ST. PHILOMENA'S COLLEGE (AUTONOMOUS)

Affiliated to University of Mysore
Accredited by NAAC with 'B ++ ' Grade
Bannimantap, Mysore, Karnataka,
India - 570015

REGULATIONS, CURRICULUM & SYLLABUS

FOR

MASTER OF COMPUTER APPLICATIONS (MCA)

Under Choice Based Credit System (CBCS)

Syllabus Document for the Academic Year 2024 - 2026 onwards

Duration: 2 Years

Program Level: Postgraduate

DEPARTMENT OF MCA

The Board of Studies in Master of Computer Applications which met on 18th October, 2024, has approved the syllabus and pattern of examination for Semesters I to IV for the Academic Year 2024 - 26.

SL. NO	Name of the Department	Percentage of syllabus revision carried out during the academic year 2024 - 25
1	Computer Applications (MCA)	10% Change
Note: 10% new addition		

BOS COMMITTEE MEMBERS

SL. NO	NAME	DESIGNATION & ADDRESS	REMARKS
1	Dr. Suresha	Professor, Department of Studies in Computer Science, University of Mysore, Manasagangotri , Mysore.	**University Nominee By VC** **Email** sureshasuvi@gmail.com **Phone Number** 9449810894
2	Dr. D. S Guru	Professor, Department of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore.	**External Member** **Email** dsguruji@yahoo.com dsg@compsci.uni-mysore.ac.in **Phone Number** 9620228005
3	Dr. R. Kumar	Head, Dept. of Computer Science (PG), Kristu Jayanti College (Autonomous), Bengaluru.	**External Member** **Email** rkumar@kristujayanti.com **Phone Number** 9902933747
4	Dr. Anil Kumar K.M	Professor, Dept. of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering (SJCE), JSS Science and Technology University, Mysore.	**External Member** **Email** anilkm@jssstuniv.in anilkm@sjce.ac.in **Phone Number** 9945633380
5	Mr.Vimal Chandran	Technical Lead, Wells Fargo, 129, Marathahalli - Sarjapur Outer Ring Rd, Devarabisanahalli, Bellandur, Bengaluru, Karnataka 560103	**Industry Expert** **Email** vimalchandran.cj@gmail.com **Phone Number** 8553279616
6	Mr. Noel Nitin Dsouza	Senior Project Engineer Wipro Limited, 146/147, Metagalli Industrial Area, Metagalli,Mysuru.	**Alumnus** **Email** noel.dsouza@wipro.com **Phone Number** 8722328244
7	Mrs. Vasantha Raghavan	**Associate Professor** MCA Department St. Philomena's College (Autonomous), Mysore	**Internal Member & Chairperson** **Email** vasantha_raghavan@stphilos.ac.in **Phone Number** 9900482145

SL. NO	NAME	DESIGNATION & ADDRESS	REMARKS
8	Ms. Bindushree K	**Assistant Professor** MCA Department St. Philomena's College (Autonomous), Mysore	**MCA Faculty** **Email** shreerbindu@stphilos.ac.in **Phone Number** 9008526223
9	Mr. Raghu Ram P S	**Head of MCA Department** MCA Department St. Philomena's College (Autonomous), Mysore	**MCA Faculty** **Email** psraghuram@stphilos.ac.in **Phone Number** 9889882328

Master

of

Computer Applications

2 Years / 4 Semesters

COURSE

CURRICULUM

REGULATIONS

2024

AS PER UGC GUIDELINES ON ADOPTION OF

CHOICE BASED CREDIT SYSTEM

1. **Course Title**

The course shall be called **'Master of Computer Applications Post-graduate Degree Course**'.

Duration: 2 YEARS

Number of Semesters: 4

A candidate joining the course shall pursue prescribed courses of studies.

2. **Definitions**

Course

Every course offered will have three components associated with the teaching-learning process namely

(i). Lecture – L (ii) Tutorial- T (iii) Practical - P, where

L stands for Lecture session.

T stands for Tutorial session consisting of participatory discussion / Self-study / desk work/ brief seminar presentations by students and such other novel methods that make a student absorb and assimilate more effectively the contents delivered in the Lecture classes.

P stands for Practical session and it consists of Hands-on experience / Laboratory Experiments / Field Studies / Case studies that equip students to acquire the much required skill component.

In terms of credits, every one hour session of L amounts to 1 credit per semester and a minimum of two hour sessions of T or P amounts to 1 credit per semester, over a period of one semester of 16 weeks for the teaching-learning process. The total duration of a semester is 20 weeks inclusive of semester-end examination.

A course shall have either or all the three components. That means a course may have only a lecture component, or only practical component or combination of any two or all the three components.

The total credits earned by a student at the end of the semester upon successfully completing the course are L + T + P. The credit pattern of the course is indicated as L: T: P

If a course is of 4 credits then the different credit distribution patterns in L: T: P format could be

4 : 0 : 0,	**1 : 2 : 1,**	**1 : 1 : 2,**	**1 : 0 : 3,**	**1 : 3 : 0,**
2 : 1 : 1,	**2 : 2 : 0,**	**2 : 0 : 2,**	**3 : 1 : 0,**	**3 : 0 : 1,**
0 : 2 : 2,	**0 : 4 : 0,**	**0 : 0 : 4,**	**0 : 1 : 3,**	**0 : 3 : 1**

The concerned BoS will choose the convenient credit pattern for every course based on the requirement. However, generally, a course shall be of 3 or 4 credits.

Different courses of study are labeled and defined as follows:

Core Course

A course which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.

A Core course may be a **Soft Core** if there is a choice or an option for the candidate to choose a course from a pool of courses from the main discipline / subject of study or from a sister / related discipline / subject which supports the main discipline / subject. In contrast to the phrase Soft Core, a compulsory core course is called a **Hard Core** Course.

