networks and algorithms.

<u>2.Course Outcomes (COs</u>) (5 to 8 for a 3 or 4 credit course):

After completion of this course successfully, the students will be able to..

- CO-1 Demonstrate a familiarity with concepts of network management, standards, and protocols
- CO-2 Discuss various privacy-enhancing techniques used in modern computer networks
- CO-3 Apply the knowledge of distance-vector (RIPand IGRP) and link-state (OSPF and IS-IS) routing protocols to find routing paths for a variety of networks
- CO-4 Analyse wireless LAN technologies including IEEE 802.11
- CO-5 Design efficient routing protocols for advanced computer networks (e.g., SDN and ICN)
- CO-6 Develop a framework for building a large-scale enterprise network



<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	Р О1	P O2	Р О3	P O4	Р О5	Р Об	P O7	P 08	Р О9	P O1 0	P 01 1	P O1 2	PS O1	PS O2	PS O3	PS O4
C 01	3	1	2	3	2	1	1	1	1	2	3	3	2	3	1	2
C O2	3	2	3	3	3	2	1	2	1	3	2	3	2	2	2	1
C O3	2	2	3	3	1	1	2	1	1	1	2	2	1	2	1	1
C 04	2	2	3	3	2	2	3	1	2	1	2	2	2	2	3	2
C 05	2	3	2	2	2	1	2	2	1	3	3	2	3	2	2	1
C 06	3	3	3	3	2	2	2	3	2	2	2	2	2	2	2	1

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

Unit 1: Modeling and measurement: Network traffic modeling, network measurement, simulation issues, network coding techniques

- Unit 2: Flow and congestion control, TCP variants, TCP modeling, active queue management
- Unit 3: Routing: Router design, scheduling, QoS, integrated and differentiated services
- Unit 4: Wireless networks: Mobility supports, MAC, multicast
- Unit 5: Overlay networks and Emerging applications: SDN, ICN, P2P, CDN, Web caching, cross-layer optimizations, VoIP, SIP, video over P2P

Reference Books:

- 1. Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach, 5th edition, Elsevier, 2012
- 2. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 6th edition, Pearson Education, 2013
- 3. Jean Walrand and Pravin Varaiya, High-Performance Communication Networks, 2nd edition, Elsevier, 2000



4. Research papers

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

Lectures by integrating ICT into classroom teaching; tutorials involving problem solving; being a systems course, it requires hands-on working as well as critical thinking and active learning by the students to solve practical problems; and finally, project-based learning by implementing semester-long project(s) to solve real-world issues.

<u>6.Assessment methods and weightages in brief</u> (4 to 5 sentences):

Assignments	20 marks
One at home project	30 marks
Mid Semester Examination	20 marks
End Semester Examination	30 marks

Title of the Course	: Software Engineering					
Course Code	: CSE461					
Faculty Name	: Raghu Reddy					
L-T-P	: 3-0-1					
Credits	:4					
(L= Lecture hours, T=Tutorial hours, P=Practical hours)						

1. <u>Prerequisite Course / Knowledge:</u>

Students must have taken Intro to Software Systems, Design and Analysis of Software Systems or Equivalent courses

2. Course Outcomes (COs) (5 to 8 for a 3 or 4 credit course):

After completion of this course successfully, the students will be able to...

CO-1: Demonstrate familiarity with various process models, design patterns, architecture patterns and the characteristics of good software architectures

CO-2 Apply principles of user interface design, sub-system design and analyze the designs for good Software Engineering principles

CO-3: Demonstrate the use of tools to quantitatively measure and refactor existing software systems

CO-4: Compare design trade-offs between different patterns and/or different implementations of the same pattern



CO-5: Design the major components and user interface for a small-scale software system using modeling approaches such as UML class diagrams, and sequence diagrams

CO-6: Critique the quality of a software design and use product quality metrics to assess the quality of delivered software

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	PO 1	PO 2	Р О3	Р О4	Р О5	Р Об	Р О7	Р 08	Р О9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C 01	1	2	1	1	1	1	1	1	2	2	1	1	1	1	1	1
C 02	1	2	2	2	2	2	2	1	3	3	2	2	1	1	2	1
C 03	2	1	2	3	3	1	1	1	2	2	2	2	2	2	1	2
C 04	2	1	2	2	2	1	1	1	3	3	2	2	1	1	2	2
C 05	1	1	2	2	2	1	1	1	3	3	1	1	2	1	2	2
C 06	1	2	3	3	3	1	1	2	3	3	1	2	2	2	2	2

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

2. <u>Detailed Syllabus</u>:

Unit 1: Software Development Lifecyle and importance of architecture and design in the lifecycle, Process models; Modeling using UML.

Unit 2: Anti-patterns; Metrics and Measurement; Reverse Engineering and Refactoring.

Unit 3: Design Principles and Classification of Patterns

- o Structural patterns: Adapter, Composite, Façade, Proxy, Decorator
- o Behavioral patterns: Iterator, Observer, Mediator, Command, Memento, State, Strategy,

Chain of Responsibility

o Creational patterns: Abstract Factory, Builder, Singleton, Factory Method

Unit 4: Software architecture and Architectural business cycle; Quality attributes and Tactics for achieving attributes; Architectural styles and Techniques; Designing Architectures, Case studies.



