PONDICHERRY UNIVERSITY
(A CENTRAL UNIVERSITY)

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

B.Sc. Computer Science (Honors)

B.Sc. Computer Science (Honors with Research)

REGULATIONS, CURRICULUM & SYLLABUS

(Under the National Education Policy 2020)

Effective from the Academic Year 2023 - 2024

Revised in June 2024

Board of Studies - Computer Science

S.No.	Name	Affiliation	BoS
1	Dr. S.K.V. Jayakumar	Dept. of CS, Pondicherry University	Chairman
2	Dr. M. Punithavalli	Dept. of CA, Bharathiar University	External Members
3	Dr. P.S. Grover	Dept. of CS, Delhi University	
4	Dr. M. Sornam	Dept. of CS, University of Madras	Special Invitee
5	Dr. R. Subramanian	Dept. of CS, Pondicherry University	Members of Board
6	Dr. T. Chithralekha	Dept. of CS, Pondicherry University	
7	Dr. S. Bhuvaneswari	Dept. of CS, Pondicherry University	
8	Dr. R. Sunitha	Dept. of CS, Pondicherry University	
9	The Head, Dept. of CS	Dept. of CS, Achariya Arts & Sci. College	
10	The Head, Dept. of CS	Dept. of CS, Idhaya College of Arts & Sci.	
11	The Head, Dept. of CS	Dept. of CS, Avvaiyar Govt. College	
12	The Head, Dept. of CS	Dept. of CS, Don Bosco College	
13	The Head, Dept. of CS	Dept. of CS, PSV College of Arts & Sci.	
14	The Head, Dept. of CS	Dept. of CS, Rajiv Gandhi Arts & Science	

NEP Committee

S.No.	Name	Designation	Affiliation
1	Prof. S. K.V. Jayakumar	Professor and Head	Dept. of CS, Pondicherry University
2	Prof. T. Chithralekha	Professor	
3	Prof. P. Sujatha	Professor	
4	Dr. R. Sunitha	Associate Professor	
5	Dr. V. Uma	Associate Professor	
6	Dr. K. S. Kuppusamy	Associate Professor	
7	Dr. M. Sathya	Assistant Professor	

Syllabus Revision Committee (Computer Science)

S.No.	Name	Designation	Affiliation
1	Prof. S. K.V. Jayakumar	Professor and Head	Dept. of Computer Science, Pondicherry University
2	Prof. K. Suresh Joseph	Professor	
3	Dr. M. Sathya	Asst. Professor	

Table of Contents

1. PREAMBLE & PROGRAMME OUTCOMES

1.1. Preamble

The Bachelor of Science (B.Sc.) programme in Computer Science is a dynamic and comprehensive academic journey designed to equip students with a strong foundation in the principles and practices of computing. Rooted in the ever-evolving field of technology, this programme is crafted to cultivate a deep understanding of computer science theories, algorithms, and applications.

The curriculum encompasses a balanced blend of foundational courses and specialized electives on experiential learning, offering opportunities for internships, industry projects, and participation in coding competitions. Students will engage in practical applications of their knowledge, honing their skills through hands-on experiences that mirror the challenges and demands of the rapidly evolving technological landscape.

Recognizing the global nature of technology, the B.Sc. in Computer Science incorporates an international perspective. Students will explore global technology trends, multicultural influences, and ethical considerations, preparing them to contribute responsibly to the global digital community.

The B.Sc. in Computer Science at Pondicherry University is a transformative educational experience that empowers students to become adept problem solvers, innovators, and leaders in the field of computer science. By fostering a passion for continuous learning and providing a solid foundation in both theory and application, the programme sets the stage for a successful and fulfilling career in the dynamic world of technology.

1.2 Programme Outcomes

Upon completion of the Bachelor of Science (B.Sc.) programme in Computer Science, students will demonstrate the following outcomes at:

UG Certificate Level

- Acquire foundational knowledge in computer science.
- Demonstrate basic skills in problem-solving and programming.

UG Diploma Level

- Develop intermediate-level knowledge and skills in computer science.
- Apply problem-solving and programming concepts to practical scenarios.

UG Degree Level

- Attain advanced knowledge and skills in computer science.
- Demonstrate proficiency in problem-solving, programming, and system design.

UG Degree with Honors / Honors with Research

- Demonstrate proficiency in programming languages and software development.
- Apply principles of data structures and algorithms to solve complex problems.
- Design and implement efficient solutions for real-world computing challenges.
- Exhibit effective communication skills in conveying technical concepts orally and in writing.
- Engage in collaborative projects and demonstrate the ability to work effectively in a team.
- Apply ethical considerations in professional and societal contexts related to computer science.
- Possess a comprehensive understanding on their Specialization in Computer Science and in the chosen specialization.
- Exhibit a commitment to lifelong learning and adaptability to evolving technologies.

2. DEFINITIONS

Terms used in the NEP Regulations shall have the meaning assigned to them as given below unless the context otherwise requires:

A. Credit: "Credit" is a unit by which the coursework is measured. It determines the number of hours of instruction required per week during a semester (Minimum 15 weeks). One credit is equivalent to 15 hours of teaching (lecture and /or tutorial) or 30 hours of practical and/or field work or community engagement and service per semester.

B. Academic Year: "Academic Year" means the year starting in the month of June and ending in the succeeding month of May.

C. Semester: "Semester" means 15-16 weeks of teaching-learning session of which two weeks shall be set apart for examination and evaluation; A semester comprises 90 working days and an academic year is divided into two semesters.

D. Summer Term: "Summer term" is for 8 weeks during summer vacation. Internship/apprenticeship/work based vocational education and training can be carried out during the summer term, especially by students who wish to exit after two semesters or four semesters of study.

E. Grade: "Grade" means a letter grade assigned to a student in a Course for her/his performance at academic sessions as denoted in symbols of: O(outstanding), A+(Excellent), A(Very good), B+(Good), B(Above average), C(Average), P(Pass) F(Fail) and Ab(Absent) with a numeric value of O=10, A+=9, A=8, B+=7, B=6, C=5 P=4, and F=0, Ab=0;

F. SGPA: "Semester Grade Point Average (SGPA)" is computed from the grades as a measure of the students' performance in a given semester.

G. CGPA: "Cumulative GPA (CGPA)" is the weighted average of all courses the student has taken in a given Programme;

H. Programme: "Programme" means a set of Courses that allows a student to structure and study to attain the status of being admitted to a Degree/Diploma of the University;

I. Programme Committee: "Programme Committee" means an Academic Committee constituted by the University for the purpose of conducting an Academic Programme;

J. Credit Requirement: "Credit Requirement" for a Degree/Diploma/Certificate Programme means the minimum number of credits that a student shall accumulate to achieve the status of being qualified to receive the said Degree, Diploma/Certificate as the case may be;

K. Exit Option: "Exit option" means the option exercised by the students, to leave the Programme at the end of any given Academic year;

L. Lateral Entry: "Lateral entry" means a student being admitted into an ongoing Programme of the University other than in the 1st year of the programme.

3. DURATION, ELIGIBILITY & AWARD OF UG DEGREE/DIPLOMA/CERTIFICATE

3.1. Duration of the Programme

The duration of the UG programme is 4 years or 8 semesters. Students who desire to undergo a Three-year UG Programme will be allowed to exit after completion of the 3rd year. If a student wants to leave after the completion of the first or second year, the student will be given a UG Certificate or UG Diploma, respectively, provided they secure the prescribed number of credits (as given in table below).

Students who exit with a UG certificate or UG diploma are permitted to re-enter within three years and complete the degree programme. Students may be permitted to take a break from the study, they

are allowed to re-enter the degree programme within 3 years and complete the programme within the stipulated maximum period of seven years.

3.2. Eligibility

Senior Secondary School Leaving Certificate or Higher Secondary (12th Grade) Certificate obtained after successful completion of Grade 12 or equivalent stage of education corresponding to Level-4 (Levels in NHEQF). **For detailed eligibility, refer the Admissions and Lateral Entry Section below.**

3.3. Awarding of UG Certificate, UG Diploma and Degrees Nomenclature

Four years B.Sc. Degree Programme shall have options for earning a Certificate / Diploma / UG Degree / UG Degree (Honors) / UG Degree (Honors with Research) based on the exit option exercised by the candidates.

3.3.1. UG Certificate

Students who opt to exit after completion of the first year (2 Semesters) and have earned a minimum of 42 credits will be awarded a UG certificate if, in addition, they complete work based vocational course / internship of 4 credits during the summer vacation of the first year.

3.3.2. UG Diploma

Students who opt to exit after completion of the second year (4 Semesters) and have earned a minimum of 84 credits will be awarded the UG diploma if, in addition, they complete work based vocational course / internship of 4 credits during the summer vacation of the second year.

3.3.3. Three-year UG Degree

Students who wish to discontinue after the 3-year (6 Semesters) UG programme will be awarded a UG Degree in Computer Science after successful completion of three years, earning a minimum of 124 credits and satisfying the minimum credit requirements as mentioned in Table1 below.

3.3.4. Four-year UG Degree (Honors)

A four-year UG Honors degree in the Computer Science will be awarded to those who complete a four-year (8 Semesters) degree programme, earning a minimum of 164 credits and have satisfied the credit requirements as mentioned in Table1 below.

3.3.5. Four-year UG Degree (Honors with Research)

Students who secure a minimum of 7.5 CGPA in the first six semesters and wish to undertake research at the undergraduate level can choose a research stream in the fourth year. They should do a research project or dissertation under the guidance of a faculty member of the University.

The research project/dissertation will be in the major discipline, Computer Science. The students who secure a minimum of 164 credits, including 12 credits from a research project/dissertation, will be awarded UG Degree in Computer Science (Honors with Research).

3.3.6. Programme overview

As per the guidelines of NEP, students are mandated to complete 124 credits to complete a basic Bachelor's Degree in 3 years. With an additional 40 credits of course work one can pursue 4th Year Honors or Honors with Research Degree. The UG Programme will consist of the following categories of courses and the minimum credit requirements for 3-year UG and 4-year UG(Honors) or UG (Honors with Research) programmes are given below.

Table 1: Breakup of Credits and Courses – Minimum Requirements

S.No.	Component	3 Year UG	4 Year UG (Honors/ Honors with research)
1	Major Disciplinary - Computer Science	60 Credits (15 Courses of 4 credits)	80 Credits (20 Courses of 4 credits)
2	Minor Disciplinary – Specialization Courses	24 Credits (6 Courses of 4 Credits)	32 Credits (8 Courses of 4 credits)
3	Multi-Disciplinary Courses	9 Credits (3 courses of 3 credits)	9 Credits (3 courses of 3 credits)
4	Ability Enhancement Courses	12 Credits (4 courses of 3 credits)	12 Credits (4 courses of 3 credits)
5	Skill Enhancement Course – On the chosen Specialization	9 Credits (3 courses of 3 credits)	9 Credits (3courses of 3 credits)
6	Value-Added Courses	8 Credits (4 courses of 2 credits)	8 Credits (4 courses of 2 credits)
7	Summer Internship	4 Credits (Included in Major courses of 60 credits)	4 Credits (Included in Major courses of 80 credits)
8	Community Engagement and Service	2 Credits (1 course)	2 Credits (1 course)
9	Research Dissertation Project	-	12 Credits
	Total	**124**	**164**

Note: Honors students not undertaking research will do 3 courses for 12 credits in lieu of a Research Project / Dissertation.

3.3.7. Degree and Nomenclature

Candidates who complete Eight semesters and earn a minimum of 164 credits and have satisfied the credit requirements as mentioned in the table below will be awarded either of the following degrees.

- B.Sc. Computer Science*
- B.Sc. Computer Science (Honors) #
- B.Sc. Computer Science (Honors with Research) ##

* for candidates who wish to exit at the end of third year with 124 credits earned and satisfied the other minimum requirements given in 3.3.9.

\# for candidates who complete 3 theory courses (MJD 21, MJD 22, and MJD 23) instead of the research project work in the Eighth Semester

for candidates who complete a research project work in the Eighth Semester

3.3.8. Degree with Specialization

Out of the above said 164 credits (Table1) the candidates shall earn 107 credits (87 credits out of 124 credits in the case of 3-years UG) from the Hardcore courses (Major Disciplinary, Multi-disciplinary, Ability Enhancement, Value added Courses and Community Engagement and Service) and the remaining 57 credits (37 credits in the case of 3 year UG) shall be earned from the subjects they choose to study from the list of softcore courses. These 61 credits shall be earned through studying the specialization courses in Minor Disciplinary – Specialization Courses, Skill Enhancement Courses in all the semesters and the Research Project or the Courses the candidates choose to study in the Eighth Semester. The Programme Structure is detailed in the following figure.

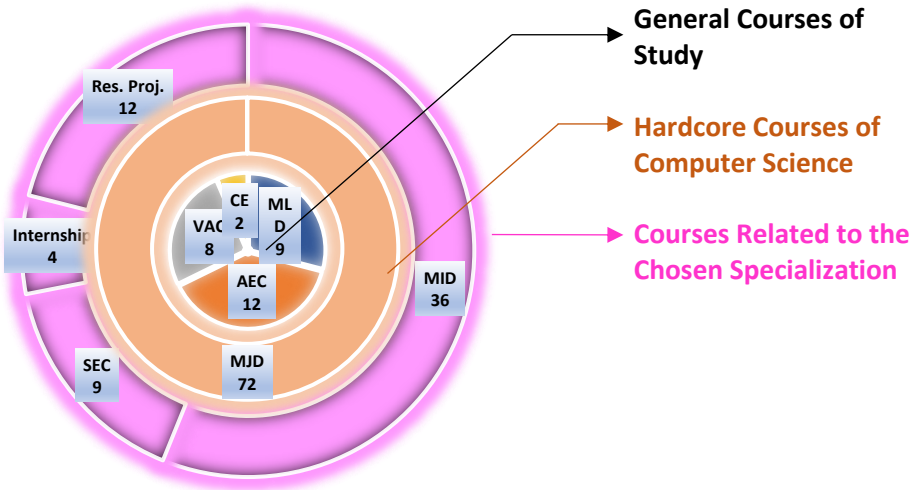

Figure 1: Programme Structure with credit breakup

3.3.9. Exit Options and Nomenclature of Certificate, Diploma

Candidates can exercise the following exit options and obtain the said certificate or diploma or degree, if the minimum required credits are earned and other conditions are met. Students exercising the option of exit at the end of 2nd semester or 4th semester need to have completed an internship for atleast 8 weeks along with the necessary credit requirements to qualify for the relevant certificate or diploma. In any case, every student, whenever exit (or complete the 4 year programme), should have completed atleast one internship for a minimum period of 8 weeks.

Exit after 2nd Semester: Certificate in Problem Solving and Programming will be awarded for candidates who exit the course at the end of 2nd semester and earned a minimum of 42 credits and have completed a Summer Internship of 4 credits for a minimum period of 8 weeks, during the summer vacation post 2nd semester.

Exit after 4th Semester: Diploma in Computer Science will be awarded for candidates who exit the course at the end of 4th semester and earned a minimum of 84 credits and have completed a Summer Internship of 4 credits for a minimum period of 8 weeks, during the summer vacation post 4th semester.

Exit after 6th Semester: UG Degree in Computer Science (B.Sc. (CS)) will be awarded for candidates who exit the course at the end of 6th semester and earned a minimum of 124 credits and have completed a Summer Internship of 4 credits for a minimum period of 8 weeks, during the summer vacation post 4th semester.

Exit after	Credits and other requirements	Awards
2nd Semester	Min: 42 Credits & Internship	Certificate in Problem Solving and Programming
4th Semester	Min: 84 Credits & Internship	Diploma in Computer Science
6th Semester	Min: 124 Credits & Internship	B.Sc. Computer Science

4. STRUCTURE OF THE UNDERGRADUATE PROGRAMME

This B.Sc Honors programme is offered by the University shall confirm to the structure specified hereunder. As per the decided programme mandate, the students to complete 124 credits to complete a basic Bachelor's Degree in 3 years. With an additional 40 credits of course work one can pursue 4th Year Honors or Honors with Research Degree. The UG Programme will consist of the categories of courses and the minimum credit requirements for 3-year UG and 4-year UG(Honors) or UG (Honors with Research) programmes as given in Table1 at section 3.3.6.

4.1. Types of Courses

Hardcore Courses	Softcore Courses (Specialization specific)
Major Disciplinary - Computer Science	Minor Disciplinary
Multi-Disciplinary Courses	Skill Enhancement Courses
Ability Enhancement Courses	Summer Internship
Value Added Courses	Research Dissertation Project
Community Engagement and Service	

4.2. Description of Courses

The following are the types of courses in this programme:

4.2.1. Major Discipline: 60 Credits - 3 Year UG, & 72 Credits - 4 Year UG

Major discipline here means to Computer Science. Students should secure the prescribed number of credits (not less than 50% of the total credits) through core courses in the major discipline. The major discipline would provide the opportunity for a student to pursue in-depth study of a particular subject or discipline. A student may choose to change the major discipline within the broad discipline at the end of the second semester provided all the prerequisites of the respective degree programme are fulfilled.

4.2.2. Minor Discipline / Specialization: 24 Credits - 3 Year UG, & 40 Credits - 4 Year UG

Minor discipline helps a student to gain a broader understanding beyond the major discipline. For example, if a student pursuing this programme obtains the required 40 credits for a 4 year UG (24 credits for 3 Year UG) from a bunch of courses in Artificial Intelligence & Machine Learning, (along with the Skill Enhancement Courses chosen from the same specialization, and other conditions met), then the student will be awarded B.Sc. degree in Computer Science with a Specialization in Artificial Intelligence & Machine Learning. 12 credits of these courses shall be allotted to vocational courses.

4.2.3. Multidisciplinary courses (MD): 9 Credits

All UG students are required to undergo 3 introductory-level courses relating to any of the broad disciplines given below. These courses are designed and developed by every other department for the benefit of this discipline students and are pooled by under 5 baskets for students to choose any 3 courses from 3 broader areas (one each from any three broad areas) from the basket. Students are not allowed to choose or repeat courses already undergone at the higher secondary level (12th class) under this category. Detailed list is given in the curriculum.

4.2.3.a. Natural and Physical Sciences

Students can choose basic courses from disciplines such as Natural Science, for example, Biology, Botany, Zoology, Biotechnology, Biochemistry, Chemistry, Physics, Biophysics, Astronomy and Astrophysics, Earth and Environmental Sciences, and other related subjects.

4.2.3.b. Mathematics, Statistics, and Computer Applications

Courses under this category will facilitate the students to use and apply tools and techniques in their major and minor disciplines. The course may include training in programming software like Python

among others and applications software like STATA, SPSS, Tally and similar others. Basic courses under this category will be helpful for science and social science in data analysis and the application of quantitative tools.

4.2.3.c. Library, Information, and Media Sciences

Courses from this category will help the students to understand the recent developments in information and media science (journalism, mass media, and communication)

4.2.3.d. Commerce and Management

Courses include business management, accountancy, finance, financial institutions, fintech and other related subjects.

4.2.3.e. Humanities and Social Sciences

The courses relating to Social Sciences, for example, Anthropology, Communication and Media, Economics, History, Linguistics, Political Science, Psychology, Social Work, Sociology and other related subjects will enable students to understand the individuals and their social behaviour, society, and nation. Students be introduced to survey methodology and available large-scale databases for India. The list of Courses that can include interdisciplinary subjects such as Cognitive Science, Environmental Science, Gender Studies, Global Environment & Health, International Relations, Political Economy and Development, Sustainable Development, Women's and Gender Studies and similar subjects. will be useful to understand society.

4.2.4. Ability Enhancement Courses (AEC): 12 credits

Modern Indian Language (MIL) & English language focused on language and communication skills. Students are required to achieve competency in a Modern Indian Language (MIL) and in the English language with special emphasis on language and communication skills. The courses aim at enabling the students to acquire and demonstrate the core linguistic skills, including critical reading and expository and academic writing skills, that help students articulate their arguments and present their thinking clearly and coherently and acquaint with the cultural and intellectual heritage of languages.

4.2.5. Skill Enhancement Courses (SEC): 9 credits

These courses are aimed at imparting practical skills, hands-on training, soft skills, and other skills to enhance the employability of students. Courses are designed as per the students' needs with the available resources. Students can choose these courses from the list of courses offered in the chosen specialization as said in 4.2.2.

4.2.6. Value-Added Courses (VAC) Common to All UG Students: 8 credits

4.2.6.a. Understanding India

This course aims at enabling the students to acquire and demonstrate the knowledge and understanding of contemporary India with its historical perspective, the basic framework of the goals and policies of national development, and the constitutional obligations with special emphasis on constitutional values and fundamental rights and duties. The course would also focus on developing an understanding among student-teachers of the Indian knowledge systems, the Indian education system, and the roles and obligations of teachers to the nation in general and to the school/community/society.