Elective Course

Generally a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline / subject of study or which provides an extended scope or which enables an exposure to some other discipline / subject/domain or nurtures the candidate's proficiency/ skill is called an Elective Course. Elective courses may be offered by the main discipline/ subject of study or by sister / related discipline / subject of study. A Soft Core course may also be considered as an elective.

An elective course chosen generally from an unrelated discipline / subject, with an intention to seek exposure is called an **open elective**.

An elective course designed to acquire a special / advanced knowledge, such as supplement study / support study to a project work, and a candidate studies such a course on his own with an advisory support by a teacher is called a **Self -Study**.

A core course offered in a discipline / subject may be treated as an elective by other discipline / subject and vice versa.

Project work/Dissertation work is a special course involving application of knowledge in solving / analyzing /exploring a real life situation / difficult problem. A project work up to 4 credits is called Minor Project work. A project work of 6 to 8 credits is called Major Project Work. Dissertation work can be of 10-12 credits. A Project work/Dissertation work may be a hard core or a soft core as decided by the BoS concerned.

3. Eligibility for Admission

Candidates possessing a degree of University of Mysore, or of any other University, equivalent there to and complying with the eligibility criteria:

Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree.

OR

Passed B.Sc./ B.Com./ B.A. with Mathematics at 10+2 Level or at Graduation Level (with additional bridge Courses as per the norms of the concerned University).

Admission to MCA courses shall be open for candidates who have passed the Bachelor degree examinations with not less than 50% of the marks in the aggregate of all the years of the Degree examinations. However, in the case of candidates from Karnataka belonging to SC/ST and Category- I, the aggregate percentage of marks in the qualifying examinations shall not be less than 45%. Provided that for admission to MCA, the candidate shall have passed Bachelor Degree with not less than 50% of marks with Mathematics / Statistics / Computer Science / Computer Programming / Computer Application / Business Mathematics / Business Statistics as one of the optional or electives at degree level. Provided further that in respect of candidates who have studied and passed one of the subjects specified in the first provision in the Pre- university course with 50% of marks in that subject shall also be considered for admission.

However, in the case of candidates belonging to SC/ST and Category-I, 45% of marks in that subject shall also be considered for admission.

4. Scheme of Instructions

4.1. A Master's Degree program is of 4 semesters - two years duration. A candidate can avail double the duration (in one stretch) to complete Master's Degree (including blank semesters, if any). Whenever a candidate opts for blank semester(s)/DROP in a course or in courses or is compelled to DROP a course or courses as per the provision of the regulation, he/she has to study the prevailing courses offered by the department as per the prevailing scheme, when he/she continues his/her study.

4.2. A candidate has to earn a minimum of 76 credits, for successful completion of a Master's degree with a distribution of credits for different courses as given in the following table.

Course Type	Credits
Hard Core	40
Soft Core	A minimum of 32 but not exceeding 52
Open Elective	A minimum of 04

Every course including project work/Dissertation work, practical work, field work, seminar, self-study elective should be entitled as hard core or soft core or open elective by the BOS concerned.

4.3. A candidate can enroll for a maximum of 24 credits per semester with the approval of the concerned department.

4.4. Only such candidates who register for a minimum of 20 credits per semester in the first two semesters and complete successfully 76 credits in 4 successive semesters shall be considered for declaration of ranks, medals and are eligible to apply for student fellowship, Scholarship , free ships and hostel facilities.

4.5. In addition to the minimum of 76 credits for master's degree in the concerned discipline / subject of study, a candidate can opt to complete a minimum of 20 extra credits to acquire an **add-on proficiency diploma** in that particular discipline / subject along with the masters' degree. In such cases where in, a candidate opts to earn at least 4 extra credits in different disciplines / subjects in addition to a minimum of 76 credits at masters level, then an **add on proficiency certification** will be issued to the candidate by listing the courses studied and grades earned.

4.6. A candidate admitted to a Master's program can exercise an option to exit with a Bachelor Honors degree / PG diploma after earning 40 credits successfully.

5. Continuous Assessment, Earning of Credits and Award of Grades

The evaluation of the candidate shall be based on continuous assessment. The structure for evaluation is as follows:

5.1. The first component (C1), of assessment is for 15 marks. This will be based on tests, assignments and seminars. During the first half of the semester, the first 50% of the syllabus will be completed. This shall be consolidated during the 8th week of the semester. Beyond 8th week, making changes in C1 is not permitted.

5.2. The second component (C2), of assessment is for 15 marks. This will be based on tests, assignments, and seminars. The continuous assessment and scores of the second half of the semester will be consolidated during the 16th week of the semester. During the second half of the semester the remaining units in the course will be completed.

5.3. The outline for continuous assessment activities for Component-I (C1) and Component-II (C2) will be proposed by the teacher(s) concerned before the commencement of the semester and will be discussed and decided in the respective Departmental Council. The students should be informed about the modalities well in advance. The evaluated courses/assignments during component I (C1) and component II (C2) of assessment are immediately returned to the candidates after obtaining acknowledgement in the register maintained by the concerned teacher for this purpose.

5.4. During the 18th – 20th week of the semester, a semester-end examination of 3 hours duration shall be conducted for each course. This forms the third/final component of assessment (C3) and the maximum marks for the final component will be 70.

A candidate will be assessed on the basis of:

a. Knowledge of relevant processes
b. Skills and operations involved
c. Results / products including calculation and reporting.

Setting questions papers and evaluation of answer scripts.

I. Question papers in three sets shall be set for each course. one set by the internal examiner and two sets by External examiners. Whenever there are no sufficient internal examiners, the chairman of BoE shall get the questions papers set by external examiners.

II. The Board of Examiners shall scrutinize and approve the question papers.

III. **(i).** There shall be a single valuation for all theory papers. 40% by internal Examiner and 60% by external Examiner. In case, the number of internal examiners falls short, both can be external examiners.