Reference Books:

- 1. Design Patterns: Elements of Reusable Object- Oriented Software. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Pearson, 2015, ISBN-13 : 978-9332555402
- 2. Refactoring: Improving the Design of Existing Code. Martin Fowler. Addison-Wesley, 2018. ISBN-13:978-0134757599
- 3. Software Architecture in Practice, 3rd edition by Len Bass, Paul Clements and Rick Kazman, Addison- Wesley, 2012. ISBN-13: 978-9332502307

5. Teaching-Learning Strategies in brief (4 to 5 sentences):

The course is delivered using project based learning methodology. Topics like software subsystems modeling, design analysis, design trade-offs, language agnostic designs and component-based software development are taught and reinforced via unit level projects. The lectures emphasize the study and development of software subsystems, comprehension and analysis of design quality attributes. The focus is on application of these concepts to concrete design problems through in-class design exercises and analysis of existing designs of currently implemented software systems. Entire class is run in a studio mode to facilitate discussion between student teams and discuss design trade-offs among students within student teams. Students present their designs and implementations to other students who are expected critique the designs.

6. Assessment methods and weightages in brief (4 to 5 sentences):

Final Exam	22 %
Mid-term Quiz	12 %
Unit Questions	12 %
3 Unit Projects (2 * 17) + (1 * 10)	44 %
Other In-class Activities	10 %

Title of the Course

: Statistical Methods in Artificial Intelligence : Vineet Gandhi

:4

Faculty Name Name of the Academic Program : Btech in CSE and Btech in ECE **Course Code** :CSE471 L-T-P : 3:1:0 Credits (L=Lecture hours, T=Tutorial hours,

P=Practical hours)

1.Prerequisite Course / Knowledge:

Basic probability theory


Basic Linear Algebra

Good programming skills in Python

<u>2.Course Outcomes (COs)</u> (5 to 8 for a 3 or 4 credit course):

After completion of this course successfully, the students will be able to..

CO-1: Data processing: process raw data and convert it into machine exploitable format

CO-2: Problem formulation: formulate a practical problem as a machine learning problem (classification, clustering etc.)

CO-3: Classical algorithms: In depth investigation of theory and practice of classical algorithms in supervised and unsupervised learning (e.g. SVM, Kmeans, decision trees).

CO-4 Deep Learning: Introduction to theory and practice of deep learning and recent advances

CO-5 System building: design practical systems incorporating basic machine learning

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	РО 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C O1	3	2	2	1	3	2	2	3	2	2	2	2	2	2	1	2
C O2	3	3	3	3	1	3	2	3	1	2	2	3	1	3	2	2
C O3	1	2	2	2	2	1	2	1	1	1	2	3	3	2	2	3
C O4	2	2	2	3	3	1	1	1	1	1	2	3	3	2	2	3
C O5	3	1	1	2	3	3	3	2	3	2	2	2	2	3	3	3

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs. Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4.Detailed Syllabus:

Unit 1: Review of basic statistics, linear algebra, probability



Unit 2: Problem formulation in ML, Decision Trees, Nearest Neighbours

Unit 3: Supervised Machine Learning (SVM, Random Forest, Boosting etc.)

Unit 4: Unsupervised Machine Learning (kmeans, recommendation, anomaly detection, PCA, LDF etc.)

Unit 5: Deep Learning

Reference Books:

1. Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, 2nd Edition, John Wiley and Sons, October 2000

2. Christopher M. Bishop, Pattern Recognition and Machine Learning, 2nd Edition, Springer, 2011

3. Ian Goodfellow and Yoshua Bengio and Aaron Courville, *Deep Learning*, 1st Edition, MIT Press, 2016

<u>5.Teaching-Learning Strategies in brief</u> (4 to 5 sentences):

The course involves heavy theory and programming components. The strategy is to first discuss a problem statement, introduce an algorithms and work out the details of the algorithm, and then use the algorithm to solve the problem. A lot of teaching on black board to discuss theory, large assignments are given for covering practical aspects and a large project is given mid-way of the course to cover the system building aspect.

<u>6.Assessment methods and weightages in brief</u> (4 to 5 sentences):

Programming Assignments: 25%

Quiz1: 10%

Quiz2:15%

Final exam: 25%

Course Project: 25%

Title of the Course	: Informatio	on Retrieval and Extraction
Name of the Faculty	: Vasudeva	Varma
Course Code	: CSE.474	
L-T-P	: 3-1-1.	
Credits	: 4	
(L= Lecture hours, T=T	utorial hours, P=	Practical hours)
Name of the Academic P	rogram:	B.Tech. in Computer Science and Engineering

1. Prerequisite Course / Knowledge:



Basic principles of Computer programming, Statistical Methods in Artificial Intelligence, Programming languages, and Algorithms.

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to..

- CO-1. Develop algorithms to retrieve information from unstructured data
- CO-2. Design and architect information retrieval systems for world wide web
- CO-3: Design Web crawling systems
- CO-4. Design algorithms to process noisy data in document repositories

CO-5. Develop information extraction systems

<u>3.Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C O1	3	3	3	3	3	1	1	-	2	2	2	2	3	3	2	2
C O2	3	3	3	3	3	1	1	1	2	2	2	2	3	3	2	2
C O3	3	3	3	3	2	1	1	1	2	2	2	2	3	3	2	2
C O4	3	3	3	3	2	1	1	-	3	2	2	2	3	3	2	2
C O5	3	3	3	3	2	1	1	-	3	2	2	2	3	3	2	2

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus:

Unit 1: Introduction to Information retrieval, Information Extraction and Information Access systems. (6 hours)

Unit 2: Information Retrieval Models and Evaluation of IR systems (7.5 hours);

Unit 3: Web Information Retrieval (4.5 hours)

Unit 4 Natural Language Processing in IR (7.5 hours)

Unit 5: Machine Learning in Information Retrieval Systems (12 hours)

Unit 6: Information Extraction (4.5 Hours)



Unit 7: IR Applications (12 Hours)

References:

- Introduction Information Retrieval Chris Manning et al (the Stanford IR Book) (ISBN-13: 978-0521865715)
- Search Engines: IR in Practice Bruce Craft et al (ISBN-13: 978-0136072249)
- Research papers