The course will attempt to deepen knowledge about and understanding of India's freedom struggle and of the values and ideals that it represented to develop an appreciation of the contributions made by people of all sections and regions of the country, and help learners understand and cherish the values enshrined in the Indian Constitution and to prepare them for their roles and responsibilities as effective citizens of a democratic society.

4.2.6.b. Environmental Science / Education, Higher Order Thinking

This course seeks to equip students with the ability to apply the acquired knowledge, skills, attitudes, and values required to take appropriate actions for mitigating the effects of environmental degradation, climate change, and pollution, effective waste management, conservation of biological diversity, management of biological resources, forest and wildlife conservation, and sustainable development and living. The course will also deepen the knowledge and understanding of India's environment in its totality, its interactive processes, and its effects on the future quality of people's lives.

4.2.6.c. Digital and Technological Solutions

Courses in cutting-edge areas that are fast gaining prominences, such as Artificial Intelligence (AI), 3-D machining, big data analysis, machine learning, drone technologies, and Deep learning with important applications to health, environment, and sustainable living that will be woven into undergraduate education for enhancing the employability of the youth.

4.2.6.d. Health & Wellness, Yoga Education, Sports, and Fitness, Universal Human Values

Course components relating to health and wellness seek to promote an optimal state of physical, emotional, intellectual, social, spiritual, and environmental well-being of a person. Sports and fitness

activities will be organized outside the regular institutional working hours. Yoga education would focus on preparing the students physically and mentally for the integration of their physical, mental, and spiritual faculties, and equipping them with basic knowledge about one's personality, maintaining self-discipline and self-control, to learn to handle oneself well in all life situations.

4.2.7. Vocational Training / Education: 12 Credits

These courses are meant to provide the students with adequate knowledge and skills for employment and entrepreneurship. The requirements of industries in the chosen specialization are incorporated in these courses to groom the students to take up gainful employment or becoming entrepreneurs.

Vocational education courses are designed to relate the skills provided with the content of general education in order to ready the students for work at each exit point of the programme. A minimum of 12 credits are allotted to the minor or specialization stream relating to vocational education and training (as given in 4.2.2).

4.2.8. Summer Internship: 4 Credits

All students will undergo internships / Apprenticeships in a firm, industry, or organization or Training in labs with faculty and researchers in their own or other HEIs/research institutions during the summer term. Students will be provided with opportunities for internships to actively engage with the practical side of their learning and, as a by-product, further improve their employability. Summer internship shall be conducted for a minimum of 8 weeks.

4.2.9. Community Engagement and Service: 2 Credits

The curricular component of 'Community Engagement and Service' seeks to expose students to the socio-economic issues in society so that the theoretical learnings can be supplemented by actual life experiences to generate solutions to real-life problems. This can be part of summer term activity or part of a major or minor course. Community Engagement shall be conducted for a minimum of 2 weeks.

4.2.10. Research Project / Dissertation: 12 Credits

Students choosing a 4 Year Bachelor's degree (Honors with Research) are required to take up research projects under the guidance of a faculty member. The students are expected to complete the Research Project in the eighth semester.

4.2.11. Audit courses: 0 credits

Audit courses offered do not carry any credits. Evaluation will be based on continuous assessment. Students may be given a pass or fail (P/F) based on the assessment that may consist of class tests, homework assignments, and/or any other innovative assessment methodology suitable to the expected learning outcome, as determined by the faculty in charge of the course of study.

4.3. Levels of the Courses

Course codes are based on the academic rigor. The first four letters of the course code indicate the department/Centre, followed by the academic rigor level code in digits (For e.g., Comp 201). The coding structure follows:

4.3.1. 0-99: Pre-requisite courses

It is required to undertake an introductory course which will be a pass or fail course with no credits. It will replace the existing informal way of offering bridge courses that are conducted in some of the colleges/ universities.

4.3.2. 100-199: Foundation or introductory courses

These are courses which are intended for students to gain an understanding and basic knowledge about the subjects and help decide the subject or discipline of interest. These courses generally would focus on foundational theories, concepts, perspectives, principles, methods, and procedures of critical thinking in order to provide a broad basis for taking up more advanced courses.

4.3.3. 200-299: Intermediate-level courses including subject-specific courses

These courses are intended to meet the credit requirements for minor or major areas of learning. These courses can be part of a major and can be pre-requisite courses for advanced-level major courses.

4.3.4. 300-399: Higher-level Courses

These courses are required for majoring in a disciplinary/interdisciplinary area of study for the award of a degree.

4.3.5. 400-499: Advanced Courses

These courses which would include lecture courses with practicum, seminar-based course, term papers, research methodology, advanced laboratory experiments/software training, research projects, hands-on-training, internship/apprenticeship projects at the undergraduate level or First year post-graduate theoretical and practical courses.

4.4. Credit-hours for different types of courses

A three-credit lecture course in a semester means three one-hour lectures per week with each one-hour lecture counted as one credit. One credit for tutorial work means one hour of engagement per week. A one-credit course in practicum or lab work, community engagement and services, and fieldwork in a semester mean two-hour engagement per week.

The Faculty to Student Ratio in all the practical / laboratory classes shall be maintained at 1:25.

In a semester of 15 weeks duration, a one-credit practicum in a course is equivalent to 30 hours of engagement. A one-credit of Seminar or Internship or Studio activities or Field practice/projects or Community engagement and service means two-hour engagements per week. Accordingly, in a semester of 15 weeks duration, one credit in these courses is equivalent to 30 hours of engagement.

4.4.1. Pedagogical Styles

In order to achieve the expected Learning outcomes, UGC Framework has specified different Pedagogical approaches for different courses at undergraduate level.

These approaches include:

a) Lecture course

b) Tutorial course

c) Practice cum or laboratory courses

d) Seminar Course

e) Internship course

f) Studio activity-based course

g) Field practicing

h) Project work courses

i) Community engagement and service course

The details of these different types of Pedagogical methods are as follows:

Table 2: Pedagogical Approaches

COURSE TYPES	APPROACH
Lecture Courses	Regular classroom lectures by qualified / experienced Expert Teachers • These Lectures may also include classroom discussion, demonstrations, case analysis • Use of Models, Audio-Visual contents, Documentaries, PPTs may supplement.
Tutorial Courses	Problem solving Exercise classes guided discussion, supplementary readings vocational training, etc.
Practical / Lab work	Practical Lab activity with Theoretical support Mini projects, Activity based engagement, Program executions, Data processing and presentation exercise.
Seminar Course	A course requiring student to design and participate in discussions, Group Discussions, Elocution and Debate, Oral Communication Paper presentations, Poster Presentation, Role play participation, Quiz competitions, Business plan preparation/presentation, etc.
Internship course	Courses requiring students to *Learn by Doing* in the workplace external to the educational Institutions. Internships involve working in Software Companies, Research and Higher Educational Institution Laboratories, Corporate Offices, etc. All Internships should be properly guided and inducted for focused learning.
Research Project	Students need to study and analyze the recent research publications from indexed/peer reviewed journals in their area of specialization. Outcome of the study and analysis need to be presented as a thesis or research report with necessary experimental results.

5. ADMISSIONS ELIGIBILITY, LATERAL ENTRY

5.1 Admissions Eligibility

The candidates for admission to this programme shall be required to have passed 10+2 / 10+3 system of examinations or equivalent with Mathematics / Business Mathematics / Computer Science / Computer Applications / Informatics Practices / or Equivalent as one of the subjects of study.

Students shall be admitted to this programme based on admissions criteria fixed by the University / Government of Puducherry from time to time.

5.2 Admissions by Lateral Entry

In this programme, where admission was carried out adopting approved procedures in preceding years, subject to availability, lateral entry admission shall be permitted, subject to:

Candidates seeking entry at the second, third and fourth year, should meet the necessary eligibility criteria with respect to the certificate / diploma / degree they possess, with necessary minimum credits banked in the Academic Bank of Credits (ABC). Such students who get admitted in later years, other than first year will be guided by the following clauses:

- that the University shall notify the admission process and number of vacancies open for lateral entry.
- that the Lateral entrants shall be admitted only after such transparent screening process and such procedure that the University may prescribe from time to time. University may prescribe different methods of screening for different programmes depending on the circumstances prevailing in each case.
- Lateral entry shall be permissible only in the beginning of years 2, 3, 4 of the Under Graduate / Honors programme; provided that the students seeking lateral entry shall have obtained the minimum pass marks / grades fixed by the University in their previous academic years.

6. DEGREES, DIPLOMAS AND CERTIFICATES OF THE UNIVERSITY AND MINIMUM CREDIT REQUIREMENTS

The University shall award Degrees, Diplomas and Certificates as follows:

a. A Bachelor's with Honors / Honors with Research Degree in Computer Science will be awarded to the candidate who has earned at least 164 credits in 4 years; and a basic Bachelor's Degree in Computer Science for those who have earned at least 124 credits including NEP specified courses, during first 3 years.

b. A Diploma in Computer Science for those students who have earned at least 84 credits including the NEP specified courses and the mandatory 4 credits of skill enhancement course and an internship for a minimum of 8 weeks in the summer semester.

c. A Certificate in Problem Solving and Programming for those students who have earned at least 42 credits including the NEP specific courses and the mandatory 4 credits of skill enhancement course and an internship for a minimum of 8 weeks in the summer semester.

7. ACADEMIC BANK OF CREDITS (ABC)

The scheme of academic bank of credits will facilitate the transfer and consolidation of credits by using an 'Academic Bank Account' opened by students across the country by taking up courses in any of the eligible Higher Education Institutions. The validity of the credits earned and kept in the academic credit account will be to a maximum period of seven years or as specified by the ABC time to time.

8. MINIMUM CREDITS FOR ENROLMENT, ONLINE COURSES, STUDENT STRENGTH AND MENTORSHIP

a) To be considered a full-time student, a student must be enrolled at least for 12 credits in each and every Semester. No student, unless specifically permitted by the Programme committee, be permitted to enrol in more than 30 credits in any semester (excluding the credits for writing arrear exams).

b) 40% of the credits of minor courses may be earned through online mode (Swayam or such other similar platforms) approved by the department and the University as per the existing UGC regulations. Such decisions may be taken by the department/centres after considering the course requirements and learning outcomes planned and duly approved by the Programme Committee. This does not apply to the major courses/internship/skill enhancement courses/community service/engagement or any other hands-on/vocational programmes. Such courses need to be completed offline/physical mode.

c) Course code for online courses and the number of credits assigned to each course will be approved by the programme committee, and these will be uploaded in the PU-SAMS portal.

d) A student will be permitted to register for only one minor course during one semester.

e) Students will be permitted to drop online courses within the time limits prescribed in the Academic Calendar.

f) Every student upon admission to the University shall be associated with a member of the faculty (faculty advisor) of the programme to which she/he is admitted to, who shall advise and help the student as a mentor in choosing courses that is most appropriate for the goals of the student.

g) No minor course shall be offered unless a minimum of 10 students are registered.

h) Programme Committee approved SOP may be followed for online course registration, submission of marks or grades, certificates etc.,.

9. EXIT OPTION PROCEDURES

a) Students enrolled in this programme shall have an option to exit at the end of 1st, 2nd and 3rd years of a programme, subject to fulfilment of conditions.

b) A student desiring an exit shall give a notice of such intention in writing in the prescribed format at least 8 weeks before the scheduled end of the Academic year.

c) On receipt of the notice, department shall recommend for a Certificate/ Diploma/ Degree as the case may be, based on the requirements for such degrees. In case of arrear papers, the certificate shall be provided after passing the arrear paper.

d) As soon as the student completes the requirements of the certificate/diploma/degree, as the case may be, the Department shall communicate to such officer as may be notified by the Administration.

10. LEARNING ASSESSMENT AND GRADES

All Credit courses are evaluated for 100 marks. The courses are in three categories and the assessment methods are defined for each category.

10.1. Category of Courses

Category A	Theory Courses with Lecture hours and hours allotted for Tutorials wherever required.
Category B	Practical Courses with only Practical hours or Laboratory hours. Laboratory Courses, Internships, Research Project Works and other courses allotted only with practical hours in the curriculum shall be under this category.
Category C	Theory & Practice combined Courses where Lecture and Practical hours allotted.

10.2. Learning Assessment

<table>
<tr><th>Course Types</th><th>Internal Assessment</th><th>End Semester Assessment</th></tr>
<tr><td>Category A

IA: 40 Marks
EA: 60 Marks</td><td>40 Marks

<table><tr><th>Evaluation Component</th><th>Marks</th></tr><tr><td>I. Internal Exams for 15 Marks (Two)</td><td>30</td></tr><tr><td>II. Assignments/Seminars/Others</td><td>10</td></tr><tr><td>Total</td><td>40</td></tr></table></td><td>60 Marks

(Evaluation Details given in Table 3)</td></tr>
<tr><td>Category B

IA: 50 Marks
EA: 50 Marks</td><td>50 Marks

For Practical / Internship Courses<table><tr><th>Evaluation Component</th><th>Marks</th></tr><tr><td>I. Weekly Observation Book / Report</td><td>15</td></tr><tr><td>II. Practical Record / Internship Report</td><td>15</td></tr><tr><td>III. Model Practical Exam</td><td>20</td></tr><tr><td>Total</td><td>50</td></tr></table>For Research Project Work Course<table><tr><th>Evaluation Component</th><th>Marks</th></tr><tr><td>I. Monthly Review (3 Reviews – 10 Marks each)</td><td>30</td></tr><tr><td>II. Project Report</td><td>10</td></tr><tr><td>III. Project Work</td><td>10</td></tr><tr><td>Total</td><td>50</td></tr></table></td><td>50 Marks

(Evaluation Details given in Table 3)</td></tr>
<tr><td>Category C

IA: 40 Marks
EA: 60 Marks</td><td>40 Marks

<table><tr><th>Evaluation Component</th><th>Marks</th></tr><tr><td>I. Internal Exams for 10 Marks (Two)</td><td>20</td></tr><tr><td>II. Observation Records</td><td>05</td></tr><tr><td>III. Model Practical Exam</td><td>15</td></tr><tr><td>Total</td><td>40</td></tr></table></td><td>60 Marks

(Evaluation Details given in Table 3)</td></tr>
</table>

10.3. End Semester Assessment

The Department schedules the End-Semester exams for all theory and practical subjects based on university calendar. For Theory courses with Practical components, End semester exams shall be conducted separately for Theory and Practical.

A detailed Exam Time Table shall be circulated at least 15 days before the start of exams. Question Papers shall be set based on BoS approved syllabus. All students who have a minimum of 70% attendance are eligible to attend the end-semester exams. Attendance percentage shall be calculated for each course to decide the eligibility of the candidate for writing the end-semester examination.

10.4. Breakup of End Semester Marks

The question paper shall be set as per the Bloom's Taxonomy. Table3 below gives the details of evaluation methods for Category A, B and C courses. Various levels along with it's description and sample questions are as follows:

Knowledge: Recall or remember previously learned information.

Example: List the basic data types in Python

Comprehension: Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating the main ideas.

Example: Explain how a stack data structure works.

Application: Apply knowledge and concepts to solve problems in new situations. Use learned information in a different context.

Example: Write a Python program to solve the deadlock problem.

Analysis: Break down information into parts and examine the relationships between the parts. Identify motives or causes.

Example: Analyse the efficiency of two sorting algorithms and compare their advantages and disadvantages.

Synthesis: Create a new whole by combining elements in novel ways. Use creativity to produce something original.

Example: Design a web application that can generate a time table of a school.

Table3: End Semester Assessment examination details for all three categories of courses

Course Components	Max. Marks	End-Sem Exam Duration
Category A. Theory subjects: Sec A: 10 Questions of 2 Marks each (20 Marks) *(Knowledge: 3, Comprehension: 2, Application: 3, Analysis:2)* Sec B: 5 out of 7 Questions of 4 Marks each (20 Marks) *(Knowledge: 1, Comprehension: 2, Application: 1, Analysis:3)* Sec C: 2 Either/OR choice questions of 10 Marks each (20 Marks) *(Application: 1, Analysis:1)* **Questions from all units of Syllabus equally distributed.**	60 Marks	3 Hours
Category B. Skill Enhancement/ Practical: Based on Practical examinations conducted. **Internship / Research Project Work:** Presentation of the work / Report / Viva-voce examinations	50 Marks	3 Hours -
Category C. Theory Subjects with Practical Components: **i. Theory Component:** Sec A: 10 Questions of 2 Marks each (20 Marks) *(Knowledge: 3, Comprehension: 2, Application: 3, Analysis:2)* Sec B: 5 out of 7 Questions of 4 Marks each (20 Marks) *(Comprehension: 2, Application: 3, Analysis:2)* Sec C: 2 Either or type questions of 10 Marks each (20 Marks) *(Analysis / Synthesis)* **Questions from all units of Syllabus equally distributed.** **The examination shall be conducted for 60 Marks and reduced to 40 Marks.** **ii. Practical Component:** Based on Practical examinations / Presentation / Viva-voce with external examiner appointed by the University Controller of Examinations, and schedules exclusively prepared for such practical examinations by the University Examination Section. **The examination shall be conducted for 40 Marks and reduced to 20 Marks.** **Total Marks: 60 (Theory: 40 Marks + Practical: 20 Marks)**	40 Marks 20 Marks	3 Hours 2 Hours

10.5. Consolidation of Marks and Passing Minimum

The Course Teacher of Course shall consolidate the Internal Assessment marks and marks secured by students in End-Semester examinations. The total marks will be converted into letter grades.

A student shall be declared to have passed the course only if she/he gets, a minimum of 40% marks in end semester exam and a minimum of 40% marks in aggregate (Internal Assessment + End Semester Assessment put together). However, Pondicherry University considers 40% marks as pass during first 3 years of study and students who secured less than 50 will be awarded 'P' (Pass Grade).

10.6. Supplementary Examination

- A student who gets F grade in a course shall be permitted to register for the supplementary examination in the following semester or in the subsequent semesters.
- A student who gets F grade in a course shall be given an option either to retain the previously awarded internal assessment mark or to improve it, and the higher mark out of these two options will be considered for the supplementary examination.
- A student who gets Ab grade in a course/practicum/vocational course/internship/practicum or any other hands-on skill related course is mandated to repeat the course and undergo all the stages of assessment in subsequent semesters.

10.7. Attendance Requirement

No student who has less than 70% attendance in any course shall be permitted to participate in end semester examination and she/he shall be given 'Ab' Grade - failure due to lack of attendance. she/he shall be required to repeat that course as and when it is offered.

11. LETTER GRADES, GRADE POINTS, GRADE CARD

Performance of students in each paper will be expressed as marks as well as Letter Grades.

Letter Grade	Grade Point
O (Outstanding)	10
A+ (Excellent)	9
A (Very good)	8
B+ (Good)	7
B (Above average)	6
C (Average)	5
P (Pass)	4
F (Fail)	0
Ab (due to lack of attendance)	0

In case of fractions the marks shall be rounded off to nearest integer.

The class interval K will be calculated by the formula given below:

K = (X-50)/6

where X is the highest mark secured.

According to K value, one of the following grading schemes will be followed.

(i) If K ≥ 5, then the grades shall be awarded as given in the following table.

Range of Marks in %	Letter Grade Points for	Grade Points for
X to (X-K) + 1	O	10
(X-K) to (X-2K) + 1	A+	9
(X-2K) to (X-3K) + 1	A	8
(X-3K) to (X-4K) + 1	B+	7
(X-4K) to (X-5K) + 1	B	6
(X-5K) to 50	C	5
40 – 49	P	4
Below 40	F	0
Absent (Lack of Attendance)	Ab	0

(ii) If K< 5, then the grades shall be awarded as given in the following table.

Range of Marks in %	Letter Grade Points for	Grade Points for
80-100	O	10
71-79	A+	9
66-70	A	8
61-65	B+	7
56-60	B	6
50-55	C	5
40-49	P	4
Below 40	F	0
Absent (lack of attendance)	Ab	0

The Semester Grade Point Average (SGPA) is computed from the grades as a measure of the student's performance in a given semester. The SGPA is based on the grades of the current term, while the Cumulative GPA (CGPA) is based on the grades in all courses taken after joining the programme of study.

11.1. Computation of SGPA and CGPA

The following procedure shall be followed to compute the Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA):

The SGPA is the ratio of the sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e. SGPA (Si) = $\Sigma(Ci \times Gi) / \Sigma Ci$

Where Ci is the number of credits of the i^{th} course and Gi is the grade point scored by the student in the i^{th} course.