(ii). The examination for Practical work/ Field work/Project work will be conducted jointly by two examiners , one internal & one external . However the BoE on its discretion can conduct with internal examiners in case if external examiner does not turn up. Also if a number of internal examiners are not available, Both can be external examiners at the discretion of BOE.

(iii). If a course is full of (L=0):T (P=0) type, then the examination for C3 Components will be as decided by the BOS concerned.

(iv). Challenge valuation

A student who desires to apply for challenge valuation shall obtain a Xerox copy of the answer script by paying the prescribed fee within 10 days after the announcement of the results. He / She can challenge the grade awarded to him/her by surrendering the grade card and by submitting an application along with the prescribed fee to the Registrar (Evaluation) within 15 days after the announcement of the results. This challenge valuation is only for C3 components.

The answer scripts for which challenge valuation is sought for shall be sent to another external examiner. The marks awarded will be the higher of the marks obtained in the challenge valuation and in maiden valuation.

5.5. In case of a course with only practical components a practical examination will be conducted with two examiners (ref: 5.4.III (ii)). A candidate will be assessed on the basis of a) knowledge of relevant processes b) Skills and operations involved c) Results / products including calculation and reporting. If the external examiner does not turn up then both the examiners will be internal examiners. The duration for semester-end practical examination shall be decided by the departmental council.

5.6. If **X** is the marks scored by the candidate out of 70 in C3 in theory examination, if **Y** is the marks scored by the candidate out of 70 in C3 in Practical examination, and if **Z** is the marks scored by the candidate out of 70 in C3 for a course of (L=0):T:(P=0)type that is entirely tutorial based course, then the final marks M in C3 is decided as per the following table.

L.T.P distribution	Find mark M in C_3
L:T:P	$\frac{[(L+T)*X]+[(T+P)*Y]}{L+2T+P}$
L:(T=0):P	$\frac{(L*X)+(P*Y)}{L+P}$
L:T:(P=0)	X
L:(T=0):(P=0)	X
(L=0):T :P	Y
(L=0): (T=0):P	Y
(L=0): T :(P=0)	Z

5.7. The details of continuous assessment are summarized in the following Table

Component	Syllabus Covered	Weightage	Assessment Period
C1	First 50% of the syllabus (2 units of total units)	15%	First half of the semester. Consolidated by the 8th week.
C2	Remaining 50% of the syllabus (remaining units of the course)	15%	Second half of the semester. Consolidated by the 16th week.
C3	Entire syllabus (all units)	70%	Semester-end examination to be conducted during the 18th-20th week.
Final grades announced by the 24th week.			

A candidate's performance from all 3 components will be in terms of scores, and the sum of all three scores will be for a maximum of 100 marks (15 + 15 + 70).

5.8. Finally awarding the grades should be completed latest by 24th week of the semester

5.9. Minor/ Major Project Evaluation

Right from the initial stage of defining the problem, the candidate has to submit the progress reports periodically and also present his/her progress in the form of seminars in addition to the regular discussion with the guide. Components of evaluation are as follows.

Component – I(C1): Periodic Progress and Progress Reports (15%) Component – II (C2): Results of Work and Draft Report (15%)

Component– III (C3): Final Viva-voce and evaluation (70%). The report evaluation is for 40% and the Viva-voce examination is for 30%

6. Declaration of Results

6.1. For a candidate to pass a course he/she should score a minimum of 40% from C1, C2 and C3 put together in that course provided he /she score a minimum of 30% (9 marks) in C1 and C2 put together and 30% (21 marks) in C3.

6.2. In case a candidate secures less than 30% in C1 and C2 put together in a course, the candidate is said to have DROPPED that course, and such a candidate is not allowed to appear for C3 in that course. In case a candidate's class attendance in a course is less than 75% or as stipulated by the University, the candidate is said to have DROPPED that course, and such a candidate is not allowed to appear for C3 in that course.

6.3. Teachers offering the courses will place the above details in the Department Council meeting during the last week of the semester, before the commencement of C3, and subsequently a notification pertaining to the above will be brought out by the Chairman of the Department before the commencement of C3 examination. A copy of this notification shall also be sent to the office of the Registrar & Registrar (Evaluation).

6.4. In case a candidate secures more than or equal to 30% in C3, but his/her grade (G) = 4, as per section 6.10 below, then he/she may be declared to have been conditionally successful in this course, provided that such a benefit of conditional clearance based on G=4 shall not be availed for more than 8 credits for the entire program of Master's Degree of two years.

6.5. In case a candidate secures less than 30% in C3, he/she may choose the DROP/MAKE-UP option. The candidate has to exercise his/her option to DROP immediately within 10 days from the date of notification of results.

6.6. A MAKE UP examination for odd semester courses will be conducted along with next regular odd semester examinations and for even semester courses along with the next regular even semester examinations. If a candidate is still unsuccessful, he/she may opt for DROP or again take up MAKE UP examination; however, not exceeding double the duration norm in one stretch from the date of joining the course.

6.7. A candidate has to re-register for the DROPPED course when the course is offered again by the department if it is a hard core course. The candidate may choose the same or an alternate core/elective in case the dropped course is a soft core / elective course. A candidate who is said to have DROPPED project work has to re-register for the same subsequently within the stipulated period. The details of any dropped course will not appear in the grade card.

6.8. The tentative / provisional grade card will be issued by the Registrar (Evaluation) at the end of every semester indicating the courses completed successfully. This statement will not contain the list of DROPPED courses.

6.9. Upon successful completion of Bachelors Honors / Master's degree a final grade card consisting of grades of all courses successfully completed by the candidate will be issued by the Registrar (Evaluation).