5.Teaching-Learning Strategies in brief:

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing one mini-project and a major project by the students

<u>6. Assessment methods and weightages in brief</u>:

Assignments in theory: 10 marks

Quizzes in theory: 10 marks

Mid Semester Examination: 20 marks

End Semester Examination: 60 marks

Title of the Course	: Advanced Natural Language Processing
Faculty Name	: Manish Shrivastava
Course Code	: CS7.501
Credits	: 4
L - T - P	: 3-1-0
(L - Lecture hours, T-Tut	orial hours,
P - Practical hours)	
Semester, Year :	Monsoon, 2022
(Ex: Spring, 2022)	
Name of the Program	: BTech III year, Computational Linguistics Dual Degree III year
Pre-Requisites	: None
Course Outcomes	:

After completion of this course successfully, the students will be able to -

- I. Demonstrate the knowledge of Advanced building blocks of NLP
- II. Apply NLP machine learning algorithms for Machine Translation, Summarization



- III. Demonstrate the knowledge of Dense and contextual representation for NLP
- IV. Explain the concepts behind Deep Learning models
- V. Discuss the approaches to global and contextual semantic representation
- VI. Apply the above concepts for fundamental NLP tasks.

Course Topics

- A. Distributed Semantics
 - o Contextual Distributed Semantics
- B. Models such as ELMO, BERT, ERNIE and their derivatives
- C. Statistical Machine Translation methods
 - o Early Neural Machine Translation models
- D. Extractive and Abstractive Summarization o Neural Summarization Methods
- E. Reinforcement learning for NLP

:

:

Preferred Textbooks :

None. Mostly research papers.

Reference Books

Statistical Machine Translation by Philip Koehn

Deep Learning by Ian Goodfellow

E-book Links

- 1. <u>https://www.deeplearningbook.org/</u>
- 2. http://www.cs.cmu.edu/~tom/mlbook.html

:

:

Grading Plan

(The table is only indicative)

Type of Evaluation	Weightage (in %)
Quiz-1	2.5
Mid SemExam	10
Quiz-2	2.5
End Sem Exam	20
Assignments	15
Project	40
Term Paper	10



Other Evaluation	

Mapping of Course Outcomes to Program Objectives:

	PO 1	PO 2	PO 3	PO 4	РО 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
C 01	1	3	3	2	2	2	1	-	1	1	1	3	3	3	3	2
C O2	3	3	3	3	3	1	1	1	3	3	3	3	3	3	2	3
C 03	3	3	3	3	3	-	-	-	2	2	2	3	3	3	2	3
C 04	3	3	3	3	3	-	-	-	2	2	2	3	3	3	2	3
C 05	3	3	3	3	3	-	-	-	2	2	2	3	3	3	2	3
C 06	3	3	3	3	3	1	1	1	3	3	3	3	3	3	3	3

Teaching-Learning Strategies in brief (4-5 sentences) :

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing fourassignments and a project. Evaluation based on personal viva to judge deeper understanding.

===============

Note: This course description format comes into effect from Spring 2022.

Title of the course	:	Data Analytics-I
Course Code	:	CS4.405

L-T-P: **3-1-1**. Credits: **4** (L= Lecture hours, T=Tutorial hours, P=Practical hours) TYPE-WHEN: Fifth semester and onwards FACULTY NAME: P. Krishna Reddy

1. Prerequisite Course / Knowledge:

- 1. Data and Applications, or equivalent courses that cover Data modelling, normalization, SQL
- 2. First courses on programming, data-structures and algorithms
- 3. Basics of Python language, to be able to use relevant libraries and toolkits for data analytics

2. Course Outcomes (COs)

Objective: In a computerized and networked society, vast amount of data is being collected every day in multiple domains. We are drowning in data, but starving for knowledge or actionable insights. Data mining



or data analytics constitute a collection of concepts and algorithms, which are being developed to answer "how" questions by extracting interesting and useful knowledge of from large data. Data analytics based platforms are being operated in multiple domains to extract valuable and actionable insights from the data to improve the business performance. The objective of this first level course is to learn the important concepts and algorithms related to data mining functionalities such as summarization, pattern mining, classification, clustering and outlier analysis.

The Course Outcomes (COs) are as follows:

- 1. After completing the course successfully, the students are able to
 - 1.CO-1. describe the concepts of data summarization, data warehousing, pattern mining, classification and clustering approaches
 - 2.CO-2. perform the task of data summarization, pattern mining, classification and clustering based on the requirement.
 - 3.CO-3. prescribe a single or a combination of data summarization, pattern mining, classification and clustering approaches for the problem scenario of a business/organization.
 - 4.CO-4. construct the improved data analytics methods for existing services.
 - 5.CO-5. formulate new data mining problems for creating new services and design the corresponding solutions

3. Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

	P O 1	P O 2	P O 3	P O 4	P O 5	P O 6	P O 7	P O 8	P O 9	PO 1 0	PO 11	PO 1 2	PS O 1	PS O 2	PS O 3	PS O 4
C 0 1	3	2	3	3	3	3	2	2	3	2	2	3	2	2	2	3
C O 2	3	1	3	1	3	2	1	1	3	3	2	3	2	1	2	3
C O 3	1	3	1	3	1	1	1	1	1	1	1	3	3	1	2	3
C O 4	1	2	1	3	3	1	1	1	3	1	2	3	3	1	2	3
C O 5	2	3	3	1	1	1	2	1	3	3	2	3	2	2	2	3

'3' in the box denotes 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping.