11.1.1. Example for Computation of SGPA where candidate has not failed in any course.

Semester	Course	Credit	Letter Grade	Grade Point	Credit Point (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	B	6	3 X 6 = 18
I	Course 4	3	O	10	3 X 10 = 30
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	B	6	4 X 6 = 24
		20			139
	SGPA				139/20=6.95

11.1.2. Example for Computation of SGPA where candidate has failed in one course.

Semester	Course	Credit	Letter Grade	Grade Point	Credit Point (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	B	6	3 X 6 = 18
I	Course 4	3	O	10	3 X 10 = 30
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	F	0	4 X 0 = 00
		20			115
	SGPA				115/20=5.75

11.1.2. Example for Computation of SGPA where candidate has failed in two courses.

Semester	Course	Credit	Letter Grade	Grade point	Credit Point (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	F	0	3 X 0 = 00
I	Course 4	3	B	6	3 X 6 = 18
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	F	0	4 X 0 = 00
		20			85
	SGPA				85/20=4.25

The CGPA shall also be calculated in similar way as shown in examples (i), (ii) and (iii) of SGPA for all subjects taken by the students in all the semesters. However, if any student fails more than once in the same subject, then while calculating CGPA, the credit and grade point related to the subject in which the student fails in multiple attempts will be restricted to one time only. The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts.

In case of audit courses offered, the students may be given (P) or (F) grade without any credits. This may be indicated in the mark sheet. Audit courses will not be considered towards the calculation of CGPA.

11.2. Grade Card

The University shall issue a Grade card for the students, containing the marks and grades obtained by the student in the previous Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA).

The grade card shall list:

a. The title, semester and course code of the courses taken by the student.

b. The credits associated with the course.

c. The marks and grade secured by the student.

d. The total credits earned by the student in that semester

e. The SGPA of the student.

f. The total credits earned by the students till that semester.

g. The CPGA of the student.

12. DISTINCTION, RANK AND DIVISIONS

Distinction: On successful completion of the programme, students with a CGPA of 9.00 and above who passed all the courses in first attempt shall be awarded the degree in First Class with Distinction.

University Rank: University Rank in a programme will be awarded to the student who secures the highest CGPA in a batch and passed all the courses in first attempt.

Divisions: Students with CGPA between 6.00 and 8.99 shall be placed in First Class, students with CGPA between 5.00 and 5.99 shall be placed in Second Class, and students with CGPA between 4.00 and 4.99 shall be placed in Pass Class.

Range of CGPA	Result
9.0 – 10	First Class with distinction#
6.0 - 8.99	First Class
5.0 - 5.99	Second Class
4.0 - 4.99	Pass

Distinction will be awarded ONLY to those candidates who have cleared ALL subjects in the first attempt.

13. MINIMUM CREDIT REQUIREMENTS

S. No.	Component	3-year UG			4-year UG (Honors / Honors With research)		
		Credits	Courses	Cr/Course	Credits	Courses	Cr/Course
1	Major Disciplinary/ Interdisciplinary Courses	56	14	4	76	19	4
2	Minor Disciplinary/ Interdisciplinary Courses	24	6	4	32	8	4
3	Multi-Disciplinary Courses	9	3	3	9	3	3
4	Ability Enhancement Courses	12	4	3	12	4	3
5	Skill Enhancement Courses	9	3	3	9	3	3
6	Value-added courses	8	4	2	8	4	2
7	Summer Internship (MJD 11)	4	1	4	4	1	4
8	Community Engagement and Service	2	1	2	2	1	2
9	Research Project/Dissertation	--	--	--	12	Project or 3 Courses##	
Total		**124**			**164**		

##Note: *Honors students not undertaking research will do 3 courses for 12credits in lieu of a research project/Dissertation.*

- *MJD: Major Disciplinary (Compulsory – Hardcore Subjects)*
- *MID: Minor Disciplinary (Specialization Specific – Softcore Subjects)*
- *MLD: Multi-Disciplinary*
- *AEC: Ability Enhancement Courses*
- *SEC: Skill Enhancement Courses*
- *VAC: Value Added Courses*

14. COURSE CODE

- **Course code :** 7 Characters: 4 Alphabets and 3 Digits. Ex: ABCD123
- **Alphabets :** 1st and 2nd Alphabets: Major domain
  3rd and 4th Alphabets: Specialization
- **Digits :** 1st Digit: Levels (100, 200, 300, 400…)
  2nd and 3rd Digits: Serial number of the courses in the given year

Example: CSAI312: Computer Science Artificial Intelligence, Level - 300, Serial number of the course in the given year - (12)

B.Sc. COMPUTER SCIENCE CURRICULUM

FIRST SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 1	CSCS101	Digital Logic Fundamentals	H	4	3		2
2	MID 1	CSCS102	Microprocessor and Assembly Language Programming	S	4	3		2
3	MLD 1		One course from the MLD streams (Table 10)	H	3	4		
4	AEC 1		English I / Modern Indian Languages I	H	3	2		2
5	SEC 1	CSCS103 CSCS104	S.No. 1 or 2 from Table 7	S	3	2		2
6	VAC 1		Understanding India	H	2	4		0
7	VAC 2		Environmental Sciences / Education / Higher Order Thinking	H	2	4		0
Total					**21**	**30 Hours**		

SECOND SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 2	CSCS105	Problem Solving & Programming Fundamentals	H	4	3		2
2	MID 2	CSCS106	Microcontrollers Programming	S	4	3		2
3	MLD 2		One course from the MLD streams except the stream chosen in MLD1 (Table 10)	H	3	4		
4	AEC 2		English I / Modern Indian Languages I	H	3	2		2
5	SEC 2	CSCS107 CSCS108	S.No. 3 or 4 from Table 7	S	3	2		2
6	VAC 3		Health & Wellness /Yoga Education / Universal Human Values	H	2			4
7	VAC 4	CSVA101	Digital Technologies	H	2	3		
Total					**21**	**29 Hours**		

THIRD SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 3	CSCS201	Object Oriented Programming	H	4	3		2
2	MJD 4	CSCS202	Data Structures	H	4	3		2
3	MID 3	CSCS203	System Software	S	4	3		2
4	MLD 3		One course from the MLD streams except the streams chosen in MLD1 and MLD2 (Table 10)	H	3	4		
5	AEC 3		English II / Modern Indian Languages II	H	3	2		2
6	SEC 3	CSCS204 / CSCS205	S.No. 5 or 6 from Table 7	S	3	2		2
				Total	**21**	**27 Hours**		

FOURTH SEMESTER								
S.No.	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 5	CSCS206	Computer System Architecture	H	4	3		2
2	MJD 6	CSCS207	Design and Analysis of Algorithms	H	4	3		2
3	MJD 7	CSCS208	Database Management Systems	H	4	3		2
4	MID 4	CSCS209	Embedded Application Development	S	4	3		2
5	AEC 4		English II / Modern Indian Languages II	H	3	2		2
6	Project	CSCS210	Community Engagement and Service	H	2			6
				Total	**21**	**30 Hours**		

FIFTH SEMESTER								
S.No.	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 8	CSCS301	Operating Systems	H	4	3		2
2	MJD 9	CSCS302	Mathematical Foundations of Computer Science	H	4	3	1	
3	MJD 10	CSCS303	Computer Networks	H	4	3		2
4	MID 5	CSCS304	Theory of Computation	S	4	3	1	
5	MJD 11	CSCS305	Summer Internship	H	4			6
				Total	**20**	**24 Hours**		

SIXTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 12	CSCS306	Management Strategies & Concepts	H	4	4		
2	MJD 13	CSCS307	Software Engineering Theory and Practice	H	4	3		2
3	MJD 14	CSCS308	Distributed Systems	H	4	3		2
4	MJD 15	CSCS309	Operations Research	H	4	3	1	
5	MID 6	CSCS310 / CSCS311	Any one course from Table 1	S	4	3		2
				Total	**20**	**23 Hours**		

SEVENTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 16	CSCS401	Web Engineering	H	4	3		2
2	MJD 17	CSCS402	System Modelling & Simulation	H	4	3		2
3	MJD 18	CSCS403	Wireless Communication Networks	H	4	3		2
4	MID 7	CSCS404 / CSCS405	Any one course from Table 2	S	4	3		2
5	MID 8	CSCS406 / CSCS407	Any one course from Table 3	S	4	3		2
				Total	**20**	**25 Hours**		

EIGHTH SEMESTER – B.Sc. Computer Science (Honors)								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 19	CSCS408 / CSCS409	Any one course from Table 4	S	4	3		2
2	MJD 20	CSCS410 / CSCS411	Any one course from Table 5	S	4	3		2
3	MJD 21	CSCS412	High Performance Computing	H	4	3		2
4	MJD 22	CSCS413	Cloud Computing	H	4	3		2
5	MJD 23	CSCS414	Deep Learning	H	4	3		2
				Total	**20**	**25 Hours**		

EIGHTH SEMESTER – B.Sc. Computer Science (Honors with Research)								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 19	CSCS408 / CSCS409	Any one course from Table 4	S	4	3		2
2	MJD 20	CSCS410 / CSCS411	Any one course from Table 5	S	4	3		2
3	MJD 21	CSCS415	Research Project	H	4			5
4	MJD 22	CSCS416	Project Report	H	4			5
5	MJD 23	CSCS417	Project Viva-voce	H	4			5
			Total		**20**	**25 Hours**		

Table 1: MID 6 – SIXTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MID 6	CSCS310	Unix System Programming	S	4	3		2
2	MID 6	CSCS311	Network Programming	S	4	3		2

Table 2: MID 7 – SEVENTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MID 7	CSCS404	Artificial Intelligence	S	4	3		2
2	MID 7	CSCS405	Compiler Design	S	4	3		2

Table 3: MID 8 – SEVENTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MID 8	CSCS406	Cyber Security	S	4	3		2
2	MID 8	CSCS407	Internet of Things	S	4	3		2

Table 4: MJD 19 – EIGHTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 19	CSCS408	Machine Learning	S	4	3		2
2	MJD 19	CSCS409	Full Stack Development	S	4	3		2

Table 5: MJD 20 – EIGHTH SEMESTER								
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**	**Credits**	**Hours/Week**		
						L	**T**	**P**
1	MJD 20	CSCS410	5G Communication Technologies	S	4	3		2
2	MJD 20	CSCS411	Data Mining	S	4	3		2

Table 6: MJD 21 / MJD 22 / MJD 23 – EIGHTH SEMESTER								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	MJD 21	CSCS412	High Performance Computing	H	4	3		2
2	MJD 22	CSCS413	Cloud Computing	H	4	3		2
3	MJD 23	CSCS414	Deep Learning	H	4	3		2

Table 7: SEC 1 / SEC 2 / SEC 3 – I / II / III SEMESTERs								
S.No	Component	Course Code	Title of the Course	H/S	Credits	Hours/Week		
						L	T	P
1	SEC 1	CSCS103	Python Programming	S	3	3		2
2	SEC 1	CSCS104	R Programming	S	3	3		2
3	SEC 2	CSCS107	Programming for Mobile Devices	S	3	3		2
4	SEC 2	CSCS108	Visual Programming with C#	S	3	3		2
5	SEC 3	CSCS204	3D Modelling & Animation	S	3	3		2
6	SEC 3	CSCS205	Game Programming	S	3	3		2

Table 8: List of Major Disciplinary Courses				
S.No	Component	Course Code	Title of the Course	H/S
1.	**MJD 1**	CSCS101	Digital Logic Fundamentals	**H**
2.	**MJD 2**	CSCS105	Problem Solving & Programming Fundamentals	**H**
3.	**MJD 3**	CSCS201	Object Oriented Programming	**H**
4.	**MJD 4**	CSCS202	Data Structures	**H**
5.	**MJD 5**	CSCS206	Computer System Architecture	**H**
6.	**MJD 6**	CSCS207	Design and Analysis of Algorithms	**H**
7.	**MJD 7**	CSCS208	Database Management Systems	**H**
8.	**MJD 8**	CSCS301	Operating Systems	**H**
9.	**MJD 9**	CSCS302	Mathematical Foundations of Computer Science	**H**
10.	**MJD 10**	CSCS303	Computer Networks	**H**
11.	**MJD 11**	CSCS305	Summer Internship	**H**
12.	**MJD 12**	CSCS306	Management Strategies and Concepts	**H**
13.	**MJD 13**	CSCS307	Software Engineering Theory and Practice	**H**
14.	**MJD 14**	CSCS308	Distributed Systems	**H**
15.	**MJD 15**	CSCS309	Operations Research	**H**
16.	**MJD 16**	CSCS401	Web Engineering	**H**
17.	**MJD 17**	CSCS402	System Modeling and Simulation	**H**
18.	**MJD 18**	CSCS403	Wireless Communication Networks	**H**
19.	**MJD 19**	CSCS408 / CSCS409	Machine Learning / Full Stack Development	**S**
20.	**MJD 20**	CSCS410 / CSCS411	5G Communication Technologies / Data Mining	**S**

Table 9: List of Minor Disciplinary Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**MID 1**	CSCS102	Microprocessor and Assembly Language Programming	**S**
2.	**MID 2**	CSCS106	Microcontrollers Programming	**S**
3.	**MID 3**	CSCS203	System Software	**S**
4.	**MID 4**	CSCS209	Embedded Application Development	**S**
5.	**MID 5**	CSCS304	Theory of Computation	**S**
6.	**MID 6**	CSCS310/ CSCS311	UNIX System Programming / Network Programming	**S**
7.	**MID 7**	CSCS404/ CSCS405	Artificial Intelligence / Compiler Design	**S**
8.	**MID 8**	CSCS406/ CSCS407	Cyber Security / Internet of Things	**S**

*Table 10: MLD 1 / MLD 2 / MLD 3 in Sem 1 / Sem 2 / Sem 3			
Streams	**Course Code**	**Title of the Course**	**H/S**
Natural Science		Biology	**H**
		Botany	**H**
		Zoology	**H**
		Biotechnology	**H**
		Biochemistry	**H**
Physical Sciences		Chemistry	**H**
		Physics	**H**
		Biophysics	**H**
		Astronomy	**H**
		Astrophysics	**H**
		Earth and Environmental Sciences	**H**
Mathematics & Statistics		STATA	**H**
		SPSS	**H**
		Tally	**H**
Computer Science	**COMS101**	Introduction to Python Programming	**H**
	COMS102	Foundations of Information Technology	**H**
Social Sciences		Political Sciences	**H**
		History	**H**
		Social work	**H**
		Sociology	**H**
Humanities		Anthropology	**H**
		Psychology	**H**
		Economics	**H**
Commerce & Management		Business Management	**H**
		Accountancy	**H**
		Finance	**H**
		Financial Institutions	**H**
Media Sciences		Journalism	**H**
		Mass Media	**H**
		Communication	**H**

*Courses will be announced after the approval of the respective boards.

Table 11: List of Ability Enhancement Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**AEC 1**		English I / Modern Indian Languages I	**H**
2.	**AEC 2**		English I / Modern Indian Languages I	**H**
3.	**AEC 3**		English II / Modern Indian Languages II	**H**
3.	**AEC 4**		English II / Modern Indian Languages II	**H**

Table 12: List of Skill Enhancement Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**SEC 1**	CSCS103	Python Programming	**S**
2.	**SEC 1**	CSCS104	R Programming	**S**
3.	**SEC 2**	CSCS107	Programming for Mobile Devices	**S**
4.	**SEC 2**	CSCS108	Visual Programming with C#	**S**
5.	**SEC 3**	CSCS204	3D modeling and Animation	**S**
6.	**SEC 3**	CSCS205	Game Programming	**S**

Table 13: List of Value-Added Courses				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**VAC 1**		Understanding India	**H**
2.	**VAC 2**		Environmental Education	**H**
3.	**VAC 3**		Health & Wellness / Yoga Education	**H**
4.	**VAC 4**		Digital Technologies	**H**

Table 14: Project (WP / Internship)				
S.No	**Component**	**Course Code**	**Title of the Course**	**H/S**
1.	**Project**	CSCS210	Community Engagement and Service	**H**

B.Sc. Computer Science

SYLLABUS

SEMESTER I

Year	I	Course Code: CSCS101 Course Title: Digital Logic Fundamentals	Credits	4
Sem.	I		Hours	75
			Category	C
Course Prerequisites, if any	NIL			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	● Understand the principles of digital systems and binary number operations ● Apply Karnaugh mapping to simplify Boolean expressions and optimize digital circuits ● Analyze and design basic combinational circuits. ● Synthesize and evaluate synchronous sequential circuits using storage elements and HDL ● Design and implement various types of registers and counters using HDL			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** Digital Systems – Binary Numbers – Conversions – Types – Codes – Storage and Registers – Binary Logic – Boolean Algebra – Theorems and Properties – Functions – Canonical and Standard Forms – Other Logic Operations – Digital Logic Gates – Integrated Circuits		9	
Unit II	**Gate-Level Minimization** Map Method – Four-Variable K-Map – Product-of-Sums Simplification – Don't-Care Conditions – NAND and NOR Implementation – Other Two-Level Implementations – Exclusive-OR Function – Hardware Description Language		9	
Unit III	**Combinational Logic** Analysis Procedure – Design Procedure – Binary Adder– Subtractor – Decimal Adder – Binary Multiplier – Magnitude Comparator – Decoders – Encoders – Multiplexers – HDL Models of Combinational Circuits		9	
Unit IV	**Synchronous Sequential Logic** Storage Elements – Latches – Flip-Flops – Analysis of Clocked Sequential Circuits – Synthesizable HDL Models of Sequential Circuits – State Reduction and Assignment – Design Procedure		9	
Unit V	**Registers and Counters** Registers – Shift Registers – Ripple Counters – Synchronous Counters – Other Counters – HDL for Registers and Counters		9	
Practical Component				
Exercises	1. Binary to Decimal and vice-versa 2. Decimal to Hexadecimal and Vice-Versa 3. Digital Logic Gates 4. Simplification of Boolean Functions 5. Combinational Logic Circuits i. Code Converters		30	

	ii. Arithmetic (Adders, Subtractors, Multipliers, Comparators) iii. Data Handling (Multiplexers, Demultiplexers, Encoders & Decoders) 6. Combinational Logic Circuit Design 7. Binary Adder-Subtractor Simulation 8. Decimal Adder Simulation 9. Binary Multiplier Simulation 10. Sequential Circuit Storage Elements: Flip-Flop Simulation	
Recommended Learning Resources		
Print Resources	1. M. Morris Mano, Michael D. Ciletti, "Digital design With an Introduction to the Verilog HDL", Pearson, Sixth Edition, 2018. 2. M. Rafiquzzaman, "Fundamentals of Digital Logic and Microcomputer Design", John Wiley & Sons, Inc., Fifth Edition, 2009.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	I	Course Code: CSCS102	Credits	4
Sem.	I	Course Title: Microprocessor & Assembly Language Programming	Hours	75
			Category	C
Course Prerequisites, if any	• Number Systems (binary, octal, hexadecimal) and their conversions • Boolean Algebra, logic gates, flip-flops and registers • Concepts in Combinational and Sequential logic			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Learn the architecture & organization of 8085 Microprocessor • Understand and classify the instruction set of the 8085Microprocessor • Apply the memory & I/O Interfacing with 8085 Microprocessor • Analyze the architecture and operation of Programmable Interface • Create applications to interface various peripheral IC's with Intel 8085 microprocessor			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction to Microprocessors & 8085 Assembly Language Programming** Microprocessors – Instruction set and computer languages – 8085 programming model – Instruction classification – Instruction – Data format and storage – Execute a simple program – 8085 Instruction Set	9
Unit II	**8085 Microprocessor architecture** Microprocessor Architecture and its operations – Memory – I/O Devices, 8085 MPU – 8085 based microcomputer – memory interfacing – 8155 memory segment Interfacing – Interfacing I/O devices: Basics – Interfacing input and output devices – memory mapped I/O	9
Unit III	**Programming 8085** Instruction Set of 8085 – Data Transfer – arithmetic – Logic – Branch – Writing ALP and Debugging programs – Looping – Counting and Indexing – 16-bit Arithmetic instructions – Logic operations – Counters and Time Delay	9
Unit IV	**Interfacing I/O Devices** Stack and subroutines – Restart – Conditional call and Return instruction – Advanced subroutine concepts – Code conversion – BCD Arithmetic and 16-bit operations – BCD-Binary conversion – Binary to BCD conversion – BCD to seven segment LED code conversion – Binary to ASCII and ASCII to binary conversion – BCD addition and subtraction	9
Unit V	**Interfacing Peripheral (I/O) and Applications** Interrupts: 8085 Interrupt – RST instructions – Software and Hardware interrupt – multiple Interrupts and Priorities – 8085 Vectored Interrupts – Restart as Software Instructions – 8155 – Multipurpose programmable Device – 8279 Programmable Keyboard/Display Interface – 8255 Programmable peripheral Interface	9