6.10. The grade and the grade point earned by the candidate in the subject will be as given below.

Marks	Grade	Grade Point (GP = V X G)
30 - 39	4	V * 4
40 - 49	5	V * 5
50 - 59	6	V * 6
60 - 64	6.5	V * 6.5
65 - 69	7	V * 7
70 - 74	7.5	V * 7.5
75 - 79	8	V * 8
80 - 84	8.5	V * 8.5
85 - 89	9	V * 9
90 - 94	9.5	V * 9.5
95 - 100	10	V * 10

Here, P represents the percentage of marks (P = [(C1 + C2) + M]) secured by a candidate in a course, which is rounded to the nearest integer. V is the credit value of the course, G is the grade, and GP is the grade point.

6.11. A candidate can withdraw from any course within ten days of the notification of final results. If the course is a hard core course, the candidate must re-register for the same course. For a soft core or open elective, the candidate may choose to re-register for the same course or an alternate course.

6.12. A DROPPED course is automatically considered as a withdrawn course. The Overall Cumulative Grade Point Average (CGPA) of a candidate after successfully completing the required 76 credits is calculated as:

$$\textbf{CGPA} = \Sigma \textbf{GP} / \textbf{Total number of credits}$$

7. Classification of Results

The Final Grade Point (FGP) awarded to the student is based on the Cumulative Grade Point Average (CGPA) secured by the candidate and is classified as follows:

CGPA Range	FGP	
	Numerical Index	Qualitative Index
4 ≤ CGPA < 5	5	SECOND CLASS
5 ≤ CGPA < 6	6	
6 ≤ CGPA < 7	7	FIRST CLASS
7 ≤ CGPA < 8	8	
8 ≤ CGPA < 9	9	DISTINCTION
9 ≤ CGPA <= 10	10	

Overall Percentage: The overall percentage is calculated as 10 × CGPA or is considered to be 50% in the case of CGPA < 5.

8. Attendance and Conduct

The course is a full-time program, and students **SHALL NOT** engage in any part-time or full-time employment or enrollment in other courses during their study. Violation of this rule will result in the student being removed from the course. A minimum attendance of 75% of actual working hours is required for each course. Students who fail to meet the requirements for attendance and conduct will not be permitted to appear for the examination in the concerned subject.

9. Medium of Instruction

The medium of instruction for the program shall be English. However, candidates are permitted to write the examinations in either English or Kannada. This rule does not apply to language courses.

10. **Provision for Appeal**

If a candidate is dissatisfied with the evaluation of C1 and C2 components, he/she may approach the Grievance Cell with a written submission, providing all relevant facts, including the evaluated assignments, test papers, etc. The appeal must be submitted before the commencement of the semester-end examination.

The Grievance Cell is empowered to:

1. Revise the marks if the appeal is found to be genuine.
2. Impose a penalty on the candidate if the submission is deemed baseless or unduly motivated.
3. Recommend disciplinary or corrective action against an evaluator if found guilty of misconduct.

The decision of the Grievance Cell is final.

Composition of the Grievance Cell:

1. Registrar (Evaluation) – Ex-officio Chairman/Convener.
2. One senior faculty member (not involved in the evaluation of the concerned course) from the department or a sister department/discipline.
3. One senior faculty member/subject expert from outside the University department.

MCA Program Outcome (PO)

- Upon successful completion of the MCA degree, graduates will be able to:
- *Apply knowledge from Computer Science, Mathematics, Statistics, and computing fundamentals to design and develop innovative solutions for real-world applications.*
- *Integrate and apply modern IT tools effectively, designing applications with due consideration for societal and environmental needs.*
- *Engage in lifelong learning to ensure continued career development while adhering to professional ethics, cultural values, and cyber regulations.*
- *Work effectively both as a team leader and a team member in multidisciplinary projects, demonstrating strong computing, management, and communication skills in both written and oral formats.*
- *Utilize entrepreneurial skills to successfully venture into entrepreneurial roles, with a focus on innovation and sustainability.*

Program Specific Outcome (PSO)

- *Understand and apply computing concepts in fields such as Web design and development, Mobile application development, and Network and communication technologies.*
- *Develop real-world applications by applying knowledge from coursework to solve practical problems.*
- *Leverage modern design and development tools to analyze, design, and implement solutions for a variety of technical applications.*
- *Communicate professionally, both orally and in writing, demonstrating ethical practice and a commitment to social welfare*

ST. PHILOMENA'S COLLEGE (AUTONOMOUS), MYSORE

MASTER OF COMPUTER APPLICATIONS (MCA)

CHOICE BASED CREDIT SYSTEM (CBCS) 2024 - 26

MCA Course Structure and Syllabi

The Minimum Credits Required for the MCA Degree are outlined as part of the Choice Based Credit System (CBCS) for the academic year 2024 - 26.

I to IV	Hard Core Course (HC)		Soft Core Course (SC)		Open Elective Course (OE)		Total	
Semesters	Numbers	Credits	Numbers	Credits	Numbers	Credits	Numbers	Credits
	10	40	8	32	1	4	19	76

Credit Allotment Semester Wise

Semester	Minimum Credit	Maximum Credit
I	20	24
II	24	24
III	20	24
IV	12	24
TOTAL	**76**	**96**

SCHEME AND SYLLABUS

SEMESTER I

LIST OF HARD CORE SUBJECTS

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Data Structure and Problem Solving Using C	3 : 0 : 0	3	3	
2	Operating Systems and Linux	3 : 1 : 0	4	4	
3	Computer Networks	3 : 0 : 0	3	3	
4	Computer Organization and Architecture	3 : 1 : 0	4	4	
5	Data Structure and Problem Solving Using C Laboratory	0 : 0 : 1	1	2	
6	Computer Networks Laboratory	0 : 0 : 1	1	2	

LIST OF SOFT CORE SUBJECTS (At Most Two)