4. Detailed Syllabus

(please list the order in which they will be covered)

Unit 1: Introduction, data summarization through characterization, discrimination and data warehousing techniques (9 hours)

Unit 2: Concepts and algorithms for mining patterns and associations (9 hours) Unit 3: Concepts and algorithms related to classification and regression (9 hours) Unit 4: Concepts and algorithms for clustering the data (9 hours)

Unit 5: Outlier analysis and future trends. (3 hours)

• Five mini projects related to the above syllabus will be done by students in the laboratory

Reference Books and materials:

- 1. Book: Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Third edition, 2012, Elseiver Inc.
- 2. Book: Pang-Nong Tan, Michael Steinbach and Vipin Kumar, Introduction to Data Mining, 2006, Pearson Education.
- 3. Research Papers: About 25 research papers from the proceeding of the conferences and journals related to data summarization, data warehousing, pattern mining, classification, clustering, outlier detection.

5. Teaching-Learning Strategies in brief

Lectures by integrating ICT into classroom teaching, weekly tutorials involving problem solving and active learning by students and Project-based Learning by doing 5 mini-projects in laboratory by the students

6. Assessment methods and weightages in brief

Two Class Room tests: 10 marks; Mid Semester Examination in theory: 20 marks , End Semester Examination in Theory: 40 marks, Assessment of 5 mini projects in Laboratory: 30 marks

Title of the Course : Intro to Cognitive Science

Course Code : CS9.426



Course Information

Instructor Information

Faculty:Vishnu SreekumarTAs:NancyHada and KumarNeelabhDay/Time:Mondays and Thursdays: 2:00 pm – 3:25 pm.Virtual Office Hours:By appointment (please email).E-mail:vishnu.sreekumar@iiit.ac.in

First point of contact - TA emails:nancy.hada@research.iiit.ac.in; kumar.neelabh@research.iiit.ac.in

Course Information

Course Description:Cognitive Science is a highly interdisciplinary field of study that seeks to understand how the mind works. In this course, we will discuss a diverse range of perspectives from philosophy, linguistics, psychology, neuroscience, and computer science, on how to unravel the mysteries of human cognition.

Credits:4

L-T-P: 3-1-0 (L = lecture hours, T = tutorial hours, P = practical hours)

Prerequisite: None

Textbook & Course Materials

Recommended Texts & Other Readings: Lecture slides and supplementary readings will be posted to Moodle.

Course Technology Requirements

- 1. You will need access to the following tools to participate in this course.
 - o Laptop/desktop computer
 - o webcam
 - o microphone
 - o a stable internet connection (don't rely on cellular)

Course Structure

This course will be delivered fully in-person in a physical classroom unless COVID restrictions make us move online (Microsoft Teams).

Student Expectations

In this course you will be expected to complete the following types of tasks.

- 2. communicate via email
- 3. complete basic internet searches
- 4. download and upload documents to the course site on Moodle
- 5. read documents online
- 6. view online videos



- 7. participate in online discussions
- 8. complete quizzes/tests online
- 9. upload documents to a Dropbox/Moodle
- 10. participate in synchronous online discussions

Expected Instructor/TA Response Times

- We will attempt to respond to student emails within 24 hours. If you have not received a reply from us within 24 hours, please resend your email.
 - ***If you have a general course question (not confidential or personal in nature), please post it to the Course Q&A Discussion Forum found on the course homepage on Moodle. We will post answers to all general questions there so that all students can view them. Students are encouraged to answer each other's questions too.
- o We will attempt to reply to and assess student discussion posts within 48 hours.

Course Outcomes (COs)

After successful completion of this course, students will be able to:

- 11. CO-1: demonstrate familiarity with seminal research findings in cognitive science.
- 12. CO-2: read, interpret, critique, and evaluate research in cognitive science.
- 13. CO-3: critically think about the relationship between diverse fields such as AI, philosophy, neuroscience, and cognitive science.
- 14. CO-4: identify flaws in how scientific results are communicated and critique scientific work in terms of confounds, experimental design, etc.
- **15.** CO-5: appreciate the nature of scientific debate in cognitive science and be able to generate well-informed perspectives on these debates.

You will meet the outcomes listed above through a combination of the following activities in this course:

- 16. Attend lectures and participate in class discussions (CO-1, CO-2, CO-3, CO-4, CO-5)
- 17. Debate sessions (CO-1, CO-2, CO-3, CO-5)
- 18. Quiz 1, Quiz 2, mid-semester, and end-semester exams (CO-1, CO-2, CO-3, CO-5)
- 19. Complete a term paper/debate reaction paper (CO-1, CO-2, CO-3, CO-5)

Mapping of course outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs) – Course Articulation Matrix

P	Р	PO	PO	PO	PO	PO	PO	PO	PO	PO	PO	PS	PS	PS	PS
01	О2	3	4	5	6	7	8	9	10	11	12	01	O2	O3	O4



С	2	2	3	2	1	2	1	2	1	1	-	2	2	1	1	3
0																
C	1	2	1	3	1	1	1	1	1	1	1	2	1	2	1	3
0																
2		-	•	-		•	1				-	•	•		•	-
C	3	2	2	2	-	2	1	1	1	1	2	2	2	1	2	3
3																
С	1	3	2	1	-	1	1	2	1	2	-	2	1	2	2	3
0																
4	2	2	1	2	2	1	1	1	2	2		2	2	1	1	2
C	2	2	1	5	2	1	I	1	2	3	-	2	2	1	1	3
5																

Note: '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'level' mapping or a '-' dash mark if not at all relevant

List of topics and activities

- **20. Introduction**
- **21. Evolution of Cognitive Science**
- 22. A free-form discussion on consciousness
- 23. Empirical approaches in cognitive science
- 24. Brain: Organization; Intro to sensation and perception
- **25. Sensory systems**
- 26. Perception and Perceptual Learning, Cross-modal interactions
- 27. Vision
- 28. Attention
- **29.** Learning
- **30. Development**
- **31. Memory**
- **32. Language and Cognition**
- **33. Knowledge Representation**
- 34. Special topics: e.g. Music, mind, and technology
- 35. Several debate sessions with student debate teams

Grading Policies

Graded Course Activities

Description	Percentage



Quiz 1(10 marks)	10%
Quiz 2 (10 marks)	10%
Debate reaction paper or debate team participation (20 marks)	20%
Mid-Sem exam(20 marks)	20%
End semester exam (40 marks)	40%
Total (100 marks)	100

Quizzes

Quiz 1 will cover topics covered until Quiz 1, and Quiz 2 will cover topics taught between Quiz 1 and Quiz 2. They will contain mostly multiple choice questions.