Practical Component		
Exercises	1. Assembly Language Programming for Arithmetic Operations like Addition, Subtraction, Multiplication and Division on 8, 16-bit data 2. Assembly Language Programming for different logical operations 3. Assembly Language Programming for code conversions 4. Assembly Language Programming for sorting 5. Assembly Language Programming for Searching 6. Assembly Language Programming for memory block transfer 7. Assembly Language Programming using subroutines 8. Assembly Language Programming using counters and time delay	30
Recommended Learning Resources		
Print Resources	1. Ramesh S. Gaonkar, "Microprocessor – Architecture, Programming and Applications with the 8085", Penram International Publisher, Sixth Edition, 2013. 2. Douglas V. Hall, "Microprocessors and Interfacing", Tata McGraw Hill publications, Third Edition, 2017.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	I	Course Code: CSCS103	Credits	3
Sem.	I	Course Title: Python Programming	Hours	60
			Category	B

Course Prerequisites, if any	Basic Knowledge in Programming Concepts	
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Practical): 03 hrs.
Course Outcomes	• Understand the basics of writing Python code • Implement programs using lists, tuples and dictionaries • Understand the use of control structures • Ability to write programs using packages • Understand the file manipulation	

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction, Data types** Introduction to Python – Advantages of using Python – Executing Python Programs – Python's Core data types – Numeric Types – String Fundamentals	6
Unit II	**Lists, Tuples, Dictionaries** Lists: list operations, list slices – list methods – list loop – mutability – aliasing – cloning lists – list parameters; Tuples: tuple assignment – tuple as return value; Dictionaries: operations and methods; advanced list processing – list comprehension	6
Unit III	**Control Flow, Functions, Modules** Python Statements: Assignments – Expressions – If condition – While and For Loops. Functions: Definition, Calls – Scopes – Arguments – Recursive Functions– Functional Programming tools Classes and Object-Oriented programming with Python – modules and Packages: Purpose, using packages – Exception Handling with Python	6
Unit IV	**Packages** Packages: NumPy, Pandas, Scikit learn – Machine learning with Python – Cleaning up, Wrangling, Analysis, Visualization - Matplotlib package – Plotting Graphs	6
Unit V	**File Handling** Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions	6
	Practical Component	
Exercises	1. Exchange the values of two variables 2. Finding minimum among n variables 3. Perform Simple sorting 4. Generate Students marks statement 5. Find square root, GCD, exponentiation 6. Sum the array of numbers 7. Perform linear search, binary search 8. Perform Matrix operations using NumPy 9. Perform Data frame operations using Pandas	30

	10. Use Matplotlib on dataset and visualise 11. Perform Word count, copy file operations	
Recommended Learning Resources		
Print Resources	1. Mark Lutz, “Learning Python”, Fifth Edition, O’Reilly, 2013. 2. Daniel Liang, “Introduction to programming using Python”, Pearson, First Edition, 2021. 3. Wes Mc Kinney, “Python for Data Analysis”, O’Reilly Media, 2012. 4. Tim Hall and J-P Stacey, “Python 3 for Absolute Beginners”, Apress, First Edition, 2009. 5. Magnus Lie Hetland, “Beginning Python: From Novice to Professional”, Apress, Second Edition, 2005.	
Syllabus Design: Dr. V. Uma, Associate Professor, PUDoCS		

Year	I	Course Code: CSCS104 Course Title: R Programming	Credits	3
Sem.	I		Hours	60
			Category	B
Course Prerequisites, if any	NIL			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Learn the basics in R programming • Understand to accessing variables and managing subsets of data • Design simple applications using the functions of R programming • Analyze the performance of the plotting tools in R programming • Create a project using the Lattice Package in R programming			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Downloading and Installing R – Script code – Graphing Facilities in R – Packages – General Issues in R – Getting Data into R – Importing Data	6
Unit II	**Accessing Variables and Managing Subsets of Data** Accessing variables from a Data Frame – Accessing Subsets of Data – Combining Two Datasets with a Common Identifier – Exporting Data – Recoding Categorical Variables	6
Unit III	**Simple Functions** The *tapply* Function – The *sapp*ly and *lapply* Functions – The *summary* Function – The *table* Function	6
Unit IV	**Plotting Tools** The *plot* Function – Symbols, Colours, and Sizes – Adding a Smoothing Line – Loops and Functions **Graphing Tools** Pie Chart – Bar Chart and Strip Chart – Boxplot – Cleveland Dotplots – Pairplot – Coplot – Combining Types of Plots	6
Unit V	**Lattice Package** High-level Lattice Functions – Multipanel Scatterplots – Multipanel Boxplots – Multipanel Cleveland Dotplots – Multipanel Histograms – Panel Functions – 3-D Scatterplots and Surface and Contour Plots	6
	Practical Component	
Exercises	1. Install R and RStudio, create and execute your first R script, generate basic plots using both base R and ggplot2, install and utilize R packages, and import and explore a dataset 2. Access specific variables from a data frame and manage subsets of data 3. Combine two datasets with a common identifier and export your final data set 4. Read data, explore structure using head(), summary(), and str() 5. Handle missing values, remove duplicates with duplicated() 6. Create plots (scatter, line, bar) using ggplot2 7. Create plots (pie, bar and strip chart, boxplot, Cleveland dotplots, pairplot, coplot) and a composite plot that combines multiple plot types, using R's graphing capabilities 8. Create advanced visualizations (multipanel scatterplots, boxplots, Cleveland dotplots, histograms, panel functions) using lattice functions in R	30

	9. Create 3-D scatterplots and surface and contour plots to explore complex data relationships	
Recommended Learning Resources		
References	1. Alain F. Zuur, "A Beginner's Guide to R", Springer-Verlag New York Inc., 2019. 2. Robert Knell, "Introductory R: A Beginner's Guide to Data Visualisation, Statistical Analysis and Programming in R", Amazon Digital South Asia Services Inc, Revised Edition, 2014.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

SEMESTER II

Year	I	Course Code: CSCS105	Credits	4
Sem.	II	Course Title: Problem Solving & Programming Fundamentals	Hours	75
			Category	C
Course Prerequisites, if any	NIL			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	● Analyze problems and develop top-down designs ● Write, compile, and debug basic programs ● Implement logic with conditionals and loops ● Manipulate arrays of various dimensions ● Design and implement functions with recursion			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction to Computer Problem-Solving** Problem-solving Aspect – Top-down Design – Implementation of Algorithms – Program Verification – Efficiency of Algorithms – Analysis of Algorithms	9
Unit II	**Basic programming constructs** Basic Data types (Numerical, String) – Variables – Expressions – I/O statements – Compile and Run – Debugging	9
Unit III	**Decision Making – Branching & Looping** Decision making – Relational Operators – Conditional statement, Looping Statements – Nested loops – Infinite loops – Switch Statements	9
Unit IV	**Array Techniques** Array Manipulation – Different operations – One dimensional Array – Two-dimensional Array – Multi-dimensional Array – Character – Arrays and Strings	9
Unit V	**Modular solutions** Introduction to Functions – Importance of Design of Functions – Arguments – Parameters – Return Values – Local and Global Scope – Recursion	9
	Practical Component	
Exercises	1. Program to array counting, array order reversal & find the maximum number in a set 2. Program for removal of duplicates from an ordered array & to partition an array 3. Program to find the k^{th} smallest element 4. Program to exchange the values of two variables without using a third variable 5. Program that takes a list of numbers as input and counts the total number of elements in the list 6. Program to compute the factorial of a given integer 7. Program to compute the sine of an angle (in degrees) using a series expansion 8. Program to generate the Fibonacci sequence up to a specified limit 9. Program that takes an integer as input and reverses	30

	its digits 10. Program that converts a number from one base to another (e.g., binary to decimal, decimal to binary)	
Recommended Learning Resources		
Print Resources	1. R. G. Dromey, "How to Solve it by Computer", Pearson Education India, Thirteen Edition, 2013. 2. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", Third Edition, O'Reilly Publishers, 2020.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	I	Course Code: CSCS106 Course Title: Microcontrollers Programming	Credits	4
Sem.	II		Hours	75
			Category	C
Course Prerequisites, if any	• Digital Logic Fundamentals • Microprocessors • Assembly Language Programming			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Learn the fundamentals of Microcontrollers • Understand the internal design of 8051 microcontroller along with the features and their programming • Analyze the on-chip peripherals of microcontrollers • Design different interfacing applications using microcontrollers and peripherals • Build systems using microcontrollers for real time applications			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Microprocessors and Microcontrollers** Microprocessors vs Microcontrollers – 8051 Architecture – Input/Output Pins – Ports – External Memory – Counter and Timers – Serial Data I/O – Interrupts		9	
Unit II	**Programming 8051** Addressing Modes – External Data Moves – Code Memory Read-Only Data Moves – PUSH and POP Opcodes – Data Exchanges – Logical Operations – Arithmetic Operations – Jump and Call Opcodes		9	
Unit III	**8051 Microcontroller Design** Microcontroller Specification – Design – Testing – Timing Subroutines – Lookup Tables for 8051 – Serial Data Transmission		9	
Unit IV	**Applications** Keyboards – Displays – Pulse Measurement – D/A and A/D Conversions – Multiple Interrupts		9	
Unit V	**Serial Data Communication** Network Configurations – 8051 Data Communication Modes		9	
Practical Component				
Exercises	1. Blinking LED 2. Digital Counter with Seven-Segment Display 3. Analog-to-Digital Conversion (ADC) 4. UART Communication 5. Timer Interrupt - Using a timer interrupt to perform a task at regular intervals 6. External Interrupt 7. Temperature Sensor (DS18B20) Interface 8. Matrix Keypad Interface 9. LCD Display Interface 10. Traffic Light Controller		30	

Recommended Learning Resources	
Print Resources	1. Kenneth J. Ayala, "The 8051 Microcontroller Architecture, Programming, and Applications", Delmar Cengage Learning, Third Edition, 2004. 2. Martin Bates, "PIC Microcontrollers - An Introduction to Microelectronics", Third Edition, Newnes, Elsevier, 2011. 3. Hubert Henry Ward, "C Programming for the PIC Microcontroller-Demystify Coding with Embedded Programming", Apress, UK, 2020. https://doi.org/10.1007/978-1-4842-5525-4
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS	

Year	I	Course Code: CSCS107 Course Title: Programming for Mobile Devices	Credits	3
Sem.	II		Hours	60
			Category	B
Course Prerequisites, if any	Basic computer programming skill			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the basics of Android Ecosystem • Learn to use the Android Ecosystem • Understand the programming constructs in Kotlin • Understand the process of building interactive apps, Games, Social Media apps • Understand the process of building apps for TVs, Wearable and Android Auto			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** About the Android Ecosystem – Installing Software tools – Creating an Android App – Examining a basic Android app – Improving the App.		6	
Unit II	**Android Background Material** Using Android Studio – Kotlin for Java programmers – Kotlin for Everyone – Object Orientation in Kotlin – Functional Programming in Kotlin – An Introduction to XML.		6	
Unit III	**The Building Blocks** Overview of Jetpack – Building foundations for the App – Architecture of the App – Defining App's behaviour – Interactivity.		6	
Unit IV	**Adding Cool Features** Building a Game in Android – Case study of building a Social Media App – Building Native applications		6	
Unit V	**Apps for Tablets, Watches, TVs and Cars** Apps for Tablets – Developing for Android Wear – Developing Android TV apps – Case study of App building with Android Auto		6	
Practical Component				
Exercises	1. Configure Android Studio and set the development environment 2. Build a basic Android app for numerical calculations 3. Build an Android app to use various sensors of the device 4. Case study: Build a calendar for Tablets 5. Case Study: Build a diet planner app		30	
Recommended Learning Resources				
Print Resources	1. Barry Burd, John Paul Mueller, "Android Application Development: All–in–one for Dummies", Third Edition, Wiley India, 2021. 2. Dawn Griffiths, David Griffiths, "Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin", Third Edition, O'Reilly, 2021.			
Syllabus Design: Dr. K. S. Kuppusamy, Associate Professor, PUDoCS				

Year	I	Course Code: CSCS108 Course Title: Visual Programming with C#	Credits	3
Sem.	II		Hours	60
			Category	B
Course Prerequisites, if any	Basic knowledge of computer Programming.			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Practical): 03 hrs		
Course Outcomes	• Understand the key components of the .NET Framework related to C# development • Learn the basic syntax and structure of C# programs • Design C# applications by integrating various object-oriented programming techniques in the .NET framework • Analyze the significance of graphical user interface (GUI) components and the Event Handling Model using C# programming • Learn and apply the fundamental skills to efficiently develop, test, and deploy ASP.Net Core applications			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction to .Net Framework** An Overview - Framework Components - The Common Language Runtime (CLR) - .NET Base Class Library - Common Language Specification (CLS) - Common Type System (CTS) - Metadata and Assemblies - .NET Namespaces - MSIL - JIT Compilers	6
Unit II	**Overview of C#** Program structure- Literals- Variables- Constants -Data Types- Operators-Statements and Expressions- Branching- Looping and loop control statements- Arrays- Strings manipulation- Boxing and Unboxing- Pre-processors- Namespaces	6
Unit III	**Object Oriented Programming concepts in C#** Class- Objects- Encapsulation- Constructors and its types- Inheritance-Polymorphism-Interface-Abstract class- Operator overloading- Properties- Indexers- Delegates- Collections	6
Unit IV	**Windows Forms** Introduction to Windows Forms and various controls-SDI and MDI applications- Menu Creation, Common Dialog Boxes- Events and event handling	6
Unit V	**Getting started with ASP.Net** Choosing a code editor, Creating an ASP.NET Core project, Running the ASP.NET Core application, ASP.NET Core application - Creating the project, Testing ASP.Net Core Applications - Creating a unit test project, Writing and running unit tests	6
	Practical Component	
Exercises	1. Installation of Visual Studio and creation of Simple Console Application 2. Create a simple C# program for the following concepts: a. To Check whether a given number is an Armstrong or not b. To Check whether the alphabet is a vowel or not using switch..case c. To Check whether the given string is palindrome or not using arrays	30

	3. Create a program to demonstrate boxing and unboxing operations 4. Implement the basic OOP concepts 5. Implement Interfaces and Operator Overloading 6. Create a GUI using standard controls, SDI & MDI forms 7. Design an application with menu options and a Common Dialog box 8. create a simple web application using ASP.Net 9. Develop any ONE case study listed below: a. Inventory Control b. Retail Shop Management c. Employee Information System d. Personal Assistant Program e. Students' Information System	
Recommended Learning Resources		
Print Resources	1. Herbert Schildt, "C# 4.0: The Complete Reference", First Edition**,** McGraw Hill Education, 2017. 2. Albahari. J, "C# 10 in a Nutshell: The Definitive Reference", First Edition, O'Reilly, 2022. 3. Adam Freeman. A, " Pro ASP.NET Core 7", Tenth Edition. Manning Publication, 2023.	
Syllabus Design: Prof. S. Ravi and Dr. S. L. Jayalakshmi, Assistant Professor, PUDoCS		

Year	I	Course Code: CSVA101 Course Title: Digital Technologies	Credits	2
Sem.	II		Hours	45
			Category	A
Course Prerequisites, if any	NIL			
Internal Assessment Marks: 40	End Semester Marks: 60		Duration of ESA (Theory) : 03 hrs.	
Course Outcomes	• Get introduced to the digital systems and its building blocks • Understand how the Digital Communication happens and to Learn the advantages and disadvantages including Cybersecurity • Learn the day-to-day digital activities and the initiatives on Digital India • Acquire knowledge on current Technologies and Trends in Digital Space • Explore the applications on the state of the art in Digital Technologies			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Digital Systems – Information & Communication Technology – ICT Tools. Computer Architecture – Software – Hardware – Operating System – Algorithms – Flowcharts	7
Unit II	**Communication Systems** Transmission Media – Computer Networks – Internet – Web Browsers – Search Engines – Messaging – Email – Social Media – Online Ethics **Cybersecurity** Threats – Significance – Challenges – Precautions – Safety Measures – Cyber Crime Awareness	7
Unit III	**Digital India & e-Governance** Initiatives - Unified Payment Interface – Aadhar online services – Credit / Debit Cards – e-Wallets – Mobile and Internet Banking – NEFT / RTGS / IMPS – Online Payments & PoS – Digital Accessibility	7
Unit IV	**Emerging Technologies & Applications** (Basic introduction only**)** Overview of Artificial Intelligence – Cloud Computing – Big Data – Internet of Things – Virtual Reality – 5G – 3D Printing	7
Unit V	**Case Studies** Any one case study on the emerging technologies and report submission by the candidates	7
	Practical Component	
Exercises	1. Operating System Installation and configuration 2. Application Software Installation and configuration 3. Hardware understanding and minor troubleshooting 4. Networking, cabling, configuration	10
	Recommended Learning Resources	
Print Resources	1. Pramod Kumar, Anuradha Tomar, R. Sharmila, "Emerging Technologies in Computing - Theory, Practice, and Advances", Chapman and Hall / CRC, First Edition, 2021, https://doi.org/10.1201/9781003121466. 2. V. Rajaraman, "Introduction to Information Technology", PHI, Third Edition, 2018. 3. E. Balagurusamy, "Fundamentals of Computers", Tata Mc GrawHill, Second Edition, 2011.	

	4. Behrouz A. Forouzan, “Data Communications and Networking”, McGraw Hill, Fourth Edition, 2007. 5. Rajkumar Buvya, James Broberg, and Andrzej Gosciniski, “Cloud Computing-Principals and Paradigms”, Wiley, 2011. 6. Stuart Russel and Peter Norvig, “Artificial Intelligence - A Modern Approach”, Pearson Education, Third Edition, 2010. 7. Samuel Greengard, “Internet of Things”, The MIT Press, 2015, https://doi.org/10.7551/mitpress/10277.001.0001. 8. C.S.V. Murthy, “E- Commerce – Concept, Models &Strategies”, Himalaya Publishing House, 2015. 9. Hurwith, Nugent Halper, Kaufman, “Big Data for Dummies”, Wiley & Sons, First Edition, 2013.
Syllabus Design: Dr. S. K. V. Jayakumar, Professor, PUDoCS	

SEMESTER III

Year	II	Course Code: CSCS201 Course Title: Object Oriented Programming	Credits	4
Sem.	III		Hours	75
			Category	C
Course Prerequisites, if any	Basic Programming knowledge			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the principles of OOP and the concept of class and objects • Apply the concept of Object initialization and overloading • Understand the concept of inheritance and reusability • Understand file operations and exception handling • Apply OOP to design and implement solutions to real-world problems			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Principles of Object-Oriented Programming (OOP)** Object Oriented Programming Paradigm-Basic Concepts of OOP-Benefits of OOP - Application of OOP - Simple C++ program - Compiling and Linking	9
Unit II	**Classes and Objects** Specifying class - Member functions - Nesting of Member functions - Access specifier - Static Data members and functions - Arrays within a Class - Arrays of Objects - Objects as Arguments - Returning Objects - Friend Function	9
Unit III	**Object Initialization and Overloading** Types of Constructors - Dynamic Initialization of Objects - Destructors Operator overloading - function Overloading - Manipulation of Strings	9
Unit IV	**Inheritance** Derived Classes - Types of inheritance - Virtual Base Classes - Abstract Classes - Pointers to Derived Classes - Virtual base class - Method Overriding - Pure Virtual Functions	9
Unit V	**File operations and Exception handling** Classes for File Operations - File Modes - opening and closing a File - Basics of Exception Handling - Try-Catch block - Case Studies on Real Time Applications	9
	Practical Component	
Exercises	**1.** Write a Program to Read and Print Number Input from the User 2. Write a simple program using a class and objects 3. Write a program to demonstrate the usage of a constructor and destructor in a class 4. Write a program to overload + operator to add two complex numbers 5. Write a program to demonstrate the usage of function overloading 6. Write a program to display employee information using multiple inheritance 7. Write a program to demonstrate multilevel inheritance	30

	8. Write a program to copy a file from one location to another location	
Recommended Learning Resources		
Print Resources	1. E Balagurusamy, “Object oriented Programming with C++”, Seventh edition, Tata McGraw Hill, 2020.	
Syllabus Design: Dr. T. Vengattaraman, Associate Professor, PUDoCS		