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Mathematical Foundations for Computer Science	3 : 1 : 0	4	4	
2	Business Systems	3 : 1 : 0	4	4	
3	Data Mining and Data Warehousing	3 : 0 : 1	4	4	
4	Probability and Statistics	3 : 1 : 0	4	4	

SEMESTER II

LIST OF HARD CORE SUBJECTS

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Java Programming	2 : 1 : 0	3	4	
2	Advanced Database Management Systems	3 : 0 : 0	3	4	
3	Java Programming Laboratory	0 : 0 : 1	1	2	
4	Advanced Database Management Systems Laboratory	0 : 0 : 1	1	2	

LIST OF SOFT CORE SUBJECTS (At Most Three)

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Analysis and Design of Algorithms	2 : 1 : 1	4	4	
2	Big Data Analytics	3 : 1 : 0	4	4	
3	Foundation of Cyber Security	3 : 1 : 0	4	4	
4	Introduction to Cloud Computing	3 : 1 : 0	4	4	
5	Artificial Intelligence and Robotics	3 : 1 : 0	4	4	
6	Web Technologies	3 : 1 : 0	4	4	

OPEN ELECTIVE (ONLY ONE)

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Offered by Non - Computer Science Department	4 : 0 : 0	4	4	

SEMESTER III

LIST OF HARD CORE SUBJECTS

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Software Engineering	3 : 1 : 0	4	4	
2	Python Programming	3 : 0 : 0	3	4	
3	Python Programming Laboratory	0 : 0 : 1	1	2	

LIST OF SOFT CORE SUBJECTS (At Most Four)

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Machine Learning	2 : 1 : 1	4	4	
2	Cloud Computing Architecture	3 : 1 : 0	4	4	
3	Machine Learning in Cyber Security	3 : 1 : 0	4	4	
4	Data Science	3 : 1 : 0	4	4	
5	Digital Image Processing	3 : 0 : 1	4	4	
6	Internet of Things	3 : 1 : 0	4	4	
7	Mobile Application Development	2 : 1 : 1	4	4	
8	Ethical Hacking	1 : 1 : 2	4	4	

SEMESTER IV

LIST OF HARD CORE SUBJECTS

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Major Project	0 : 2 : 6	8	16	

LIST OF SOFT CORE SUBJECTS (At Most Four)

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Research Methodology	3 : 1 : 0	4	4	
2	BlockChain Technology	3 : 1 : 0	4	4	
3	Theory of Languages and Automata	3 : 1 : 0	4	4	
4	Cloud Security	3 : 1 : 0	4	4	
5	Software Testing	3 : 0 : 1	4	4	
6	E-Commerce and E-Governance	3 : 1 : 0	4	4	
7	Communication Skills and Profession Management	3 : 1 : 0	4	4	
8	Soft Computing	3 : 1 : 0	4	4	

LIST OF OPEN ELECTIVE SUBJECTS OFFERED FROM THE DEPARTMENT

SL. NO	COURSE TITLE	L - T - P	CREDITS	CONTACT HOURS	COURSE CODE
1	Python Programming	3 : 1 : 0	4	4	
2	Artificial Intelligence	3 : 1 : 0	4	4	
3	Web Technologies	3 : 1 : 0	4	4	

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Data Structure and Problem Solving Using C</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">48</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">03</td><td>Credits</td><td>03</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of programming concepts (preferably in C).
2. Familiarity with basic mathematical concepts, such as logic and set theory.
3. Introductory knowledge of computer systems and memory management.

COURSE OBJECTIVE: This course aims to enable learners:

1. Understand the foundational concepts of C programming and its application in problem-solving.
2. Learn the core data structures such as arrays, stacks, queues, and linked lists and their implementations.
3. Develop efficient algorithms for sorting, searching, and other operations using C.
4. Apply dynamic memory allocation and recursion to solve complex problems.
5. Explore fundamental tree structures and their applications in problem-solving.
6. Gain the ability to analyze and apply data structures in real-world scenarios, focusing on efficiency and optimization.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Demonstrate proficiency in C programming concepts such as arrays, pointers, recursion, and dynamic memory allocation for solving computational problems.
CO2	Implement and apply abstract data types (ADT) such as arrays, strings, stacks, and queues to solve algorithmic problems.
CO3	Analyze and implement stack and queue operations, and understand their applications in expression evaluation, recursion, and algorithmic design.
CO4	Understand and implement linked lists (single, circular, and doubly linked lists) and apply them to practical problems involving dynamic data structures.
CO5	Implement binary trees, binary search trees, and graph data structures, and apply traversal, searching, and sorting techniques to real-world problems.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction and Overview of C Programming** Introduction to C programming, Variables, Data types, Constants, Declarations, Operators, Precedence, Associativity, Order of evaluation. Input and output statements; Control Statements, Arrays – Single dimension, Two-dimensional, Multi-dimensional Arrays, Strings. Functions, Categories of functions. Examples: Pointers, Pointer arithmetic, Call by value, Pointer Expression, Pointer as function arguments, recursion, passing strings to functions, Call by reference, Functions returning pointers, Pointers to functions, Programming Examples. Structures and Unions.	10
2	**UNIT II - Introduction to Data Structures** Abstract Data Types (ADT), Sequences as Value Definitions, ADT for Varying-Length Character Strings, Pointers and Dynamic Memory Allocation (malloc, calloc, realloc, free). Array as ADT, Arrays as Parameters, String as ADT	10
3	**UNIT III - The Stack & Queues** **Stacks:** Definition and Examples, Primitive Operations, Stack as ADT, Representing Stacks, Implementing pop and push. Infix, Postfix, and Prefix Expressions. Applications of Stacks: Expression Evaluation and Conversion, Recursion, Binary Search, Towers of Hanoi **Queues:** Sequential Representation, Queue as ADT, Priority Queue, Circular Queue, Double Ended Queue.	10
4	**UNIT IV - Linked Lists** Linked Lists: Inserting/Removing Nodes, Linked Implementations of Stacks and Queues, Circular Lists, Doubly Linked Lists, Applications of Linked Lists (Stacks, Queues, Deques).	10
5	**UNIT V - Trees** **Trees:** Binary Tree, Properties of binary tree, Representation of binary tree, Common Binary Tree operations. Binary Search Tree-Definitions, ADT, Binary search Tree operations and implementation.	8