Mid-semester exam (20 marks)

The mid-semester exam will cover all material taught up to that point, and may include both multiple choice and descriptive questions.

End semester exam (40 marks)

The end semester exam will cover material taught during the whole semester and will include both multiple choice and descriptive type questions.

Debate participation (20 marks = 10 marks for presenting + 10 marks for a short report)

We will reserve at least 3-4 lecture slots for student debates on contemporary issues in Cognitive Science. A list of representative topics are as follows:

- 1. Are there top-downinfluences on basic perception? Evidence for and against.
- 2. Do 3 year olds have a theory of mind?
- 3. Is cognition/consciousness a computational process?
- 4. Do we need representations for cognition?



Each debate team will have 3 members. They will read the recommended material for the chosen topic, and organize their arguments distributed across the 3 members. Each member gets 5 minutes to present their arguments (15 minutes per team). They may choose to use slides or not but the arguments must be clearly presented. At the end of both teams' presentations, each team gets 5 minutes for rebuttal when they can pick 2-3 claims made by the opposite team and present counterarguments.

The students participating in debate teams will only be required to write a short report but the remaining students will need to write a reaction paper to any one debate session OR write a term paper on any other topic that they choose (see next main section).

For debate team students (each person writes this separately without discussion with other team members, plagiarism software will be used to check your work), your short report should contain the following:

The paper will first summarize the problem (2 marks), and then summarize the arguments made by both sides (3 marks), and then will provide the student's OWN opinion about where they stand on the debate and what arguments were convincing to them (5 marks).

Recommended: 2-3 pages, font size 12, single-spaced.

The debate teams will be made on a first-come first-serve basis. TAs will open sign-up forms and make announcements on the course page on Moodle. It is important to check announcements on Moodle regularly for this reason.

Submission window for the short report: Nov 1-10

No extensions will be given because this is a wide window.

You are welcome to make multiple submissions within this window.

IMPORTANT: See the last section of this syllabus for policies about plagiarism. There will be no exceptions to those policies.

Term Paper or debate reaction papers for non-debate team students (20 marks)

- 1. Introduction and clarity of describing the background literature and specifying the nature of the problem 3 marks
- 2. Describing the different schools of thought that tackle the question -7 marks
- 3. Offer your own thinking on the matter (either siding with one school of thought, or offering a new insight or suggestions for experiments or investigations, providing appropriate justifications) 5 marks
- 4. Overall clarity, organization of thoughts, and originality 3 marks



5. Formatting (Citations, References) – 2 marks

Recommended: 8-10 pages, font size 12, single-spaced.

Submission window for the term paper/debate reaction paper: Nov 1-10

No extensions will be given because this is a wide window.

You are welcome to make multiple submissions within this window.

Participation

Students are expected to participate in all activities as listed on the course calendar. *Failure to* participate will result in students being unable to complete the term paper satisfactorily. The exams may also include questions from the in-class activities such as the debates and any resulting effect on the final grade is entirely the student's responsibility.

Complete Assignments

All assignments for this course will be submitted electronically through the course page on Moodle unless otherwise instructed. Assignments must be submitted by the given deadline or special permission must be requested from instructor before the opening of the submission window with documented evidence of an emergency.

Late or missing assignments will affect the student's grade.

Late Work Policy

Be sure to pay close attention to deadlines—there will be no make-up assignments or quizzes, or late work accepted without a serious and compelling reason and instructor approval.

Viewing Grades on Moodle

Points you receive for graded activities will be posted to the course page on Moodle. Click on the Grades link to view your points.

Letter Grade Assignment

Final grades assigned for this course will be based on the percentage of total points earned and are assigned as follows:

Letter Grade	Percentage
А	[92,100]%
A-	[84,92)%
В	[76,84)%
В-	[68,76)%



С	[60,68)%
C-	[52,60)%
D	[45,52)%
F	< 45%

IMPORTANT NOTE:[x,y) indicates that x is included (square bracket) in the range and y is not (curly bracket). The normal rules of rounding will apply: So if you get 75.5, it will be rounded to 76 and you will get a B. However, if you get 75.444, it can only be rounded downwards and hence the final grade will be B-. No disputes on this matter will be entertained and such emails will not get a response.

Course Policies

Netiquette Guidelines

Netiquette is a set of rules for behaving properly online. Your instructor and fellow students wish to foster a safe online learning environment. All opinions and experiences, no matter how different or controversial they may be perceived, must be respected in the tolerant spirit of academic discourse. You are encouraged to comment, question, or critique an idea but you are not to attack an individual. Working as a community of learners, we can build a polite and respectful course community.