Year	II	Course Code: CSCS202 Course Title: Data Structures	Credits	4
Sem.	III		Hours	75
			Category	C
Course Prerequisites, if any	Introductory knowledge about Computing			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	● Learn basic terminologies of linear and nonlinear data structures and algorithms ● Understand the concept of polynomial addition and sparse matrices using arrays ● Apply linked lists to solve problems related to stacks, queues, and sparse matrices ● Understand the operations and traversals of binary trees ● Apply graph algorithms to solve problems like topological sorting and finding minimum cost spanning trees			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** Basic terminologies – Linear and Nonlinear data structures – Algorithm - Definition – Pseudo code – Analysis – Design Techniques		7	
Unit II	**Arrays, Stacks and Queues** Representation – Polynomial Addition – Sparse Matrices – Multidimensional Arrays - Stacks and Queues - Stack ADT – Operations – Evaluation of Expressions – Queue ADT – Operations – Application – Multiple Stacks and Queues		11	
Unit III	**Lists** Singly Linked Lists – Linked Stacks and Queues – Operations – Circularly Linked Lists – Equivalence Relations – Sparse Matrices – Doubly Linked Lists		9	
Unit IV	**Trees** Basic Terminologies – Binary trees – Representation, Operations, Traversals, Types – Applications of Trees		9	
Unit V	**Graphs** Basic Terminologies – Representation, Operations, Traversals – Applications - Shortest path problem, Topological sorting, Minimum Cost Spanning trees		9	
Practical Component				
Exercises	1. Searching Algorithms (with the number of key comparisons) - Sequential, Binary and Fibonacci search algorithms 2. Evaluation of arithmetic expression 3. Stack, Queue, Circular queue, priority queue 4. Singly Linked List, Doubly Linked List, Circular Linked List 5. Tree Traversal techniques 6. Graph Traversal techniques 7. Dijkstra's Algorithm to obtain the shortest paths		30	

Recommended Learning Resources	
Print Resources	1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Fundamentals of Data Structures in C", India University Press, Second Edition, 2008 2. Debasis Samanta, "Classic Data Structures", Prentice-Hall of India, Pvt. Ltd., India, Seventeenth Printing, Second Edition, 2009 3. Dinesh P Mehta & Sartaj Sahni, Handbook of Data Structures and Applications, Second Edition, Chapman and Hall, 2020
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS	

Year	II	Course Code: CSCS203 Course Title: System Software	Credits	4
Sem.	III		Hours	75
			Category	C
Course Prerequisites, if any	• Knowledge of digital logic design • Introductory knowledge in problem Solving			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand basic computer architecture via Simplified Instructional Computer (SIC) • Analyze differences in assemblers and machine features • Apply dynamic linking and bootstrap loaders in program preparation • Design macros demonstrating machine features • Apply machine-independent compiler features in design			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** System Software and Machine Architecture – Simplified Instructional Computer (SIC) – Traditional (CISC) Machines – RISC Machines		9	
Unit II	**Assemblers** Basic Assembler Functions – Machine Dependent and Machine Independent Assembler Features – One-Pass Assemblers – Multi Pass Assemblers – MASM assembler – SPARC assembler		9	
Unit III	**Loaders and Linkers** Basic Loader Functions – Machine Dependent and Machine Independent Loader Features – Linkage Editors – Dynamic Linking - Bootstrap Loaders		9	
Unit IV	**Macro Processors** Basic Macro Processor Functions – Machine Dependent and Machine Independent Macro Processor Features – Macro Processor Design Options		9	
Unit V	**Compilers** Basic Compiler Functions – Machine-Dependent Compiler Features – Machine Independent Compiler Features – Compiler Design Options – YACC		9	
Practical Component				
Exercises	1. Simulate a simple arithmetic operation (e.g., addition, subtraction) in both a CISC-like and RISC-like manner. The CISC simulation should perform the operation in a single step, while the RISC simulation should break it down into simpler steps 2. Design a program that translates a small set of assembly-like instructions (define your simple instruction set) into a simulated machine code. Your program should handle basic operations like load, store, add, and subtract 3. Design a program that simulates the basic functions of a linker and loader for a simplified computational system 4. Implement a simple macro processor that allows for the definition and expansion of macros within a text file. The macros should perform simple text replacement or predefined operations (like incrementing a number) 5. Design and implement a simple arithmetic expression evaluator using YACC. The evaluator should be capable of		30	

	handling basic arithmetic operations (+, -, *, /) and correctly respects the standard mathematical precedence of operations and handles parentheses to alter the precedence order	
Recommended Learning Resources		
Print Resources	1. Leland L. Beck, D. Manjula "System Software – An Introduction to Systems Programming", Third Edition, Pearson India, 2007. 2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Second Edition, Pearson Addison Wesley, 2023.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	II	Course Code: CSCS204 Course Title: 3D Modelling & Animation	Credits	3
Sem.	III		Hours	60
			Category	B
Course Prerequisites, if any	Basic Computer Knowledge			
Internal Assessment Marks: 50	End Semester Marks: 50	Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the basics of 3D modeling and animation concepts. • Learn the various stages of the production pipeline. • Acquire skills to handle digital images, videos, and process them • Become proficient in the usage of 3D modeling and adding visual effects, lighting, and rendering • Develop a model for a given specification • Develop an animated game, story, virtual tour of a building, etc.			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Defining 3D Animation, Exploring the 3D Animation Industry – History of 3D Animation: Early Computers – The Dawn of Computer Animation – The Building Blocks of 3D Animation – The Foundations of Modern Computing – 3D Animation Achieves Commercial Success – The Refining of 3D Animation.	9
Unit II	**Production Pipeline** Understanding the Production Pipeline's Components- Working in 3D Animation Preproduction – Working in 3D Animation Production – Working in 3D Animation Postproduction – Using Production Tools	9
Unit III	**Understanding Digital Imaging and Video** Understanding Digital Imaging – Understanding Digital Video - Exploring Animation, Story, and Pre-visualization: Using Principles of Fine Art and Traditional Animation- Building a Good Story – Using Pre-visualization Techniques	9
Unit IV	**Understanding Modeling and Texturing** Modeling: Polygons, NURBS, Subdivision Surfaces – Texturing: UVs, Texture Maps, Texturing Workflows – Rigging and Animation	9
Unit V	**Understanding Visual Effects, Lighting, and Rendering** Creating Visual Effects – Lighting – Rendering – Hardware and Software Tools of the Trade: Choosing a computer – Using Monitors / Displays – Working with Graphics Tablets – Using 3D Scanners – Setting Up Render Farms – Finding Data Storage Solutions – Choosing Software	9
	Practical Component	
Exercises	1. Implementing basic rendering techniques and effects 2. Developing storyboards, scripts / screenplay, 3D Production layout for a sample scene Ex: Friends meeting at a bus stop 3. Creating 3D models of characters, props, and environments for the above scene 4. Adding visual effects to the above scene 5. Adding texturing and minimal animation to the above scene 6. Setting up lighting and rendering scenes to achieve desired visual results for early morning moon and night time happening of the above scene	30

	7. Animating the above scene when the friends board the bus and the bus moves 8. Developing an animated game 9. Developing an animated story 10. Developing an animated virtual building tool	
Recommended Learning Resources		
Print Resources	1. Andy Beane, "3D Animation Essentials", First Edition, Wiley & Sons, 2012. 2. Magesh Chandramouli, "3D Modeling & Animation: A Primer", CRC Press, 2021. 3. Tony Mullen, "Introducing Character Animation with Blender", Second Edition, Wiley Publishers, 2011.	
Syllabus Design: Dr. T. Chithralekha, Professor, PUDoCS *Dr. S.L .Jayalakshmi, Assistant Professor, PUDoCS*		

Year	II	Course Code: CSCS205 Course Title: Game Programming	Credits	3
Sem.	III		Hours	60
			Category	B

Course Prerequisites, if any	Basic Programming Knowledge Computer Graphics	
Formative Assessment Marks: 50	Summative Assessment Marks: 50	Duration of ESA (Practical): 03 hrs
Course Outcomes	• Develop creativity and problem-solving skills • Enable students to develop games individually or in teams • Understanding the underlying technologies in game development	

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Game Designing** Magic Words – Importance of Skills a Game Designer Need – Important Skill – The Five Kinds of Listening – The Secret of the Gifted	9
Unit II	**3D Programming Concepts** Coordinate Systems – 3D Models – Shapes – Displaying 3D Models – Transformation – Rendering – Scene Graphs – 3D Audio – 3D Programming – Programmed Translation – Programmed Rotation – Programmed Scaling – Programmed Animation – 3D Audio – Basic Programming Concepts.	9
Unit III	**Game Programming** Torque Script – Strings – Objects – Data – Blocks – Game Structure – Server versus Client Design Issues – Common Functionality – Preparation – Root Main – Control Main – Initialization – Client – Server – Player – Running Emaga4	9
Unit IV	**Game Play** The Changes – Folders – Modules – Control Modules – Client Control Modules – Server Control Modules – Running Emaga5 – Creating GUI Elements	9
Unit V	**Game Sound and Music** Player Sounds – Footsteps – Weapon Sounds – Vehicle Sounds – Environmental Sounds – Interface Sounds – Music.	9
	Practical Component	
Exercises	1. Developing a Puzzle game 2. Developing a Multiplayer game using unity 3. Developing a 2D game 4. Developing a 3D game 5. Understand and develop the UI design in games 6. Understanding and apply the role of AI in Games	30
	Recommended Learning Resources	
Print Resources	1. Jesse Schell, “Art of Game Design”, A K Peters/CRC Press, Third edition, 2019. 2. Kenneth C. Finney, “3D Game Programming- All in One”, Cengage Learning, Inc, Third Edition, 2012.	

Syllabus Design: Dr. Sukhvinder Singh, Assistant Professor, PUDoCS

SEMESTER IV

Year	II	Course Code: CSCS206 Course Title: Computer System Architecture	Credits	4
Sem.	IV		Hours	75
			Category	C
Course Prerequisites, if any	Fundamentals of Computers			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the concept of digital electronics and logic circuits • Working with binary and arithmetic operations • Understand the organization of CPU and working principles • Understand the Input-Output organization in a computer • Understand the Memory organization in a computer			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Digital Logic Circuits** Digital Computers – Logic Gates – Boolean Algebra – Map Simplification – Combinational – Circuits – Flip-Flops – Sequential Circuits – Digital Components	9
Unit II	**Data Representation and Transfer** Datatypes – Complements – Fixed – Point Representation – Floating Point Representation – Register Transfer – Bus and Memory Transfer – Arithmetic – Logic and Shift Microoperations	9
Unit III	**CPU Organization** Register and Stack – Instruction Format – Addressing Modes – Data Transfer and Manipulation – Program Control – RISC – Basics of Pipelining	9
Unit IV	**Input-Output Organization** Peripheral devices – I/O Interface – Asynchronous data transfer – Modes of transfer – Priority Interrupt – DMA – Serial Communication	9
Unit V	**Memory Organization:** Memory Hierarchy – Main Memory – Auxiliary Memory – Associative Memory – Cache Memory – Virtual Memory – Memory Management Hardware	9
	Practical Component	
Exercises	1. Simplify Boolean expressions using Karnaugh maps 2. Design a combinational circuit 3. Implementing Logical Left and Right Shifts 4. Understand different data types and how to calculate complements 5. Evaluate performance improvement through instruction level parallelism 6. Analyze the effect of cache performance on system performance 7. Understand the impact of memory hierarchy on access time	30
	Recommended Learning Resources	
Print Resources	1. Morris Mano, Computer System Architecture, Pearson Education, 2017.	
Syllabus Design: Dr. Sukhvinder Singh, Assistant Professor, PUDoCS		

Year	II	Course Code: CSCS207 Course Title: Design and Analysis of Algorithms	Credits	4
Sem.	IV		Hours	75
			Category	C
Course Prerequisites, if any	Basic Knowledge in Data Structures and Programming			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Analyze the efficiency of algorithms and compare their performance using appropriate metrics • Understand the general approach of Brute Force and Divide and Conquer algorithms • Understand the principles of the Greedy Method in algorithm design • Understand the principles of Dynamic Programming • Understand the principles of Backtracking and branch and bound strategies			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Notation of Algorithm – Analysis of Algorithm Efficiency – Asymptotic Notations and Basic Efficiency classes – Mathematical Analysis of Non-recursive and recursive Algorithms	9
Unit II	**Divide and Conquer** Brute Force and Divide and conquer – Binary Search – Finding the maximum and minimum – merge sort – quick sort	9
Unit III	**Greedy Method** General method – Knapsack problem – Job Sequencing – Spanning Trees – Prims's Algorithm and Kruskal's Algorithm	9
Unit IV	**Dynamic Programming** General method – Principle of Optimality – Multistage Graphs – 0/1 Knapsack – Travelling Salesman Problem	9
Unit V	**Backtracking & Branch Bound** Backtracking – General Method – 8-Queen Problem – Sum of Subsets – Hamiltonian Cycles – Branch and Bound: Introduction FIFO Solution – LC Branch and Bound – 0/1 Knapsack	9
	Practical Component	
Exercises	1. Write recursive and iterative algorithms and analyze the time complexities of using Big-O notation 2. Implement and compare the efficiency of sorting algorithms (e.g., bubble sort, quicksort) on different input sizes 3. Implement merge sort and analyze its time complexity with different input sizes 4. Implement a greedy algorithm for the knapsack problem and analyze its efficiency 5. Implement Prim's algorithm for finding the minimum cost spanning tree 6. Implement Kruskal's algorithm for the same purpose and compare the results 7. Solve the 0/1 knapsack problem using dynamic programming and analyze the time complexity 8. Implement a backtracking solution for the subset sum problem and analyze its efficiency	30

Recommended Learning Resources	
Print Resources	1. Horowitz E. and Sahani S., "Fundamentals of Computer Algorithms", Second Edition, Universities press, 2008. 2. S. Sridar, "Design and Analysis of Algorithms", Oxford University Press, 2014.
Syllabus Design: Dr. T. Vengattaraman, Associate Professor, PUDoCS	

Year	II	Course Code: CSCS208 Course Title: Database Management Systems	Credits	4
			Hours	75
Sem.	IV		Category	C
Course Prerequisites, if any	Knowledge of data structures and file-handling			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the fundamentals of relational Model • Design real time applications using database query language (SQL) • Familiarize with the different kinds of PL/SQL objects • Understand the various database applications using the Relational model, ER model and EER model • Construct and normalize conceptual data models			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction to Relational model** Structure of relational database – Database schema – Keys – Schema diagram – Relational Query language – Relational Algebra		9	
Unit II	**Introduction to SQL** SQL data definition – basic structure of SQL Queries – set operations – null values – aggregate functions – nested subqueries		9	
Unit III	**Intermediate and advanced SQL** Join expressions, views – transaction – integrity constraints – functions and procedures – triggers		9	
Unit IV	**Database design using ER model** The Entity-Relationship model – complex attributes – mapping cardinalities – primary key – removing redundant attributes in entity sets – reducing ER diagrams to relational schemas – extended ER features		9	
Unit V	**Relational database design** Decomposition using functional dependencies – normal forms – functional dependency theory – algorithms for decomposition using functional dependencies – decomposition using multivalued dependencies		9	
Practical Component				
Exercises	1. Implement the DDL commands using SQL 2. Implement the DML commands 3. Implement the DDL constraints, DCL, and TCL commands 4. Implement various built functions and aggregate functions 5. Implement the various join operations 6. Implement the various nested subqueries 7. Creation and manipulation of Views 8. Practice the basics of PL/SQL [control structures] 9. Create the functions and procedures using PL/SQL 10. Create the Triggers using PL/SQL 11.		30	
Recommended Learning Resources				
Print Resources	1. Abraham Silberschatz, Henry F. Korth and S. Sundarshan, “Database System Concepts “, Seventh Edition, McGraw Hill International Edition, 2021.			

	2. Brumm B, “Beginning Oracle SQL for Oracle Database 18c: From Novice to Professional”, First Edition, Apress, 2019. 3. Kevin Loney, Bob Bryla, “Oracle Database 12c: The Complete Reference”, First Edition, McGraw Hill, 2013.
Syllabus Design: Dr. S. L. Jayalakshmi, Assistant Professor, PUDoCS	

Year	II	Course Code: CSCS209 Course Title: Embedded Application Development	Credits	4
Sem.	IV		Hours	75
			Category	C
Course Prerequisites, if any	• Microprocessor & Microcontrollers introduction • Assembly Language Programming • Operating System and Computer Organization Concepts			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the basics of Embedded Systems • Appreciate the application domains of Embedded Systems • Gain proficiency in programming embedded systems • Explore interfacing techniques for sensors, actuators, and other peripheral devices commonly used in embedded applications • Develop skills in designing, implementing, and debugging embedded software			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Embedded Systems and General-purpose Computer Systems – History – Classifications – Applications – Purpose of Embedded Systems – Characteristics and Quality Attributes	9
Unit II	**Embedded Systems** Application specific – washing machine – domain specific – automotive Embedded Hardware: Memory – I/O – Interrupt – Processors – External peripherals Peripherals: Control and Status Registers – Device Driver – Timer Driver – Watchdog Timers	9
Unit III	**Microcontrollers** Microcontrollers and Embedded processors – Overview of 8051 family. 8051 hardware – I/O pins – Ports – Circuits – External Memory Programming: Data Types – I/O Programming – Logic operations – Data conversion Programs	9
Unit IV	**Designing Embedded System with 8051 Microcontroller** Factors to be considered in selecting a controller – 8051 Microcontroller – Designing with 8051 Programming: Structure of embedded program – infinite loop – compiling, linking & debugging	9
Unit V	**Real Time Operating System (RTOS)** Operating system basics – Types of OS – Real-Time Characteristics – Selection Process of an RTOS Design and Development: Embedded system development Environment – IDE – types of file generated, disassembler – de-compiler – simulator – emulator and debugging, embedded product development life-cycle, trends in embedded industry	9
	Practical Component	
Exercises	1. Configure timer control registers of 8051 and develop a program to generate given time delay 2. Port I/O: Use one of the four ports of 8051 for O/P 3. interfaced to eight LED's. Simulate binary counter (8 bit) on LED's 4. Serial I/O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages	30

	to PC and display on PC screen. Signify end of message by carriage return 5. Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope 6. Interface the microcontroller with external devices (e.g., sensors, displays, or other microcontrollers) using serial communication. Implement simple data exchange protocols and verify communication 7. Generate PWM signals to control the brightness of LEDs or the speed of a motor. Experiment with different duty cycles and frequencies 8. Write programs to store and retrieve data from non-volatile memory (e.g., EEPROM or Flash). Implement dynamic memory allocation techniques using RAM	
Recommended Learning Resources		
Print Resources	1. Shibu K V, "Introduction to Embedded Systems" Second Edition, Tata McGraw Hill, 2017. 2. Rajkamal, “Embedded Systems - Architecture, Programming and Design", Third Edition, McGraw Hill Education, 2008.	
Syllabus Design: Dr. S.K.V. Jayakumar, Professor, PUDoCS		

SEMESTER V

Year	III	Course Code: CSCS301 Course Title: Operating Systems	Credits	4
			Hours	75
Sem.	V		Category	C
Course Prerequisites, if any	Knowledge of computers & computer organization			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• To understand the basic concepts of Operating System and Process • To learn the various mechanisms of CPU scheduling, process synchronization and deadlocks • To understand how the memory is utilized • To analyze various File System methods and Disk scheduling algorithms • Evaluate system structures in various operating systems, such as Linux and Windows and identifying similarities and differences			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Overview and Process management** Introduction: Operating System Structures – Operating systems services – System calls. Process Management**:** Process Concept – process scheduling – operation on processes – Inter process communications – Threads		9	
Unit II	**Scheduling algorithms and Process Synchronization** CPU Scheduling: Basic Concepts – Scheduling Algorithms Process Synchronization**:** Critical Section problem – Semaphores – Classical problems of synchronization – Monitors Deadlock: Deadlock Characterization – Deadlock Handling – Deadlock Prevention – Deadlock Avoidance – Deadlock Detection – Deadlock Recovery		9	
Unit III	**Memory Management** Main Memory: Contiguous Memory Allocation – Paging – Structure of the Page Table – Swapping Virtual Memory: Demand Paging – Page Replacement – Thrashing		9	
Unit IV	**Storage Management** Mass Storage structure: Overview – HDD (Disk) Scheduling – storage management – RAID Structure File Systems: File concepts – Access methods – Directory Structure – File Protection – File system Implementation – File System Structure – File System Operations – Allocation methods		9	
Unit V	**Case Studies** The Linux system: Design principles – kernel modules – process management – Scheduling – Memory Management – Linux File System Windows Operating system: Systems components – Windows File System		9	
Practical Component				
Exercises	1. Practice File handling utilities, Process utilities, Disk utilities, and Networking commands 2. Write a program to implement various system call operations		30	

	3. Write a program to demonstrate various File management operations 4. Write a program to simulate CPU scheduling algorithms: FCFS, SJF, Round Robin, and priority 5. Write a program to simulate Intra & Inter – Process Communication (IPC) techniques: Pipes, Messages Queues, and Shared Memory 6. Write a program to simulate solutions to Classical Process Synchronization Problems: Dining Philosophers, Producer – Consumer, Readers – Writers 7. Write a program to simulate Bankers Algorithm for Deadlock Avoidance 8. Write a program to simulate Page Replacement Algorithms: FIFO, Optimal, LRU 9. Write C programs to simulate implementation of HDD Scheduling Algorithms: FCFS, SCAN, C–SCAN 10. Case study on Linux and Windows Operating systems features and prepare a report on the same	
Recommended Learning Resources		
Print Resources	1. Abraham Silberschatz Peter B Galvin, G. Gagne, “Operating Systems Concepts”, Tenth Edition, Addison Wesley, 2018. 2. William Stallings, “Operating Systems: Internals and Design Principles”, Tenth Edition, Prentice Hall, 2021.	
Syllabus Design: Dr. S. L. Jayalakshmi, Assistant Professor, PUDoCS		