Reference Books

1. “Programming in ANSI C”, Third Edition, E. Balaguruswamy. 6th Edition (2013).
2. “The Complete Reference C”, Herbert Schildt, Fifth Edition, Tata McGraw Hill.
3. “Problem Solving With C”, M T Somashekara, Eastern Economy Edition.
4. “Data Structures Using C”, Reema Thareja, Oxford University Press, 2nd Edition (2014).
5. “Fundamentals of Data Structures in C”, Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, Universities Press, 2nd Edition (2008).

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Operating System and Linux</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic structure of the operating system.
2. To acquire the knowledge of processes and threads.
3. To develop skills to manage memory and other resources.
4. To understand the structure of files and directories and to understand the various storage techniques.
5. To understand the various security issues and to acquire the knowledge of Linux operating systems.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Recognize the structure of the operating system, interaction of an operating system and application programs.
CO2	Analyze the various programming paradigms viz. multi-process and multi-threaded programming.
CO3	Examine the various resources and memory management techniques.
CO4	Examine the file system and various storage techniques .
CO5	Identify current issues in system security; demonstrate various factors can influence the overall performance of an operating system.

Course Content

Unit No	Content	Hours
1	**UNIT-I-Computer and Operating Systems Structure:** Basic Elements, Processor Registers, Instruction Execution, The Memory Hierarchy, Cache Memory, I/O Communication Techniques, Introduction to Operating System, Mainframe Systems, Desktop Systems, Multiprocessor Systems, Distributed Systems, Clustered Systems, Real - Time Systems, Handheld Systems, Feature Migration, Computing Environments. System Structures: System Components, Operating–System Services, System Calls, System Programs, System Structure, Virtual Machines, System Design and Implementation, System Generation	12
2	**UNIT-II-Process Management and Mutual Execution:** Process, Process States, Process Description, Process Control, Execution of the Operating System, Security Issues, Processes and Threads, Symmetric Multiprocessing(SMP), Micro kernels, CPU Scheduler and Scheduling. Principles of Concurrency, Mutual Exclusion: Hardware Support.	10
3	**UNIT-III- Deadlock and Memory Management:** Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategy, Dining Philosophers Problem Memory Management: Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing	10
4	**UNIT-IV-File System and Secondary Storage:** File Concept, Access Methods, Directory Structure, File System Mounting, File Sharing, Protection, File – System Structure, File – System Implementation, Directory Implementation, Allocation Methods, Free–Space Management**,** Disk Structure, Disk Scheduling, Disk Management.	10
5	**UNIT-V- Computer Security and Shell Programming:** The Security Problem, User Authentication, Program Threats, System Threats. Linux System Linux history, Design Principles, Kernel modules, Process, management, scheduling, Memory management, File systems, Input and output, Inter-process communications. Shell Programming: Introduction, Shell as a Programming Language, Types of shells, shell syntax, pipes and Redirection, Environment variables, working with files.	10

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 8th edition, Wiley India, 2009.
2. William Stallings, “Operating System Internals and Design Principles” Pearson, 6th edition, 2012
3. Daniel P.Bovet & Macro Cesati(2006). Understanding Linux Kernel (3rd edition), O’reily Series
4. MichaelBeck, Harald Bohme, Robert Magnus, DirkVerwoner.(2002).LinuxKernelProgramming Pearson Education Ltd.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Computer Networks</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">48</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">03</td><td>Credits</td><td>03</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Understanding of Computer Hardware and Software
2. Familiarity with Operating Systems
3. Problem Solving and Analytical skills
4. Basic programming Skills
5. Familiarity with mathematical concepts, such as logic , Different number systems

COURSE OBJECTIVE: This course aims to enable learners:

1. To familiarize the fundamental concepts of computer networking models.
2. To understand different components of computer networks and various protocols with their applications
3. To understand basics of Cryptography and methods to secure a message over insecure channels by various classical encryption techniques.
4. To learn about modern cryptographic techniques
5. such as DES, AES and Secure Hash Algorithms(SHA)
6. Analyze various symmetric ciphers and data integrity algorithms.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Explore the basic concepts, principles and techniques of Data Communication along with the layers of OSI and TCP / IP models.
CO2	Discuss various error detection and correction techniques in data link layer and analyze the performance of network layer
CO3	Demonstrate the services of transport layer and the basic concepts of Application-Layer Paradigms.
CO4	Explore the fundamental concepts of Cryptography by applying various Encryption techniques.
CO5	Analyze various symmetric ciphers and data integrity algorithms.