The following netiquette tips will enhance the learning experience for everyone in the course:

- 1. Do not dominate any discussion.
- 2. Give other students the opportunity to join in the discussion.
- 3. Do not use offensive language. Present ideas appropriately.
- 4. Be cautious in using Internet language. For example, do not capitalize all letters since this suggests shouting.
- 5. Avoid using vernacular and/or slang language. This could possibly lead to misinterpretation.
- 6. Never make fun of someone's ability to read or write.
- 7. Share tips with other students.
- 8. Keep an "open-mind" and be willing to express even your minority opinion. Minority opinions have to be respected.
- 9. Think and edit before you push the "Send" button.
- 10. Do not hesitate to ask for feedback.
- 11. Always assume good intentions and ask for clarification. Communication online is difficult without facial and gestural cues.

Adapted from:



Mintu-Wimsatt, A., Kernek, C., &Lozada, H. R. (2010). *Netiquette: Make it part of your syllabus*. Journal of Online Learning and Teaching, 6(1). Retrieved from <u>http://jolt.merlot.org/vol6no1/mintu-wimsatt_0310.htm</u>

Shea, V. (1994). Netiquette. Albion.com. Retrieved from: http://www.albion.com/netiquette/book/.

Build Rapport

If you find that you have any trouble keeping up with assignments or other aspects of the course, make sure you let your instructor know as early as possible. As you will find, building rapport and effective relationships are key to becoming an effective professional. Make sure that you are proactive in informing your instructor when difficulties arise during the semester so that we can help you find a solution.

Inform Your Instructor of Any Accommodations Needed

If you have a documented disability and wish to discuss academic accommodations, please contact your instructors as soon as possible.

Statement of Policy

The instructors of this course will modify requirements as necessary to ensure that they do not discriminate against qualified students with disabilities. The modifications should not affect the substance of educational programs or compromise academic standards; nor should they intrude upon academic freedom. Examinations or other procedures used for evaluating students' academic achievements may be adapted. The results of such evaluation must demonstrate the student's achievement in the academic activity, rather than describe his/her disability.

If modifications are required due to a disability, please inform the instructor

Commit to Integrity

As a student in this course (and at IIIT Hyderabad) you are expected to maintain high degrees of professionalism, commitment to active learning and participation in this class and also integrity in your behavior in and out of the classroom.

IIIT Hyderabad Academic Honesty Policy & Procedures

Student Academic Disciplinary Procedures

- (1) Academic misconduct is an act in which a student:
- (a) Seeks to claim credit for the work or efforts of another without authorization or citation;
- (b) Uses unauthorized materials or fabricated data in any academic exercise;
- (c) Forges or falsifies academic documents or records;
 - (d) Intentionally impedes or damages the academic work of others;
- (e) Engages in conduct aimed at making false representation of a student's academic performance; or



(f) Assists other students in any of these acts.

(2) Examples of academic misconduct include, but are not limited to: cheating on an examination; collaborating with others in work to be presented, contrary to the stated rules of the course; submitting a paper or assignment as one's own work when a part or all of the paper or assignment is the work of another; submitting a paper or assignment that contains ideas or research of others without appropriately identifying the sources of those ideas; stealing examinations or course materials; submitting, if contrary to the rules of a course, work previously presented in another course; tampering with the laboratory experiment or computer program of another student; knowingly and intentionally assisting another student in any of the above, including assistance in an arrangement whereby any work, classroom performance, examination or other activity is submitted or performed by a person other than the student under whose name the work is submitted or performed.

We will be using plagiarism detection software. Please do not copy-paste from other papers. If you use direct quotes, you have to use the quotation marks "xyz" and cite your source: e.g. (Johnson & Johnson, 1988, p. 5). Please use APA format. If plagiarism is detected, for the first violation, you will get 0 for the term paper or assignment in question. If plagiarism is detected a second time in another assignment/project write-up, then one letter grade will be deducted from the final grade (e.g if you get a B/B-, that will be changed to C/C-) and you will be reported to the appropriate authorities for further disciplinary action.

Note: This syllabus was adapted from a template provided at <u>www.uwsp.edu</u>

Title of the Course	: Spatial Data Science
Faculty	: K S Rajan
Course Code	: (PG-2 level course)
Credits	: 4
L - T – P	: 3-1-0

(L - Lecture hours, T-Tutorial hours,

P - Practical hours)

Semester, Year : Spring 2022

(Ex: Spring, 2022)

Name of the Program : Open to All Programs on Campus at UG, PG/PhD Level

Pre-Requisites: Basic understanding of Locational Data and Computing – Any UG3, UG4, M.Tech., MS, and Ph.D. student should be able to take it.

Prior course work in Spatial Informatics may help.

٠

Course Outcomes



CO-1: Describe how Spatial Data Science helps uncover patterns CO-2: Apply Geospatial techniques to Prepare the data for analysis

CO-3: Analyze the spatial and temporal data and interpret its outcomes

CO-4: Assessment of application of Spatial data science in key domain areas

CO-5: Design research projects that helps synthesize the learning into an application

Course Topics :

Module 1: Introduction to Spatial Data Science

- What is special about Spatial Data and Geo-AI?
- How Spatial and Spatio-temporal Big Data helps uncover patterns?
- Spatial Data Handling including spatial data models, data formats
- Challenges to computing approaches when applied to Spatial Data Effects of Topology

Module 2: Geospatial Data Analysis and Modelling

- Vector Data Spatial Analysis
- Raster Data Spatial Analysis
- How to use temporal data in conjunction with Spatial data
- GeoSpatial Data Modelling Module 3: Spatial Sciences
- Spatial Statistics including Spatial auto-correlation, Spatial tessellation

o Data Mining applications on Spatial data including Spatio-temporal Data Mining

- Network Analysis and Graph theory
- Few relevant topics from Computational Geometry
- Geovisualization Maps to WebGIS Module 4: Spatial Classification and Prediction
- Spatial decision trees
- Machine learning as applied to Spatial Data including Spatial-aware Neural Networks
- Hotspot Analysis
- Spatial Outliers detection