Year	III	Course Code: CSCS302 Course Title: Mathematical Foundations of Computer Science	Credits	4
Sem.	V		Hours	75
			Category	A
Course Prerequisites, if any	Basic Knowledge in Mathematics			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA(Theory): 03 hrs.		
Course Outcomes	• Understand logical statement structures • Apply operations in problem-solving • Analyze integer representations and congruences • Understand counting principles • Evaluate combinatorial solutions			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Logic and Proofs** Propositional Logic – Predicates and Quantifiers – Rules of Inference – Proofs – Methods and Strategy	15
Unit II	**Basic Structures** Sets – Functions – Sequences and Summations – Matrices Relations – properties – representation	15
Unit III	**Number Theory** Divisibility and Modular Arithmetic – Integer Representations and Algorithms – Primes and Greatest Common Divisors – Congruences	15
Unit IV	**Induction and Recursion** Mathematical Induction – Strong Induction and Well Ordering – Recursive Definitions and Structural Induction	15
Unit V	**Counting** Basics – Pigeonhole principle – Permutations and Combinations – Binomial Coefficients	15
	Practical Component	
-	-	-

Recommended Learning Resources	
Print Resources	1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Seventh Edition, McGraw Hill, Seventh Edition, 2017. 2. Trembley. J.P and Manohar. R., "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill, 2020.
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS	

<table>
<tr><td>Year</td><td>III</td><td colspan="2" rowspan="2">Course Code: CSCS303
Course Title: Computer Networks</td><td>Credits</td><td>4</td></tr>
<tr><td>Sem.</td><td>V</td><td>Hours</td><td>75</td></tr>
<tr><td colspan="4"></td><td>Category</td><td>C</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">Fundamentals of Computers</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="2">End Semester Marks: 60</td><td colspan="3">Duration of ESA (Theory): 03 hrs.
Duration of ESA (Practical): 03 hrs.</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">● Learn the basics of Network topology
● Learn about the various physical network media
● Understand the functionalities of all the network layers
● Familiarize the protocols of different layers
● Able to implement the various network protocols</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Content</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Introduction
Introduction to Networks – Topology – Network Architecture – Reference Models – Transmission Media-Multiplexing – Switching</td><td colspan="2">9</td></tr>
<tr><td>Unit II</td><td colspan="3">Data link layer
Design Issues – Error Detection and Correction – Elementary Data – Link Protocols – Sliding window Protocols</td><td colspan="2">9</td></tr>
<tr><td>Unit III</td><td colspan="3">Network Layer
Design Issues – Routing – Logical Addressing – IP Working- IPV4 Vs IPV6 – Address Mapping – delivery – Forwarding and routing</td><td colspan="2">9</td></tr>
<tr><td>Unit IV</td><td colspan="3">Transport Layer
The Transport Service – Service provided to the Upper Layers – Flow Control & Buffering – TCP Congestion Control – UDP – TCP Vs UDP</td><td colspan="2">9</td></tr>
<tr><td>Unit V</td><td colspan="3">Application layer
Domain Naming System – DNS Namespace – Resource Records – Name Servers – Electronic mail – Messages Formats – Message Transfer</td><td colspan="2">9</td></tr>
<tr><td colspan="6">Practical Component</td></tr>
<tr><td>Exercises</td><td colspan="3">1. Implementation of Basic Chat
2. Implementation of Multiple User Chat
3. Implementation of File Transmission
4. Implementation of Simple Mailing Application
5. Implementation of Client Server Application
6. Given IP address and subnet mask, Computation of
(i) Subnet addresses
(ii) Number of hosts in each subnet
(iii) IP addresses of hosts in each subnet
7. Implementation of Error Detection / Error Correction Techniques
8. Implementation of socket program Remote Procedure Call
9. Implementation of any one routing protocol
10. Implementation of congestion control protocol</td><td colspan="2">30</td></tr>
</table>

Recommended Learning Resources	
Print Resources	1. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, Fifth Edition, Prentice Hall publisher, 2022. 2. Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Fifth Edition, Morgan Kaufmann Publishers Inc., 2015. 3. James F. Kurose, Keith W. Ross,” Computer Networking - A Top-Down Approach Featuring the Internet”, Seventh Edition, Pearson Education, 2022.
Syllabus Design: Dr. G. Krishnapriya, Assistant Professor, PUDoCS	

Year	III	Course Code: CSCS304 Course Title: Theory of Computation	Credits	4
Sem.	V		Hours	75
			Category	A
Course Prerequisites, if any	• Knowledge in Mathematics for Computer Science			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs.		
Course Outcomes	• Understand foundational concepts of formal languages • Apply regular expressions to create DFA for lexical analyzers • Analyze equivalence and transformations between NFA, DFA, and TG • Evaluate context-free grammars and limitations of regular grammars • Design models using PDA			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Languages** Alphabets – String – Language – Basic Operations on Language – Concatenation – Union – Kleene Star		15	
Unit II	**Regular Expressions and Finite Automata** Regular expressions – Deterministic finite automata (DFA)		15	
Unit III	**Regular Languages** Non-Deterministic Finite Automata (NFA) – Relationship Between NFA and DFA – Transition Graphs (TG) – Properties of Regular Languages – The Relationship Between Regular Languages and Finite Automata – Kleene's Theorem		15	
Unit IV	**Non-Regular Languages and Context Free Grammars** Pumping Lemma for Regular Grammars – Context-Free Grammars (CFG)		15	
Unit V	**PDA and Context-Free Languages (CFL)** Deterministic And Non-Deterministic Pushdown Automata (PDA) – Parse Trees – Leftmost Derivation – Pumping Lemma for CFL – Properties Of CFL		15	
Practical Component				
-	-		-	
Recommended Learning Resources				
Print Resources	1. Cohen, D. I. A, "Introduction to Computer Theory", Second Edition, Wiley India, 2011. 2. Lewis, H.R. & Papadimitriou, H. R., "Elements of the Theory of Computation", Second Edition, Prentice Hall of India (PHI), 2015.			
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS				

SEMESTER VI

<table>
<tr><td rowspan="2">Year</td><td rowspan="2">III</td><td rowspan="2" colspan="2">Course Code: CSCS306
Course Title: Management Strategies and Concepts</td><td>Credits</td><td>4</td></tr>
<tr><td>Hours</td><td>75</td></tr>
<tr><td>Sem.</td><td>VI</td><td>Category</td><td>A</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">NIL</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="2">End Semester Marks: 60</td><td colspan="3">Duration of ESA (Theory): 03 hrs.</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">• Understand the fundamentals of Management Theories
• Learn the management & communication Process Concepts
• Analyse the performance of decentralized and centralized organizational structures
• Analyse the different leadership styles and their effects on team performance and organizational culture
• Evaluate the effectiveness of the strategies in enhancing productivity and efficiency</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Content</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Management Theories
Science Theory and Practice – Management and Society – Social Responsibility and Ethics – The nature and purpose of planning – objectives – Strategies Policies and planning premises</td><td colspan="2">15</td></tr>
<tr><td>Unit II</td><td colspan="3">Decision Making
Process of decision making – organizing – Nature and purpose of organizing – Basics of departmentalization – Line/Staff Authority and Decentralization – Effective Organizing and organizational structure & culture</td><td colspan="2">15</td></tr>
<tr><td>Unit III</td><td colspan="3">Human Resource Management & Selection
Staffing – Manpower planning – Recruitment & Selection – Performance appraisal and career strategy – Organizational development</td><td colspan="2">15</td></tr>
<tr><td>Unit IV</td><td colspan="3">Managing the Human factor
Motivation – Leadership – Communication</td><td colspan="2">15</td></tr>
<tr><td>Unit V</td><td colspan="3">The System & Process of Controlling
Control techniques and Information Technology – Productivity and Operations Management – Overall and Preventive Control – Towards a Unified – Global management theory</td><td colspan="2">15</td></tr>
<tr><td colspan="6">Recommended Learning Resources</td></tr>
<tr><td>Print Resources</td><td colspan="5">1. Herald Knootz and Heinz Weihrich, “Essentials of Management”, Eleventh Edition, McGraw-Hill Publishing Company, 2020.
2. Fred R. David and Forest R. David, "Strategic Management: Concepts and Cases", Prentice Hall India Learning Private Limited, Sixteenth Edition, 2020.</td></tr>
<tr><td colspan="6">Syllabus Design: Dr. S. L. Jayalakshmi, Assistant Professor, PUDoCS</td></tr>
</table>

Year	III	Course Code: CSCS307 Course Title: Software Engineering Theory and Practice	Credits	4
Sem.	VI		Hours	75
			Category	C
Course Prerequisites, if any	• Basic knowledge of programming and information systems			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the fundamental concepts of design thinking • Analyze and document the software requirements • Apply appropriate software engineering design concepts to develop software. • Apply software testing strategies • Understand and consider the significance of security in software development process			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction to Design Thinking** Design process - Traditional design - Design thinking - Existing sample design projects - Study on designs around us - Compositions/structure of a design - Innovative design - Breaking of patterns - Reframe existing design problems - Principles of creativity Empathy - Customer Needs - Insight-leaving from the lives of others/standing on the shoes of others - Observation.	9
Unit II	**Software Engineering and Software Requirements** Defining software engineering, Software life cycle models, Selection of a life cycle model - Requirements engineering, Types of requirements, Feasibility studies, Requirements elicitation, Requirement analysis, Requirement documentation, Requirement validation.	9
Unit III	**Software Project Planning** Size estimation, Cost estimation, Models, Constructive cost model, Software risk management, Software design, Modularity, Strategy of design, Function oriented design, Object oriented design.	9
Unit IV	**Testing Strategies** A strategic approach to software testing, Test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, The art of Debugging.	9
Unit V	**Secure Software Engineering** Introduction - The problem – Software assurance and software security – Threats to software security – Software insecurity – Benefits of detecting software security defects early – Managing secure software development – Defining Properties – Influencing the security properties of software – To assert and specify desired security properties.	9
	Practical Component	
Exercises	1. Conceptualize a novel app that will help to save: a) Energy b) Water c) Food 2. Apply the phases of Software Development Life Cycle	30

	for the following applications and develop the same : a) Library Management System b) Hospital Management System 3. Design the above two systems with security features and implement the same.	
Recommended Learning Resources		
Print Resources	1. Tim Brown, "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation", First Edition, HarperCollins Publishers Ltd, 2019. 2. Roger S. Pressman, Bruce Maxim, "Software Engineering, A Practitioner's Approach", Ninth Edition, McGraw Hill International Edition, 2023. 3. Julia H. Allen, "Software Security Engineering: A Guide for Project Managers", First Edition, 2008.	
Syllabus Design: Dr. T. Chithralekha, Professor, PUDoCS ***Dr. G. Krishnapriya, Assistant Professor, PUDoCS***		

<table>
<tr><td>Year</td><td>III</td><td colspan="2" rowspan="2">Course Code: CSCS308
Course Title: Distributed Systems</td><td>Credits</td><td>4</td></tr>
<tr><td rowspan="2">Sem.</td><td rowspan="2">VI</td><td>Hours</td><td>75</td></tr>
<tr><td>Category</td><td>C</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">Basic knowledge in operating systems and computer networks</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="2">End Semester Marks: 60</td><td colspan="3">Duration of ESA (Theory): 03 hrs.
Duration of ESA (Practical): 03 hrs.</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">• Learn basic concepts of Distributed Systems
• Understand state-of-the-art distributed system
• Design and develop Client/Server Applications
• Learn to setup fault tolerance and replication servers
• Design and implement CORBA and DCOM</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Content</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Introduction
Definition – Goals – Hardware and Software Concepts – Client/Server Model Communication – Layered Protocols RPC – Remote Object Invocation – Message Oriented Communication</td><td colspan="2">9</td></tr>
<tr><td>Unit II</td><td colspan="3">Client Server
Client Server and Naming Entity – Threads – Client Server – Code Migration – S/W Agents – Naming Entity – Location Mobile Entity</td><td colspan="2">9</td></tr>
<tr><td>Unit III</td><td colspan="3">Synchronization
Distributed Transactions – Synchronization – Clock Synchronization – Logical Clocks – Global States – Election Algorithms – Mutual Exclusion – Distributed Transaction Consistency and Replication – Data Centric Consistency – Fault Tolerance – Distributed Commit – Recovery</td><td colspan="2">9</td></tr>
<tr><td>Unit IV</td><td colspan="3">Distributed Objects
Distributed Object Database System – CORBA – DCOM – GLOBE</td><td colspan="2">9</td></tr>
<tr><td>Unit V</td><td colspan="3">Distributed File System
Introduction - Distributed File System – Distributed Document based System – WWW – Distributed Coordination based System – JINI</td><td colspan="2">9</td></tr>
<tr><td colspan="6">Practical Component</td></tr>
<tr><td>Exercises</td><td colspan="3">1. Perform arithmetic operation using RMI
2. Calculate simple and compound interest using RMI
3. Implementation of ATM using RMI
4. Implementation of Telephone Directory using RMI
5. Implementation of Quiz Server using Servelets
6. Implementation of Online Shopping System using servelets</td><td colspan="2">30</td></tr>
</table>

	7. Implementation of matrimonial System using servelets 8. Implementation of servelet based Airline Reservation system 9. Create a Word Document with text using DCOM and Visual Basic	
Recommended Learning Resources		
Print Resources	1. Andrew S. Tanenbaum, Maarten van Steer, "Distributed Systems: Principles and Paradigms", Third Edition, Prentice Hall India, 2017. 2. George Couloursis, Jean Dollomore and Tim Kinderberg, "Distributed Systems: Concepts and Design", Addison-Wesley, Fifth Edition, 2011.	
Syllabus Design: Dr. T. Sivakumar, Assistant Professor, PUDoCS		

Year	III	Course Code: CSCS309 Course Title: Operations Research	Credits	4
Sem.	VI		Hours	75
			Category	A
Course Prerequisites, if any	Basic Mathematical and Problem-Solving Skills			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs.		
Course Outcomes	• Understand and comprehend the basics of Linear Programming Problem (LPP) • Learn LPP solving methods and explore duality in LPP • Solve assignment problems and their variants • Find feasible and optimal solutions for transportation problem • Perform critical path analysis and reviewing of a project			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Operation Research – Definition – Characteristics – Techniques – Applications. LPP – Introduction – Applications and components of LPP – Steps in solving LPP	15
Unit II	**LPP** Mathematical formulation – Graphical method – Simplex method – Artificial variables – Big-M method – Two-phase method – Degeneracy and unbound solutions – Duality in LPP – Formulation – Relationship between primal and dual problems	15
Unit III	**Assignment Model** Mathematical formulations – Hungarian Method – Variants of the Assignment problem	15
Unit IV	**Transportation Problem** Mathematical formulation – Finding basic feasible solutions – NWCR, LCM and VAM – Optimal solution – MODI method	15
Unit V	**Network Scheduling** Introduction – Basic components – Logical sequencing – Rules of network construction – Concurrent Activities – Critical Path Analysis -Activity Time and Floats – Project Evaluation and Review Technique (PERT) – Three Time Estimates – Critical Path Analysis of PERT network – Probability of completion of Project	15
	Practical Component	
	-	

Recommended Learning Resources	
Print Resources	1. Kanti Swarup, P.K. Gupta, Man Mohan, "Operations Research", Sultan Chand & Sons, Twentieth Edition, 2023. 2. Taha H.A., "Operations Research: An Introduction", Pearson Education, Tenth Edition, 2019.

Syllabus Design: Dr. G. Krishnapriya, Assistant Professor, PUDoCS
Revised by: Dr. M. Nandhini, Professor, PUDoCS

<table>
<tr><td>Year</td><td>III</td><td colspan="2" rowspan="2">Course Code: CSCS310
Course Title: Unix System Programming</td><td>Credits</td><td>4</td></tr>
<tr><td>Sem.</td><td>VI</td><td>Hours</td><td>75</td></tr>
<tr><td colspan="4"></td><td>Category</td><td>C</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">• Computer Organization and Architecture
• Operating System</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="2">End Semester Marks: 60</td><td colspan="3">Duration of ESA (Theory): 03 hrs.
Duration of ESA (Practical): 03 hrs.</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">• Understand Unix history, features, and system architecture
• Manage files, directories, processes, and memory
• Implement IPC with shared memory and semaphores
• Develop network applications using socket programming
• Write and execute shell scripts for text and pattern manipulation</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Content</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Introduction
Introduction to Unix – History – Salient features of Unix – Unix System Architecture – Unix Programming Environment – Unix Process</td><td colspan="2">9</td></tr>
<tr><td>Unit II</td><td colspan="3">Standard I/O, Process and Memory Management
File Management: File input/output – Directory related System Calls – Process Management
Processes: Creation – Execution – Termination – Process States – Process Control – Process groups – Thread – Memory Management</td><td colspan="2">9</td></tr>
<tr><td>Unit III</td><td colspan="3">Inter-Process Communication
Introduction to IPC – Shared Memory: Creating Shared Memory – Controlling Shared memory Segment – Process Synchronization: Semaphore</td><td colspan="2">9</td></tr>
<tr><td>Unit IV</td><td colspan="3">Socket Programming
Socket – Types of Sockets – Socket Data Structure – System Calls – I/O Models – Name and Address Conversion – Resource records</td><td colspan="2">9</td></tr>
<tr><td>Unit V</td><td colspan="3">Tools and Programming
Shell Scripting – Shell Scripting Operations – Text Manipulation – Pattern Matching – Text Transformation</td><td colspan="2">9</td></tr>
<tr><td colspan="6">Practical Component</td></tr>
<tr><td>Exercises</td><td colspan="3">1. Perform operations like file creation, deletion, copying, moving, listing directory contents
2. Write a shell script that takes a directory name as an argument and lists all files and directories inside it
3. Write programs using ‘fork’, ‘exec’, and ‘Wait’ system calls to create processes
4. Create a program that uses unnamed pipes for communication between a parent and its child process
5. Write a simple client-server application using TCP sockets where the client sends a message to the server, and the server echoes it back</td><td colspan="2">30</td></tr>
</table>

	6. Implement a program that manipulates file permissions, accesses file metadata (like inode information), and performs file locking 7. Develop a simple shell that can interpret commands, launch programs, and support basic piping and redirection	
Recommended Learning Resources		
Print Resources	1. Vineeta khemchandani, Dappan Anand, Mishra, Sandeep Harit, "Unix Programming", BPB Online, 2022.	
Syllabus Design: Dr. S. K. V. Jayakumar, Professor, PUDoCS		

Year	III	Course Code: CSCS311 Course Title: Network Programming	Credits	4
Sem.	VI		Hours	75
			Category	C
Course Prerequisites, if any	• Computer Networking Fundamentals • Programming Languages			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand client-server networking and socket API • Learn server architectures: single-threaded, multithreaded, and async servers • Implement message queues, caching, and HTTP handling • Understand various networking protocols (TCP, UDP, POP, IMAP, etc.) • Explore case studies using Cisco Packet Tracer, Network Simulator 2, and GNS3			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Client Server Networking – UDP – TCP – DNS – Client-server Model – Socket API – Socket Addresses	9
Unit II	**Architecture of Servers** Data and Errors on Internet: Strings and bytes – SSL/TLS – Architecture of Server – Single Threaded Server – Multithreaded Servers – Async Servers	9
Unit III	**Message Queues and Caches** Memory Caching - Hashing and Sharding - Message Queues – HTTP Client – Server Handling HTTP – World Wide Web – SMTP	9
Unit IV	**Protocols** TCP – UDP – POP – IMAP – IPV4 – IpV6 – BGP – Telnet – SSH – FTP – RPC	9
Unit V	**Case Studies** Cisco Packet Tracer – Network Simulator 2 – GNS3	9
	Practical Component	
Exercises	1. a) Study of different types of network cables and practically implement cross wired cable and straight through cable using clamping tool b) Study of network devices and network IP in detail 2. Study of network IP and practically connect the computers in LAN a) Study of basic network command and network configuration commands b) Configure a network topology using CPT 3. Configure a network using Distance vector/Link state routing protocol 4. Simulation of Sliding Window Protocol 5. Half Duplex Chat Using UDP 6. Full Duplex Chat Using TCP/IP	30
	Recommended Learning Resources	
References	1. John Galbraith, "Network Programming in Python: The Basic", First Edition, BPB Publications, 2022.	
Syllabus Design: Dr. S.K.V. Jayakumar, Professor, PUDoCS		