Unit No	Course Content	Hours
1	**UNIT-I: Basics of Data Communications and Physical Layer** Data Communications , Components, Data Representation, Data Flow Networks; Network Criteria, Physical Structures. Network Types: LAN, WAN, Switching. Network Models: Protocol Layering: Scenarios , Principles of Protocol Layering, Logical Connections. TCP/IP Protocol Suite: Layered Architecture, Layers in the TCP/IP Protocol Suite, Description of each layer, Encapsulation and Decapsulation , Addressing, Multiplexing and De- multiplexing, The OSI Model. Introduction to Physical Layer: Data and Signals: Analog and Digital data , Analog and Digital Signals ,Periodic and Non Periodic. Periodic Analog Signals: Sine wave , Phase , Wavelength , Time and frequency domains , Bandwidth , Digital Signals : Bit Rate, Bit length. Transmission Impairment, Data Rate Limits, Performance	10
2	**UNIT-II- Data Link Layer and Network Layer** Introduction to Data-Link Layer, Link-Layer Addressing: Address Resolution Protocol (ARP). Error Detection and Correction: Introduction, Types of Errors, Redundancy, Detection versus Correction, Coding, Block coding , Cyclic Codes : Cyclic redundancy check. Introduction to Network Layer, Network-Layer Services: Packetizing, Routing and Forwarding, Other services, Packet Switching: Datagram Approach, Virtual-Circuit Approach, Network Layer performance, Routing Algorithm: Distance Vector Routing , Congestion Control Algorithm : Leaky bucket algorithm	10
3	**UNIT-III-Transport layer and Application layer** Introduction to Transport-Layer: Transport- Layer Services.Transport-Layer Protocols: Simple Protocol, FSMs, Stop & wait Protocol, FSMs, Go Back N Protocol and Selective repeat Protocol. Introduction to Application Layer, Providing Services, Application-Layer Paradigms, , Client Server Programming. World Wide Web and HTTP: FTP: Two Connections, Control Connection, Data Connection, Domain Name System (DNS): Name Space, DNS on the Internet, Resolution. Self-Learning Component: Caching, Resource Records, DNS Messages, Registrars , DDNS, Security of DNS.	10
4	**UNIT-IV-Introduction to Cryptography** Security Goals , Attacks, Services & techniques , Security Mechanisms, Model for Network Security, Standards, Classical Encryption techniques, Traditional block cipher structure, DES, DES Example.	10
5	**UNIT-V- Asymmetric Ciphers and Data Integrity Algorithms** Principles of Public-key Cryptosystems, RSA Algorithms, Diffie-Hellman Key exchange, Applications of Cryptographic hash functions, two simple hash functions, Hash functions based on Cipher block chaining, Secure hash Algorithm(SHA), Message Authentication Requirement, Message authentication Functions, Message authentication codes.	8

Reference Books:

1. B. A. Forouzan, “Data Communications and Networking”, 5th Edition, McGraw Hill Education (India) Private Limited, 2016.
2. William Stallings, “Cryptography and Network Security”, 7th Edition, Pearson, 2018
3. William Stallings, Data and Computer Communications, 10th Edition, Pearson, 2015.
4. Behrouz A. Forouzan, “Cryptography and Network Security”, Tata McGraw-Hill Publishing, 2010.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Computer Organization and Architecture</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE:

1. Problem Solving and Analytical skills
2. Familiarity with Fundamental mathematical concepts, such as logic , Different number systems etc.

COURSE OBJECTIVE: This course aims to enable learners

1. To understand the basic structure and operation of a digital computer.
2. Study different number system representation and conversion from one number system to another and Boolean algebra.
3. To Learn the working of flip-flops, logical gates, multiplexers, and adders.
4. Study the memory system, cache memories and virtual memory.
5. Explain the different ways of communicating with I/O devices and standard I/O interfaces.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the basics of Combinational logic.
CO2	Realize the concept of Computer System Organization
CO3	Knowledge about Assembly Language & concepts of Input/output Organization.
CO4	Implement Assembly language programming concepts
CO5	Analyze the Performance of Memory System and Memory Management . Analyze the performance benchmarks .

Unit No	Course Content	Hours
1	**UNIT- I -Binary Systems and Combinational Logic** Digital Computers and Digital Systems, Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, subtraction using r's and r-1 complements, Binary Code, Binary Storage and Registers, Binary Logic, Integrated Circuits. Axiomatic Definition of Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Canonical and Standard Forms, Other Logic Operations, Digital Logic Gates, The map Method, Two – and Three – Variable Maps, Four – Variables Map.	12
2	**UNIT- II-Arithmetic Circuits and Sequential Logic:** NAND and NOR Implementation, Other Two- Level Implementations, Don't Care Conditions. Introduction, Adders, Subtractors, Binary Parallel Adder, Decimal Adder, Magnitude Comparator, Decoders, Multiplexers, BOOTH algorithm for signed numbers with example. Sequential Logic: Introduction, different types of Flip – Flops, Triggering of Flip- Flops.	10
3	**UNIT- III -Assembly Language and Input /Output Organization:** Computer types , functional units , basic operational concepts , Bus structure , software , Performance. Multiprocessing and Multi computers, Machine Instruction: Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.	10
4	**UNIT- IV- Assembly Language Programming & Accessing I/O Devices** Basics of Assembly Language Program, Examples from Assembly Language Programming. Accessing I/O Devices, Interrupts, Enabling & Disabling Interrupts, Handling Multiple devices.	10
5	**UNIT- V- The Memory System and Performance Evaluation** Basic Concepts, Semiconductor RAM Memories, Internal organization of memory chips, static memories, dynamic RAM, Synchronous D-RAM, Structure of larger Memories. Read – Only Memories, Speed, Size, and Cost, Cache Memories, Virtual Memories, Memory Management Requirements. Performance evaluation-SPEC marks, Transaction Processing benchmarks.	10

Reference Books

1. M.Morris Mano, "Digital Logic and Computer Design", Pearson, 2012.
2. CarlHamacher, Zvonko Vranesic Safwat Zaky, "Computer Organization", 5th edition, TataMcGraw-Hill, 2011
3. Soumitrs Kumar Mandal,"Digital Electronics Principles and Applications", Tata McGraw-Hill,2010

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Data Structure and Problem Solving Using C Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">01</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of C programming concepts such as variables, data types, loops, and functions.
2. Familiarity with fundamental problem-solving techniques.
3. Knowledge of basic mathematics, including logic and set theory.