Module 5: Applications of Spatial Data Science

- Public Health monitoring and mapping diseases, risk analysis and disease spread modelling
- Agriculture crop growth monitoring, crop yield patterns and resource constraints
- Location based services routing applications, ride-sharing algorithms, optimal location

Preferred Text Books :

- 1. Spatial Computing, By Shashi Shekar and Pamela Vold. The MIT Press. 2020
- 2. GIS A computing perspective. By Micheal Worboys and Matt Duckham. CRC Press; 2nd edition 2004
- 3. Spatial Databases: A Tour. By S. Shekhar and S. Chawla, Prentice Hall, 2003, ISBN 013-017480-7.
- 4. Selected Research Papers and Articles (will be shared with the topics taught on the course portal)



Reference Books

:

1. Geographical Data Science and Spatial Data Analysis - An Introduction in R. By Lex Comber and Chris Brunsdon. SAGE Publications Ltd. 2020

E-book Links

: Will be provided in Class as appropriate

Grading Plan :

Type of Evaluation	Weightage (in %)
Class Quizzes	15.0
Mid Sem Exams – 2	20.0
End Sem Exam	30.0
Paper reviews and Presentations by each Student in Class	10.0
Project/Term paper demonstrating the Practical applications	25.0

Mapping of Course Outcomes to Program Objectives:

	Р О 1	P O 2	Р О З	Р О 4	Р О 5	P O 6	Р О 7	P O 8	P O 9	PO1 0	PO1 1	PO 1 2	PSO 1	PSO 2	PSO 3	PSO 4
C 0 1	3	1	1	3	-	-	-	1	2	-	-	3	2	-	-	-
C O 2	2	-	-	-	3	-	-	-	2	-	-	-	2	-	2	2
C O 3	3	2	-	-	3	-	-	-	2	-	-	-	2	2	3	2
C O 4	3	2	3	3	2	2	-	3	2	3	-	3	3	2	2	3
C O 5	3	3	3	3	2	3	1	3	3	3	2	3	2	3	2	3

Teaching-Learning Strategies in brief (4-5 sentences) :

Teaching - Learning



Lectures Guest Lectures

Reading research papers

Class participation in Q&A, discussions Online discussions over MS Teams

Learning by doing

Short Presentation and Discussion led by Student Course project on conceptualization and implementation Real world applications

Multi-disciplinary approach

Note: This course description format comes into effect from Spring 2022.

Title of the Course	: Real-Time Embedded Systems						
Faculty Name	: Deepak Gangadharan						
Course Code	: CS3.502						
L-T-P	: 3-1-0						
Credits	:4						
(L=Lecture hours, T=Tutorial hours,							

P=Practical hours)

Name of the Academic Program: B-Tech in Computer Science and Engineering

1. <u>Prerequisite Course/Knowledge</u>

Computer Systems Organization, Basics of Operating Systems

2. Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO-1. Explain the features of real-time systems and classify different types of real-time systems such as hard real-time, soft-real time based on the timing requirements.

CO-2. Apply an appropriate task model (such as periodic, sporadic, aperiodic, etc) based on task/application characteristics to model a real-time system.

CO-3. Analyze the schedulability of a real-time system with different types of scheduling algorithms (static vs dynamic, preemptive vs non-preemptive) on a uniprocessor

CO-4. Analyze the schedulability of a real-time system with different types of scheduling algorithms (global, partitioned, semi-partitioned) on a multiprocessor platform CO-5. Analyze the schedulability of a real-time system with shared resources

CO-6. Assess the theory and experimental results presented in a relevant research paper and present it.



CO-7. Develop scheduling algorithms in a RTOS simulator

3. <u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific</u> <u>Outcomes (PSOs) – Course Articulation Matrix</u>

	Р 01	Р О2	Р О3	Р О4	Р 05	Р Об	Р О7	P 08	Р 09	PO 10	PO 11	PO 12	PS 01	PS O2	PS O3	PS O4
C 01	1	1	1	1	-	1	-	-	-	2	-	3	1	3	2	2
C O2	3	3	1	2	1	1	2	1	-	1	-	3	3	1	2	2
C 03	3	3	-	3	1	2	2	1	-	1	-	3	3	3	2	3
C O4	3	3	-	3	1	2	2	1	-	1	-	3	3	3	2	3
C 05	3	3	-	3	1	2	2	1	-	1	-	3	3	3	2	3
C 06	1	3	-	1	-	1	1	-	-	3	-	3	2	1	-	3
С 07	3	2	-	2	3	2	1	2	3	1	-	3	3	2	1	2

Note: Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs) and PSOs.

Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

4. Detailed Syllabus

Unit 1: Real-Time Systems – Introduction and Concepts, Modeling Real-Time Systems Unit 2: Commonly used approaches to Real-Time Scheduling – Clock Driven approach, Weighted Round Robin approach, Priority Driven Approach, Dynamic vs Static Systems, Offline vs Online Scheduling, Preemptive vs Non-Preemptive

Unit 3: Clock Driven Scheduling – Scheduling Aperiodic and Sporadic Jobs, Schedulability test

Unit 4: Priority Driven Scheduling – Static Priority: Rate Monotonic and Deadline Monotonic Algorithms, Dynamic Priority: EDF Algorithm, Schedulability tests

Unit 5: Scheduling Aperiodic and Sporadic jobs in Priority Driven Systems – Deferrable Server, Sporadic Server, Constant Utilization Server, Total Bandwidth Server and Weighted Fair Queuing Server

Unit 6: Multiprocessor Scheduling



Unit 7: Resources and Resource Access Control

Reference Books:

- 1. Jane W S Liu, Real-Time Systems, Pearson Education
- 2. Giorgio C Buttazo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, 3rd edition, Springer
- 3. C.M. Krishna & Kang G. Shin , Real Time Systems, McGraw Hill

5. Teaching-Learning Strategies in brief

Weekly lectures cover the topics in the syllabus and the advanced topics from research in real-time systems. Tutorials cover how to solve some design and analysis problems related to topics covered in the lectures. There are couple of assignments that will provide the students experience in programming schedulers for RTOS platforms. There is a project which is either based on an idea the student wants to explore from the course topics or based on an existing research paper implementation and evaluation. Finally, there will be a presentation/discussion of a research paper.