SEMESTER VII

<table>
<tr><td>Year</td><td>IV</td><td colspan="2" rowspan="2">Course Code: CSCS401
Course Title: Web Engineering</td><td>Credits</td><td>4</td></tr>
<tr><td>Sem.</td><td>VII</td><td>Hours</td><td>75</td></tr>
<tr><td colspan="4"></td><td>Category</td><td>C</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">Basic understanding of programming concepts</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="3">End Semester Marks: 60</td><td colspan="2">Duration of ESA (Theory): 03 hrs.
Duration of ESA (Practical): 03 hrs</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">• Understand the process of web publishing
• Acquire skills developing web pages using HTML
• Acquire skills to style the web pages using CSS
• Acquire skills to build server-side web components
• Explore the mobile web development process</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Component</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Introduction to World Wide Web
Introduction to web publishing – Web browsers – Web servers – Uniform Resource Locators – Using browser-based developer tools.</td><td colspan="2">9</td></tr>
<tr><td>Unit II</td><td colspan="3">Introduction to HTML and CSS
Structuring a web page with HTML – Basic elements – Lists – Links – Tables – Images – Forms
Using CSS to style a site – CSS for positioning – Integrating Multimedia elements</td><td colspan="2">9</td></tr>
<tr><td>Unit III</td><td colspan="3">Introduction to JavaScript
The structure – Operators – Variables – Control structures – Functions – Arrays – Objects – Validation</td><td colspan="2">9</td></tr>
<tr><td>Unit IV</td><td colspan="3">Introduction to PHP
Setting up the server – PHP language basics – built–in functions – library functions – using includes – database connectivity – sending email – cookies and sessions – File uploads</td><td colspan="2">9</td></tr>
<tr><td>Unit V</td><td colspan="3">Mobile Web
Mobile browsing needs – text on mobile web – design and page layout – links – images and multimedia – CSS for mobile – making use mobile features – Best practices</td><td colspan="2">9</td></tr>
<tr><td colspan="6">Practical Component</td></tr>
<tr><td>Exercises</td><td colspan="3">1. Build your resume using simple static html
2. Enrich your resume with CSS
3. Implement an HTML Form with JavaScript validation
4. Build a web application to demonstrate event handling in JavaScript
5. Add a server-side component to the task #3
6. Build a server-side data storage web application
7. Build a web application to demonstrate session handling
8. Build a web application to demonstrate cookies handling
9. Implement mobile web application
10. Implement file uploads in a web application</td><td colspan="2">30</td></tr>
<tr><td colspan="6">Recommended Learning Resource</td></tr>
</table>

Print Resources	1. Laura Lemay, Rafe Coburn, Jennifer Kyrnin, “Sams Teach yourself HTML, CSS & Javascript Web Publishing, Pearson Education, 2016.
Syllabus Design: Dr. K. S. Kuppusamy, Associate Professor, PUDoCS	

Year	III	Course Code: CSCS402 Course Title: System Modelling and Simulation	Credits	4
			Hours	75
Sem.	VII		Category	C
Course Prerequisites, if any	Basic knowledge in statistics			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the fundamentals of modeling and simulation • Learn about statistical models and input modelling • Understand the techniques for random number generation • Perform the simulation of dynamic systems • Verify the simulation models			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Simulation tool – Advantages and disadvantages of Simulation – Areas of application– Systems and system environment – Components of a system – Discrete and continuous systems – Model of a system – Types of Models – DESS – Simulation of queuing systems – General Principles	9
Unit II	**Statistical Models in Simulation** Review of terminology and concepts – Useful statistical models – Discrete distributions – Continuous distributions – Poisson process – Empirical distributions – General Principles – Characteristics of queuing systems – Queuing notation – Long-run measures of performance of queuing systems – Steady-state behavior of M/G/1 queue – Networks of queues	9
Unit III	**Random-Number Generation** Properties of random numbers – Generation of pseudo-random numbers – Techniques for generating random numbers – Tests for Random Numbers – Inverse transform technique Acceptance – Rejection technique	9
Unit IV	**Input Modeling** Data Collection – Identifying the distribution with data – Parameter estimation – Goodness of Fit Tests – Fitting a non-stationary Poisson process – Selecting input models without data – Multivariate & Time – Series input models – Types of simulations with respect to output analysis – Stochastic nature of output data – Measures of performance and their estimation	9
Unit V	**Simulation Models** Measures of performance and their estimation – Output analysis for terminating simulations – Output analysis for steady – state simulations – Verification, Calibration and Validation – Optimization, Model building, Verification and Validation – Verification of simulation models – Calibration and Validation of models, Optimization via Simulation	9
	Practical Component	
Exercises	1. Simulation of Random Numbers generation 2. Implement Chi-square goodness-of-fit test 3. Implement One-sample Kolmogorov-Smirnov test 4. Implement Test for Standard Normal Distribution	30

	5. Implement Monte-Carlo Simulation 6. Simulation of Single Server Queuing System 7. Simulation of Two-Server Queuing System 8. Simulate and control a conveyor belt system 9. Implement Two-sample Kolmogorov-Smirnov test	
Recommended Learning Resources		
Print Resources	1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol, "Discrete-Event System Simulation", Fifth Edition, Pearson Education, 2013. 2. Lawrence M. Leemis, Stephen K. Park, "Discrete-Event Simulation: A First Course", Pearson Education, 2013.	
Syllabus Design: Dr. G. Krishnapriya, Assistant Professor, PUDoCS		

Year	IV	Course Code: CSCS403 Course Title: Wireless Communication Networks	Credits	4
Sem.	VII		Hours	75
			Category	C
Course Prerequisites, if any	Knowledge in computer networks			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand basics of Wireless Communication Networks • Understand the Satellite Communications concepts and compare generations wireless communications • Explore IEEE 802.11WLAN standard • Explore WAP and its application • Understand WLAN technologies			

Unit No.	Course Component	Hours
Theory Component		
Unit I	**Introduction** Wireless Communication Technology – Antennas and Propagation – Antennas, Propagation Modes, Fading in the Mobile Environment – Signal Encoding Techniques – Signal Encoding Criteria, Digital Data – Analog Signals, Analog Data – Analog Signals, Analog Data – Digital Signals	9
Unit II	**Satellite Communications** Wireless Networking – Satellite Communications – Satellite Parameters and Configurations, Capacity Allocation – Frequency Division, Capacity Allocation –Time Division Cellular Wireless Networks – Principles of Cellular Networks, First Generation Analog, Second Generation – TDMA, CDMA, 3G Systems	9
Unit III	**Wireless LAN Standards** Evolution of IEEE 802.11 – Introduction to IEEE 802.11 – General Description – Medium Access Control (MAC) for the IEEE 802.11 – WLANs Physical Layer for IEEE 802.11 – WLANs; Radio Systems – IR Systems Applications	9
Unit IV	**Mobile IP** Introduction, operation of Mobile IP, Mobile IP terminologies, Wireless Access Protocols: Introduction, Architecture overview, Wireless application environment	9
Unit V	**Wireless LAN Technology** Wireless LAN – application, requirements, Technology: Infrared, spread spectrum, Narrowband microwave (radio), Introduction Bluetooth Technologies (Only Overview)	9
Practical Component		
Exercises	1. Study about different Wireless devices like Wi-Fi Dongler, Wireless Access Point, Antenna, Wi-Fi Router 2. Configure a wireless LAN using CISCO Packet Tracer 3. Develop a client server application using Wireless LAN 4. Simulate BlueTooth Communication after pairing in CISCO Packet Tracer	30

Recommended Learning Resource	
Print Resources	1. William Stallings, “Wireless Communications and Networks” 2nd edition, Pearson Prentice Hall, 2005.
Syllabus Design: Dr. T. Sivakumar, Assistant Professor, PUDoCS	

Year	IV	Course Code: CSCS404 Course Title: Artificial Intelligence	Credits	4
Sem.	VII		Hours	75
			Category	C
Course Prerequisites, if any	Basic Programming Skills			
Internal Assessment Marks: 40	End Semester Marks: 60		Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.	
Course Outcomes	● Familiarize with the diverse traits of a problem-solving agent ● Explore methods for tackling problems amidst different constraints ● Implement AI techniques in various applications ● Grasp the distinct models of learning ● Develop an expert system			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Foundation and History of AI – Intelligent Agents – Agents and Environments – The Concept of Rationality – Nature of Environments – Structure of Agents – Problem Solving Agents – Examples	9
Unit II	**Searching** Searching for Solutions, Uniformed Search Strategies – Heuristics Search Strategies – Local Search Algorithms and Optimization Problems – Hill Climbing- Simulated Annealing – Local Beam Search – Genetic Algorithms – Optimal Decisions in Games – Alpha-Beta Pruning	9
Unit III	**Agents** Logical Agents – Knowledge-Based Agents – The Wumpus World – Logic – Propositional Logic – Propositional Theorem Proving – Effective Propositional Model Checking – Agents Based on Propositional Logic	9
Unit IV	**First Order Logic** Introduction – Syntax and Semantics – Inference – Propositional Vs First-Order Inference – Unification and Lifting – Forward Chaining – Backward Chaining – Resolution	9
Unit V	**Learning** Forms of Learning – Supervised Learning – Learning Decision Trees – Hypothesis – Theory of Learning – Prolog – Programs – Data Objects	9
	Practical Component	
Exercises	1. Implement Breadth First Search 2. Implement Depth First Search 3. Implement Tic-Tac-Toe game 4. Implement 8-Puzzle problem 5. Implement Water-Jug problem 6. Implement Monkey Banana Problem 7. Implement Alpha-Beta Pruning 8. Develop an expert system using Prolog	30

Recommended Learning Resources	
Print Resources	1. S. Russell and P. Norvig, "Artificial Intelligence – A Modern Approach", Pearson Education, Third Edition, 2010. 2. Max Bramer, Logic Programming with Prolog, Springer, 2005.
Syllabus Design: Dr. P. Shanthi Bala, Professor, PUDoCS	

Year	IV	Course Code: CSCS405 Course Title: Compiler Design	Credits	4
Sem.	VII		Hours	75
			Category	C
Course Prerequisites, if any	• Knowledge in any programming language such as Java or C • Knowledge in Assembly Programming, Basic Arithmetic, and Data Structures			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the functional components of compilers • Apply knowledge of lexical analysis by implementing scanners • Analyze and differentiate between various parsing techniques • Evaluate and integrate syntax-directed definitions and type checking in compiler construction • Design and create components of a runtime environment and a code generator			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** Language Processors – Structure of a Compiler – Evolution of Programming Languages – Applications of Compiler Technology – Tool based Approach to Compiler Construction		9	
Unit II	**Lexical Analysis** Interface with Input – Parser and Symbol Table – Tokens, Patterns and Lexemes – Difficulties in Lexical Analysis – Error Reporting – Regular Definitions – Transition Diagrams – Lex		9	
Unit III	**Syntax Analysis** CFGs – Ambiguity – Associatively – Precedence – Top-Down Parsing – Recursive – Descent Parsing – FIRST and FOLLOW – LL (1) Grammars – Predictive Parsing – Bottom-Up Parsing – LR Parsing		9	
Unit IV	**Syntax Directed Definitions** Inherited and Synthesized Attributes – Dependency Graphs – Ordering the Evaluation of Attributes – L and S Attributed Definitions – Type Checking		9	
Unit V	**Run Time Environments** Storage Organization – Stack Allocation of Space – Parameter Passing – Symbol Table – Dynamic Storage Allocation **Code Generation** Issues in the Design of a Code Generator – Addresses in the Target Code – Basic Blocks and Flow Graphs – Optimization of Basic Blocks – Code Generator – Peep Hole Optimization		9	
Practical Component				
Exercises	1. Using Lex or a similar tool, implement a lexical analyzer for a simple programming language or a subset of an existing language 2. Write a recursive descent parser in a programming language of your choice for a simple arithmetic expression grammar that includes addition, subtraction, multiplication, division, and parentheses. Ensure your parser handles operator precedence correctly 3. Implement a program that builds a parse tree for an expression and evaluates its attributes according to your definitions			

	4. Implement a simple type checker that can handle basic data types (integers, floats), type conversions, and function/operator overloading 5. Create a simulation of a runtime environment that demonstrates stack allocation, parameter passing, and dynamic storage allocation 6. Given a set of basic blocks, implement an optimization routine that applies peephole optimization techniques	
Recommended Learning Resources		
Print Resources	1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, "Compilers: Principles, Techniques, & Tools", Second Edition, Pearson Addison Wesley, 2023. 2. Allen I. Holub, "Compiler Design in C", First Edition, Pearson India, 2015.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	IV	Course Code: CSCS406 Course Title: Cyber Security	Credits	4
Sem.	VII		Hours	75
			Category	C
Course Prerequisites, if any	Basic Knowledge of Programming and Information Security Principles			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Learn the definitions and categories of cybercrimes • Comprehend the tools and techniques employed in cybercrimes • Examine the legal frameworks surrounding cybercrime legislation • Assess the effectiveness of cybersecurity measures • Examine current cyber threats and vulnerabilities			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction to Cybercrime** Cybercrime Definition – Cybercrime and Information Security – Classification of Cybercrimes – Email Spoofing, Spamming, Data Diddling, Web Jacking, Hacking, Password Sniffing – Categories of Cybercrime – Passive attack – Active attack – Reconnaissance		9	
Unit II	**Tools and Methods used in Cybercrime** Cyberstalking – Cybercafe and Cybercrimes – Botnets – Proxy Servers and Anonymizers – Password Cracking – Keyloggers and Spyware – DoS and DDoS attacks – Virus and Worms – Trojan horses and Backdoors – SQL injection – Steganography		9	
Unit III	**Mobile and Wireless Devices** Proliferation of Mobile and Wireless Devices – Trends in Mobility – Security Challenges Posed by Mobile Devices – Authentication Service Security – Attacks on mobiles and cellphones – Credits Card Frauds in mobile and Wireless Computing Era – Organizational measures for Handling Mobile		9	
Unit IV	**Phishing and Identify Theft** Buffer Overflow – Phishing: Methods of Phishing, Phishing Techniques, Spear Phishing, Types of Phishing Scams, Phishing Tool Kits and Spy Phishing, Phishing Countermeasures – Identify Theft (ID Theft): Types of Identify Theft, Techniques of ID theft – ID Theft Counter Measures – Personally Identifiable Information		9	
Unit V	**Cybercrime and Cyber Security Legal Perspectives** The Indian IT Act – Challenges to Indian Law and Cybercrime Scenario in India – Digital Signatures and The Indian IT Act – Amendments to the Indian IT Act – Cybercrime and Punishment		9	
Practical Component				
Exercises	1. Create a simple program that encrypts and decrypts a text message using a basic cipher (e.g., Caesar cipher). Demonstrate encryption of a given plaintext and then decryption back to the original text 2. Simulate a basic SQL injection attack against a sample web application. Demonstrate how unauthorized access to data can be obtained through poorly sanitized input fields. Show the effect of the attack and suggest mitigation strategies		30	

	3. Use a password cracking tool on a set of hashed passwords. Demonstrate the process of cracking by identifying weak passwords from the hash values. Discuss the importance of strong password policies 4. Set up and configure a basic firewall on a network or computer system. Demonstrate how to block and allow specific traffic types. Test the firewall setup by attempting to access the protected resources with varying types of network traffic 5. Analyze a set of emails to identify characteristics of phishing attempts. Explain the indicators of phishing and suggest methods for verifying the authenticity of suspicious emails. Discuss the impact of phishing attacks and preventive measures 6. Create virtualized network environments with cybersecurity simulation software, guiding participants through defense strategies against various cyber-attacks 7. Equip participants with forensic analysis tools, presenting simulated cyber-attack scenarios to investigate, analyze evidence, and prepare forensic reports	
Recommended Learning Resources		
Print Resources	1. Nina Godbole and Sumit Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, First Edition, Wiley India Pvt. Ltd., 2011. 2. Anand Shinde, “Introduction to Cyber Security: Guide to the World of Cyber Security”, First Edition, Notion Press, 2021.	
Syllabus Design: Dr. M.Sathya, Assistant Professor, PUDoCS		

Year	IV	Course Code: CSCS407 Course Title: Internet of Things	Credits	4
Sem.	VII		Hours	75
			Category	C
Course Prerequisites, if any	Basic knowledge of programming and networking			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand IoT fundamentals, including design, protocols, and technologies • Explore domain-specific applications such as home automation and industry • Learn about M2M applications and system management • Develop IoT systems using platforms like Raspberry Pi • Manage IoT server and cloud infrastructure, focusing on security			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Definition, Characteristics of IoT, Physical Design of IoT, Protocols, Logical Design of IoT, IoT Enabled Technologies, IoT Levels and Templates	9
Unit II	**Domain Specific IoT Applications** Home Automation, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle	9
Unit III	**M2M and IoT System Management** M2M Applications, Software Defined Networks, Network Function Virtualization. Need for IoT System Management, Simple Network Management Protocol, IoT System Management with NETCOZF-YANG	9
Unit IV	**Developing IoT Systems** IoT Platforms Design Methodology, Steps for IoT Design, Case Study on IoT System for Weather Monitoring, Introduction to Raspberry PI, Interfaces (serial, SPI, I2C), Programming Raspberry Pi, IoT Devices	9
Unit V	**IoT Server and Cloud Management** Introduction to Cloud Storage Models and Communication APIs, Webserver – Web Server for IoT, Cloud for IoT, Security Management in an IoT System	9
	Practical Component	
Exercisers	1. Identify and list different types of IoT devices and their functionalities 2. Sketch a physical design for a home automation system using IoT devices 3. Compare and contrast different IoT protocols such as MQTT, CoAP, and HTTP 4. Set up a basic communication protocol between two IoT devices using MQTT 5. Discuss the role of cloud computing in enabling IoT solutions 6. Implement a simulation of the home automation system using IoT platforms like Arduino or Raspberry Pi 7. Investigate and compare M2M applications in industries such as healthcare and logistics	30

	8. Program a Raspberry Pi to collect weather data from sensors and display it on a web server 9. Explore different cloud storage models (e.g., public, private, hybrid) and their suitability for IoT applications 10. Implement security measures such as encryption and authentication in an IoT system using cloud-based services	
Recommended Learning Resources		
Print Resources	1. Arshdeep Bahga and Vijay Madisetti, “Internet of Things - A Hands-on Approach”, First Edition, Orient Blackswan Private Limited, 2015. 2. Rajesh Singh, Anita Gehlot, Bhupendra Singh, Sushabhan Choudhury, ” Internet of Things (IoT) Enabled Automation in Agriculture”, Second Edition, CRC Press, 2022.	
Syllabus Design: Dr. T. Vengattaraman, Associate Professor, PUDoCS		

SEMESTER VIII

Year	IV	Course Code: CSCS408 Course Title: Machine Learning	Credits	4
			Hours	75
Sem.	VIII		Category	C

Course Prerequisites, if any	Probability and Statistics	
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.
Course Outcomes	• Understand the basic concepts and types of Machine Learning (ML) • Prepare the data for ML model, train the model and evaluate the model's performance • Understand the fundamentals of features and feature engineering • Build a ML model with the appropriate supervised algorithm for the data • Build a ML model with the appropriate unsupervised algorithm for the data	

Unit No.	Course Content	Hours
	Theory Component	
UNIT I	**Introduction to Machine Learning** Human Learning – MachineLearning –Types of Machine Learning – Supervised learning – Unsupervised Learning – Reinforcement Learning – Applications **Preparing to model** Types of data – structure – quality and remediation – pre–processing	9
UNIT II	**Modelling and Evaluation** Selecting – Training – Model representation and interpretability – Performance evaluation **Feature Engineering** Introduction – Transformation – Feature subset selection – Issues in high dimensional data – Feature selection – Key drivers – Measures – process – Approaches	9
UNIT III	**Supervised Learning – Classification** Introduction – Example – Model – Learning steps – Algorithms – k–Nearest neighbor – Decision tree – Random Forest model – Support Vector machines	9
UNIT IV	**Supervised Learning – Regression** Introduction – Example – Model – Algorithms – Simple and Multiple linear regression – Assumptions – Main problems in regression analysis – Logistic regression – Maximum Likelihood estimation	9
UNIT V	**Unsupervised Learning** Introduction – Applications – Clustering – Types – Partitioning methods – Hierarchical clustering – Density-based methods – DBSCAN – The Apriori algorithm for association rule learning	9
	Practical Component	

Exercises	1. Develop a Python script that uses a decision tree classifier forprediction 2. Develop a ML model that runs a random forest for classification 3. Create a Python program that uses SVM to classify images fromthe MNIST dataset 4. Implement K–Means clustering to segment customers into groups based on their shopping data such as purchase historyand customer demographics 5. Implement a linear regression model **6.** Develop a program to perform multiple linear regression topredict house prices Implement logistic regression to classify emails as spam or notspam	30
Recommended Learning References		
Print Resources	1. Saikat Dutt, Chandramouli.S, Amit Kumar Das., " Machine Learning", Pearson, 2018. 2. Alpaydin, E., "Introduction to Machine Learning", MIT Press,Fourth Edition, 2020.	
Syllabus Design: Dr. M. Nandhini, Professor, PUDoCS		