6. Assessment methods and weightages in brief

Type of Evaluation	Weightage (in %)
Quizzes	15
Assignments	15
End Sem Exam	20
Project	30
Research Paper Presentation	20

Title of the Course: Mechatronics System DesignL-T-P: 3-1-0Credits: 4(L= Lecture hours, T=Tutorial hours,
P=Practical hours)

1. Prerequisite Course / Knowledge:

Basic programming (Python, C++), Linear Algebra, Numerical methods, Basic microcontroller knowledge.



2.<u>Course Outcomes (COs</u>):

After completion of this course successfully, the students will be able to CO-1 Describe important elements of mechatronics system

CO-2 Apply the previous knowledge of microcontroller programming for controlling multidisciplinary mechatronic systems.

CO-3 Describe and design basic mechanical elements and their feedback control. CO-4 Synthesize and analyze a range of mechanisms.

CO-5 Design and execute a multidisciplinary project based on the given specifications as part of a team.

3.<u>Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes</u> (PSOs) – Course Articulation Matrix

	Р О1	Р О2	Р О3	Р О4	Р О5	Р Об	Р О7	Р 08	Р 09	PO 10	PO 11	PO 12	PS 01	PS O2	PS O3	PS O4
C 01	3	2	3	1	1	1	2	2	1	2	1	3	3	2	2	2
C O2	3	3	2	1	3	1	1	1	3	1	2	3	3	2	2	3
C 03	3	2	2	1	2	1	1	1	1	2	1	2	1	1	1	3
C 04	3	3	3	1	2	1	1	1	1	2	1	3	1	1	1	3
C 05	3	3	3	2	3	2	2	1	3	3	2	3	1	2	3	3

4. <u>Detailed Syllabus</u>:

Unit 1: Sensors and Actuators:

Sensors for robotics application - position, speed, acceleration, orientation, range. Actuators - general characteristics, motors, control valves.

Unit 2: Computer based feedback control:

Sampled data control, sampling and hold, PID control implementation, stability, bilinear transformation.

Unit 3: Introduction to mechanical elements and transformations, basic concepts of kinematics and dynamics.

Unit 4: Design and analysis of mechanisms.



Unit 5: Programming and hardware experiments.

Reference Books:

- 1. Bentley, John P. "Principles of measurement systems," Pearson education, 2005.
- 2. D.R. Coughanowr, "Process system analysis and control," McGraw Hill, 1991
- 3. G.F. Franklin, J.D. Powell and M.L. Workman, "Digital control of dynamic systems", Addison Wesley, 3rd edition, 1998.
- 4. Hartenberg, R., & Danavit, J, "Kinematic synthesis of linkages," McGraw Hill, 1964.
- 5. http://wiki.ros.org/
- 6. User manual of microcontroller and data sheets of sensors and actuators

5. Teaching-Learning Strategies in brief:

This course aims to teach the students about designing and developing a mechatronics system by providing them with essential hardware and software. Part of the class is devoted to a learn-by-doing lesson where the students will learn theory and get hands-on experience with various aspects of the mechatronics system.

The goal for the students is to design, build, and debug the electromechanical system for a given task as a part of the course project.

6. Assessment methods and weightages in brief:

Mid semester exam 20%

Assignments 40%

The class work assignments will be based on the application of a step-by-step engineering design process to a problem assigned in the course.

Project 40% Proposal (5%) Project demonstration (25%) Final report (10%)



List of Electives

Advanced Algorithms Advanced Data Systems Advanced NLP (100) Advanced Optimization: Theory and Applications Advances in Robotics & Control Algorithms and Operating Systems Analog IC Design **Applied Electromagnetics** Behavioral Research & Experimental Design Behavioral Research: Statistical Methods CMOS Radio Frequency Integrated Circuit Design **Cognitive Neuroscience** Cognitive Science and AI **Computational Social Science** Data Foundation Systems Design for Social Innovation Design for Testability **Differential Equations** Digital Image Processing Digital VLSI Design **Distributed Data Systems** Distributing Trust and Block Chains **Eco-Informatics** Environmental Science & Technology Fairness, Privacy and Ethics in AI Flexible Electronics FPGA based Accelerator Design **Functional Analysis** Green Buildings Hydro Informatics ICTs for Development Information Security Audit and Assurance Internals of Application Servers Intro to UAV Design Introduction to Game Theory Introduction to Neural and Cognitive Modeling Introduction to Neuroeconomics Introduction to NLP Linear Partial Differential Equations and Variational Calculus Mobile Robotics Multivariate Analysis Music, Mind, and Technology

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Online Privacy Optical Remote Sensing Principles of Semiconductor Devices Quantum Algorithms Radar Systems Research in Information Security Robotics: Dynamics and Control **Robotics: Planning and Navigation** Signal Detection and Estimation Theory Social Science Perspective on HCI Speech Signal Processing System and Network Security Technology Product Entrepreneurship Time Frequency Analysis Topics in Applied Optimization Topics in Deep Learning Topics in Machine Learning **Topics in Signal Processing** Topics in Software Engineering Topics in Software Foundations Wireless Communications