Year	IV	Course Code: CSCS409 Course Title: Full Stack Development	Credits	3
Sem.	VIII		Hours	75
			Category	C
Course Prerequisites, if any	Basic programming concepts, OOPs, Web Technology, Database, any Scripting Languages			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Create responsive web pages using HTML and CSS • Develop interactive web applications with JavaScript • Understand the concept of ReactJS component-based architecture • Develop RESTful APIs with Node.js and Express.js • Design scalable MongoDB database schemas for web applications			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**HTML and CSS** Tags – Attribute and Elements – Comments – Lists and Links – Images and Tables – CSS to HTML – Selectors – Properties and Values – CSS Box Model – Margins – Padding – Borders – Text,Font Properties	9
Unit II	**Java Script** Internal and external script – Document and Window Object – Variables and Operators – Data Types and Type Conversion – Math and String Manipulation – Objects and Arrays – Conditional Statements – Functions – Java libraries – jQuery – Angular	9
Unit III	**ReactJS Development** Templating using JSX – Components – State and Props – Lifecycle of Components – Rendering List and Portals – Error Handling – Routers – Redux and Redux Saga – Immutable.js – Service Side Rendering – Unit Testing – Webpack	9
Unit IV	**NodeJS Development** Basics and Setup Console – Node js Command Utilities – Node js Module – Concepts – Events – Node js with Express js – Node js Database Access	9
Unit V	**MongoDB** SQL and NoSql Concepts – Create and Manage MongoDB – Migration of Data – MongoDB with PHP – MongoDB with NodeJS – Services – MongoDB with Python	9
	Practical Component	
Exercisers	1. Applying all the concept of HTML and CSS for the building of Web portal 2. Create a dynamic and interactive web page to interact with visitors and execute complex actions 3. Build complex UI interactions that communicate with the server in record time with JavaScript–driven pages 4. Build back–end services like APIs, Web App or Mobile App using Nodejs 5. Using JSON store structure and unstructured data	30
	Recommended Learning Resources	

Print Resources	1. Shama Hoque, “Full Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js”, Second Edition, 2020. 2. Eric Sarrion, “JavaScript from Frontend to Backend: Learn full stack JavaScript development using the MEVN stack with quick and easy steps” Packt, 2022.
Syllabus Design: Dr. Sukhvinder Singh, Assistant Professor, PUDoCS	

Year	IV	Course Code: CSCS410 Course Title: 5G Communication Technologies	Credits	4
			Hours	75
Sem.	VIII		Category	C

Course Prerequisites, if any	Basic knowledge of computers	
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.
Course Outcomes	• Understand the basics of 5G Communication • Understand the fundamentals of 5G Architecture • Understand the various 5G radio-access technologies • Understand the various 5G Enabling Technologies • Learn about the 5G use cases	

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Background** Introduction to Cellular Technologies: Frequency reuse – Handoff – Capacity – Evolution of 1G, 2G, 3G, 4G standards and architectures Propagation mechanisms: Doppler spread – Delay spread – Coherence time and bandwidth – all types of fading (non-detail study alone)	9
Unit II	**5G Architecture** Introduction – 5G Architecture options – 5G Core Network Architecture – 5G RAN Architecture – Network Slicing – 5G physical Layer – 5G Multiple Access Principle – Physical channels and signals – frame structure – Channel structures and beamforming basics – Random Access – Downlink and Uplink User Data transmission – Downlink and uplink signaling transmission – MIMI and beamforming operation – Channel coding – Dual connectivity – Data rates – Physical Layer measurements – UE capability	9
Unit III	**5G Radio Access Technologies** Access design principles for multi-user communications – Orthogonal multiple-access systems – Spread spectrum multiple access systems – Capacity limits of multiple-access methods – OFDM numerology for small-cell deployments – Radio access for dense deployments – Radio access for V2X communication	9
Unit IV	**5G Enabling Technologies** MIMO: Introduction – Single User and Multi user MIMO – Capacity of Massive MIMO – Resource allocation and transceiver algorithms – Channel models – mmWave – Channel Propagation – Hardware Technologies – Architecture and mobility – Beamforming – Physical layer techniques	9
Unit V	**5G Use Cases** Machine type communication: Fundamental techniques – Massive MTC – Ultra-reliable low-latency MTC Device to Device Communication: Radio resource management for mobile broadband D2D – Multi-hop D2D communications for proximity and emergency services – Multi-operator D2D communication	9
	Practical Component	

Exercisers	1. Study and simulation of Handoff techniques 2. Study and simulation of types of fading 3. Computation of channel capacity 4. Calculation of bandwidth of different generations 5. Problems based on 5G Frame Structure 6. 5G Communications Link Analysis with Ray Tracing 7. Model and analyze 5G NR Waveforms generation 8. Channel modelling in 5G 9. MIMO Wireless System Design for 5G 10. 5G Beamforming Design	30
Recommended Learning Resources		
Print Resources	1. Theodore S. Rappaport, "Wireless Communications: Principles and Practice", Cambridge University Press, 2024. [Unit 1] 2. Osseiran, Afif, Jose F. Monserrat, and Patrick Marsch, "5G Mobile and Wireless Communications Technology", First edition, Cambridge University Press, 2016. 3. Jonathan Rodriguez, "Fundamentals of 5G Mobile Networks", First Edition, Wiley, 2015. 4. Harri Holma, Antti Toskala, Takehiro Nakamura, "5G Technology 3GPP New Radio", First Edition, John Wiley & Sons, 2020.	
Syllabus Design: Dr. T. Chithralekha, Professor, PUDoCS		

Year	IV	Course Code: CSCS411 Course Title: Data Mining	Credits	4
Sem.	VIII		Hours	75
			Category	C
Course Prerequisites, if any	Database Management Systems			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Gain a comprehensive understanding of data mining concepts • Acquire knowledge in data preprocessing techniques • Gain knowledge in pattern mining • Attain knowledge and skills in classification • Understand various clustering algorithms			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Overview and History – Data Mining – Types of data – Kinds of Patterns – Technologies Used – Applications – Major Issues in Data Mining – Data Objects and Attribute Types – Basic Statistical Descriptions of Data	9
Unit II	**Data Preprocessing & Data Warehouse** Data Preprocessing Overview – Data Cleaning – Data Integration – Data Reduction – Data Transformation – Data Warehouse: Basic Concepts – Data Cube and OLAP – Data Generalization by Attribute-Oriented Induction	9
Unit III	**Pattern Mining** Pattern Mining Concepts – Market Basket Analysis – Frequent Itemsets – Closed Itemsets and Association Rules – Frequent Itemset Mining Methods – Pattern Evaluation Methods	9
Unit IV	**Classification** Fundamentals – Decision Tree Induction – Bayes Classification – Rule Based Classification – Model Evaluation and selection – Techniques to Improve Classification Accuracy	9
Unit V	**Clustering** Cluster Analysis – Partitioning methods – Hierarchical methods – Agglomerative, Divisive hierarchical clustering – DBSCAN – Evaluation	9
	Practical Component	
Exercises	1. Perform preprocessing for the given dataset 2. Program to Integrate two datasets with common attributes 3. Program to transform categorical data into numerical format for analysis 4. Program to create a basic data cube and perform OLAP operations 5. Implement the Apriori algorithm for mining frequent itemsets 6. Implement K-means clustering algorithm 7. Implement K-Medoids algorithm 8. Implement DBSCAN algorithm	30

Recommended Learning Resources	
Print Resources	1. Jiawei Hen, Micheline Kambler, Jian Pie, "Data Mining Concepts and Techniques", Morgan Kaufman, 2012. 2. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining", Pearson India Education Services Pvt. Ltd, 2016.

Syllabus Design: Dr. P. Shanthi Bala, Professor, PUDoCS

Year	IV	Course Code: CSCS412 Course Title: High Performance Computing	Credits	4
Sem.	VIII		Hours	75
			Category	C
Course Prerequisites, if any	Knowledge in Computer System Architecture and Operating Systems			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the historical context, structure, and broad impact of supercomputing • Grasp key features and enabling technologies shaping HPC systems • Apply parallel algorithms such as Fork-Join and Divide and Conquer in HPC systems • Analyze components such as Amdahl's Law and memory hierarchy in symmetric multiprocessor setups • Examine case studies like OpenMP API and OpenACC to understand their applications in HPC			
Unit No.	**Course Content**		**Hours**	
Theory Component				
Unit I	**Introduction** Introduction: High Performance Computing Discipline – History of Supercomputing – Anatomy of Supercomputer – Impact of Supercomputing on Science – Society and Security		9	
Unit II	**HPC Architecture** Key Properties of HPC Architecture – Enabling Technology – Vector and Pipelining – Single-Instruction – Multiple Data Architecture – Multiprocessors – Heterogeneous Computer Structures		9	
Unit III	**Parallel Algorithm** Introduction, Fork-Join – Divide and Conquer – Manger-Worker – Halo Exchange – Permutation: Cannon's Exchange – Task Dataflow: Breath First Search		9	
Unit IV	**Symmetric Multiprocessor Architecture** Amdahl's Law Plus – Processor Core Architecture – Memory Hierarchy – PCI Bus – External I/O Interfaces		9	
Unit V	**Case Studies** OpenMP API, Essential API, Open ACC		9	
Practical Component				
Exercises	1. Install MPICH library and write a "Hello World" program 2. Write a parallel program to calculate the value of PI/Area of Circle using OpenMP library 3. Write a parallel program to multiply two matrices using MPI library and compare the execution time with it's OpenMP and Serial version 4. Write a program in C to multiply two matrices of size 10000 x 10000 each and find it's execution-time using "time" command. Try to run this program on two or more machines having different configurations and compare execution-times obtained in each run. Comment on which factors affect the performance of the program 5. Install MPICH on two and more machines and create a MPI cluster. Execute MPI programs on this cluster and check the performance 6. Implement a program to demonstrate balancing workload on MPI platform		30	

Recommended Learning Resources	
Print Resources	1. Thomas Sterling, Matthew Anderson, Maciej Brodowicz, "High Performance Computing", Morgan Kaufmann, 2017. 2. Severance, Charles, and Kevin Dowd. "High performance computing", OpenStax CNX, 2015.
Syllabus Design: Dr. S.K.V. Jayakumar, Professor, PUDoCS	

Year	IV	Course Code: CSCS413	Credits	4
Sem.	VIII	Course Title: Cloud Computing	Category	C
Course Prerequisites if any	Knowledge of Distributed Systems and Databases			
Internal Assessment Marks: 40	End Semester Marks: 60		Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.	
Course Outcomes	• Understand cloud computing's evolution and characteristics • Comprehend the architecture of cloud computing and differentiate between Infrastructure as a Service (IaaS) and Software as a Service (SaaS) • Recognize PaaS features and examples • Compare scaling hardware using SLAs and billing principles • Evaluate cloud security measures			

Unit No.	Course Content	Hours
	Theory Component	
Unit I	**Introduction** Overview of Computing Paradigm – Recent trends in Computing – Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing – Evolution of cloud computing – Cloud Computing (NIST Model) Characteristics – Pros and Cons of Cloud Computing, Cloud computing vs. Cluster computing vs Grid computing – Role of Open Standards	9
Unit II	**Infrastructure as a Service (IaaS) & Software as a Service SaaS** Cloud Computing Architecture – Cloud computing stack – Service Models (XaaS) – Deployment Models. Infrastructure as a Service (IaaS) – Introduction – Virtualization, Hypervisors, Machine Image, Virtual Machine (VM) – Examples	9
Unit III	**Platform as a Service (PaaS)** Platform as a Service (PaaS) – Introduction – Cloud Platform and Management –Examples, Microsoft Azure, SalesForce.com – Software as a Service – Introduction – Web services – Web 2.0 – Web OS – Case Study on SaaS	9
Unit IV	**Service Management in Cloud Computing** Service Management in Cloud Computing – Service Level Agreements (SLAs) – Billing & Accounting – Comparing Scaling Hardware: Traditional vs. Cloud – Economics of scaling, Scalability & Cloud Services	9
Unit V	**Cloud Security** Cloud Security – Infrastructure Security – Data security and Storage – Data privacy and security Issues, Jurisdictional issues – Identity & Access Management – Access Control – Trust, Reputation, Risk	9
	Practical Component	
Exercises	1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs 3. Install Google App Engine. Create ‘hello world’ app and other simple web applications using python/java 4. Use GAE launcher to launch the web applications 5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim	30

	6. Find a procedure to transfer the files from one virtual machine to another virtual machine 7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version) 8. Install Hadoop single node cluster and run simple applications like word count	
Recommended Learning Resources		
Print Resources	1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", First Edition, Wiley, 2013. 2. Ronald L. Krutz, Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", First Edition, Wiley, 2010.	
Syllabus Design: Dr. M. Sathya, Assistant Professor, PUDoCS		

Year	IV	Course Code: CSCS414 Course Title: Deep Learning	Credits	4
			Hours	75
Sem.	VIII		Category	C
Course Prerequisites, if any	Machine Learning			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs. Duration of ESA (Practical): 03 hrs.		
Course Outcomes	• Understand the basic architecture and workings of neural networks • Understand the fundamentals of deep neural networks • Understand the architectures and workings of deep networks • Build the model for data variants using deep network • Build and train CNN and RNN deep learning architectures			

Unit No.	Course Content	Hours
	Theory Component	
UNIT I	**Foundations of Neural Networks and Deep Learning** Neural Networks – Training Neural Networks – Activation Functions – LossFunctions – Hyperparameters	9
UNIT II	**Fundamentals of Deep Networks** Defining Deep Learning – Common Architectural Principles of Deep Networks – Parameters – Layers – Activation Functions – Loss Functions – Optimization Algorithms – Hyperparameters – Building Blocks of Deep Networks – RBMs– Autoencoders – Variational Autoencoders	9
UNIT III	**Major Architectures of Deep Networks** Unsupervised Pretrained Networks – Convolutional Neural Networks (CNNs) – Architecture – Input, Convolutional, Pooling, fully connected Layers – Applications – Recurrent Neural Networks (RNN) – Modeling the Time Dimension – 3D Volumetric Input – Architecture – LSTM Networks	9
UNIT IV	**Building Deep Networks** Matching Deep Networks to the Right Problem – Modeling CSV Data with Multilayer Perceptron Networks – Modeling Handwritten Images Using CNNs – Modeling Sequence Data using RNN	9
UNIT V	**Tuning Deep Networks** Concepts: Matching Input Data and Network Architectures – Relating Model Goal and Output Layers – Working with Layer Count, Parameter Count, and Memory – Feed-Forward Multilayer Neural Networks – Controlling Layer and Parameter Counts – Weight Initialization Strategies – Using Activation Functions – Applying Loss Functions – Understanding Learning Rates – Applying Methods of Optimization – Controlling Epochs and Mini – Batch Size – Regularization – Max–Norm Regularization – Dropout– Dealing with Overfitting	9

Practical Component		
Exercises	1. Implement a simple perceptron model and train it to perform binary classification on a given dataset. Use the sigmoid activationfunction and gradient descent for training 2. Build a multilayer feed–forward neural network from scratch. Train the network using the backpropagation algorithm on a givendataset 3. Implement and train the CNN on the MNIST dataset for handwritten digit classification 4. Develop a RNN using Keras or PyTorch and train it to generate textbased on a given dataset **5.** Fine–tune a pre–trained CNN model using python	30
Recommended Learning References		
Print Resources	1. Josh Patterson and Adam Gibson, "Deep Learning – A Practitioner's Approach", O'Reilly Media, First Edition, 2017. **2.** Nikhil Buduma and Nicholas Locascio, "Fundamentals of Deep Learning: Designing Next Generation Machine Intelligence Algorithms", O'Reilly Media, Frist Edition, 2017.	
Syllabus Design: Dr. M. Nandhini, Professor, PUDoCS		

Multi-Disciplinary Course

<table>
<tr><td>Year</td><td>I / II</td><td colspan="2" rowspan="2">Course Code: COMS101
Course Title: Introduction to Python Programming</td><td>Credits</td><td>3</td></tr>
<tr><td></td><td></td><td></td><td>Hours</td><td>60</td></tr>
<tr><td>Sem.</td><td>I / III</td><td colspan="2"></td><td>Category</td><td>A</td></tr>
<tr><td>Course Prerequisites, if any</td><td colspan="5">Problem-solving skills</td></tr>
<tr><td>Internal Assessment Marks: 40</td><td colspan="2">End Semester Marks: 60</td><td colspan="3">Duration of ESA (Theory): 03 hrs.</td></tr>
<tr><td>Course Outcomes</td><td colspan="5">• Understand Python programming constructs
• Learn about different data structures in Python
• Write programs using functions
• Explore the use of Python modules and packages
• Perform Visualization using Python package</td></tr>
<tr><td>Unit No.</td><td colspan="3">Course Content</td><td colspan="2">Hours</td></tr>
<tr><td colspan="6">Theory Component</td></tr>
<tr><td>Unit I</td><td colspan="3">Introduction
Python Basics: Working – Identifiers – Comments – Types – Operations – Buit-in, library functions
Strings: Accessing – Properties – Operations
Control-flow Instructions: Decision Control – logical operators – conditional expressions Repetition control instruction – break and continue – pass Statement</td><td colspan="2">12</td></tr>
<tr><td>Unit II</td><td colspan="3">Console Input/Output
Console Input – Console Output – Formatted printing
Lists
Definition – Accessing – Operations – Methods – Varieties – Comprehension
Tuples
Definition – Accessing – Operations – Varieties – Comprehension – Conversion – Iterators and Iterables – zip()</td><td colspan="2">12</td></tr>
<tr><td>Unit III</td><td colspan="3">Sets
Definition – Accessing – Operations – Functions – Mathematical set operations – Updating set operations Dictionaries
Definition – Accessing – Operations – Functions – Nested Dictionary</td><td colspan="2">12</td></tr>
<tr><td>Unit IV</td><td colspan="3">Functions
Definition – Communication – Types – Unpacking – Lambda, Recursive functions
Modules and Packages
Creation and importing</td><td colspan="2">12</td></tr>
<tr><td>Unit V</td><td colspan="3">Exception handling
Syntax errors – handling exceptions – try-except – user-defined exceptions – else, finally blocks – Tips
Visualization - Matplotlib package – Plotting Graphs</td><td colspan="2">12</td></tr>
<tr><td colspan="6">Recommended Learning Resources</td></tr>
<tr><td>Print Resources</td><td colspan="5">1. Aditya Kanetkar, Yashavant Kanetkar, Let us Python, BPB Publisher, 6th Edition, 2023.</td></tr>
<tr><td colspan="6">Syllabus Design: Dr. R. Sunitha, Associate Professor, PUDoCS</td></tr>
</table>

Year	I	Course Code: COMS102 Course Title: Foundations of Information Technology	Credits	3
			Hours	60
Sem.	II		Category	A
Course Prerequisites, if any	Basic knowledge of Computers			
Internal Assessment Marks: 40	End Semester Marks: 60	Duration of ESA (Theory): 03 hrs.		
Course Outcomes	• Familiarize the fundamentals of Information Technology. • Understand the management of hardware and software • Describe the basics of networking • Discuss about data management and security aspects of data • Ability to troubleshoot computer systems			
Unit No.	**Course Content**		**Hours**	
Theory Components				
Unit I	**Introduction** Overview of IT – Computer Basics – Software fundamentals – Networks & Internet – IT ethics and policies		12	
Unit II	**Hardware and Software Management** Computer Assembly and maintenance - Operating Systems – Software installation and maintenance – Virtualization, Cloud Computing		12	
Unit III	**Networking Essentials** Network Fundamentals – Hardware – Protocols and services – Wireless Networking – Security		12	
Unit IV	**Data Management and Security** Data and fundamentals of Database – Data Backup and recovery – Cyber Security – Encryption and Cryptography		12	
Unit V	**IT Support and Troubleshooting** Help desk and IT support – Troubleshooting methodologies – Diagnostic tools and utilities – Future trends in IT		12	
Recommended Learning Resources				
Print Resources	1. Floyd Fuller, Brian Larson, Computers: Understanding Technology, EMC Paradigm, Fourth Edition, 2011. 2. Mike Meyers, CompTIA A+ Certification All-in-One Exam Guide, McGraw-Hill Education, Eleventh Edition, 2023. 3. Jeffrey S. Beasley, Piyasat Nilkaew, Networking Essentials, Prentice Hall Certification, Third Edition, 2012. 4. Charles J. Brooks, Christopher Grow, Philip Craig, and Donald Short , Cybersecurity Essentials, Sybex Publisher, First Edition, 2018.			
Syllabus Design: Dr. R.Sunitha, Associate Professor, PUDoCS				