COURSE OBJECTIVE: This course aims to enable learners:

1. Provide hands-on experience in implementing core data structures such as arrays, stacks, queues, linked lists, trees, and heaps using C programming.
2. Enable students to develop problem-solving skills by applying data structures to real-world applications.
3. Strengthen understanding of dynamic memory allocation and recursion through practical implementation.
4. Implement efficient algorithms for searching, sorting, and expression evaluation in C.
5. Familiarize students with advanced topics like binary trees, heaps, and searching techniques, preparing them for complex computational problems.

COURSE OUTCOME (CO)

COURSE OUTCOMES	
CO1	Implement fundamental C programming concepts such as arrays, functions, pointers, and dynamic memory allocation in solving computational problems.
CO2	Develop and implement various data structures such as stacks, queues, and linked lists to solve algorithmic challenges.
CO3	Apply sorting, searching, and tree traversal techniques to solve complex data manipulation problems.
CO4	Analyze and solve real-world problems using binary trees, heaps, and recursion.
CO5	Evaluate and implement efficient algorithms for expression evaluation, including infix to postfix conversion and postfix evaluation.

Course Content

1. Program to calculate the mean, median, LCM, GCD, minimum, and maximum of a set of numbers.
2. Program to calculate the salary of an employee given basic pay, HRA, TA, and IT deductions.
3. Program to solve a quadratic equation and find its roots.
4. Program to calculate the average marks of students using arrays and structures.
5. Program to perform matrix operations (addition and multiplication).
6. Program to implement dynamic memory allocation using `malloc`, `calloc`, `realloc`, and `free`.
7. Program to evaluate the validity of a mathematical expression (balanced parentheses).
8. Program to evaluate a postfix expression.
9. Program to convert an infix expression to postfix.
10. Program to implement multiple stacks of integers.
11. Program to perform basic operations on a queue of integers (enqueue, dequeue, handling overflow/underflow).
12. Program to implement a circular queue for storing student information (registration number, course title, year).
13. Program to implement a double-ended queue (deque) with insertion and deletion at both ends.
14. Program to implement stack operations using a linked list (push, pop, display).
15. Program to implement queue operations using a linked list (enqueue, dequeue, display).
16. Program to create a student mark list based on rank, with student-id, name, and total marks.
17. Program to perform the following operations on a singly linked list:
    a. Insertion (at front, rear, and based on position).
    b. Deletion (at front, rear, and based on position).
    c. Display, replace, and swap nodes.
18. Program to perform the following operations on a doubly linked list:
    a. Insertion (at beginning, end, and in between).
    b. Deletion (at beginning, end, and in between).
    c. Display and swap nodes.
19. Program to represent a binary tree using an array and perform the following operations:
    a. Print left and right child of a specified node.
    b. Print ancestors, nodes at specific levels, and leaf nodes.
20. Program to represent a binary tree using a linked list and perform the same operations as the array-based tree.

21. Program to traverse a binary tree using recursive routines:
    a. Pre-order traversal, In-order traversal and Post-order traversal.
22. Program to construct a heap and sort it using heap sort.
23. Program to perform linear search on a dataset.
24. Program to perform binary search on a sorted array of integers.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Computer Networks Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">01</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Understanding of Computer Hardware and Software
2. Familiarity with Operating Systems
3. Problem Solving and Analytical skills
4. Basic programming Skills
5. Familiarity with mathematical concepts, such as logic , Different number systems

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the working principle of various communication protocols.
2. To analyze the various routing algorithms
3. To know the concept of data transfer between nodes

COURSE OUTCOME (CO)

COURSE OUTCOMES	
CO1	To understand the working principle of various communication protocols.
CO2	To analyze the various routing algorithms
CO3	To know the concept of data transfer between nodes

LIST OF LAB PROGRAMS

PART A

1. Write a program for a distance vector algorithm to find a suitable path for transmission.
2. Using TCP/IP sockets, write a client-server program to make the client send the file name and to make the server send back the contents of the requested file if present.
3. Write a program for Hamming code generation for error detection and correction.
4. Write a Program for error detection using CRC
5. Write a program for congestion control using a leaky bucket algorithm.
6. Implement RSA algorithm to illustrate Encryption & Decryption.

PART B

Introduction to NS2

1. Simulate a three node point — to — point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.
2. Simulate the network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3(a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds
3. Simulate to study transmission of packets over Ethernet LAN and determine the number of packets drop at destination
4. Simulate the different types of internet traffic such as FTP and TELNET over a wired network and analyze the packet drop and packet delivery ratio in the network.
5. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

   Note : Students may be asked to simulate other protocols and analyze the working.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Mathematical Foundations for Computer Science</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. To analyze statistical data through different techniques
2. To understand various types of errors in numerical computations and different operations on matrices.
3. To study the different techniques to solve linear equations.
4. To use the properties of relations and functions in real life problems.
5. To understand the applications of graphs and trees in solving problems.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Implement statistical measures and explore its applications
CO2	Analysis of computational errors and design of algorithms to solve a set of linear equations.
CO3	Applying the concepts of vector and linear functions in real time applications.
CO4	Apply the notion of relations on finite structures, like strings and analyze algorithms using the concept of functions.
CO5	Explore the properties of Graph theory and its applications in computer science.

Course Content

Unit No	Content	Hours
1	**UNIT- I-Statistics** Univariate data – different measures of location, dispersion, relative dispersion, skewness and kurtosis, Moments, Measures based on them – comparison with moment measures, Correlation and Regression Analysis.	12
2	**UNIT- II-Number Systems and Vector & Matrix Algebra** Errors in Numerical Computations, Types of Errors, Analysis and Estimation of Errors, Vector Algebra: Vector spaces with real field, Basis and dimension of a vector space, Orthogonal vectors, Properties of Matrices and Determinants: Matrix Operations, Elementary Matrices, Inverse Matrix, Diagonal Matrix, Symmetric Matrix, and Determinant Matrix.	10
3	**UNIT- III-Linear Algebraic Systems** Numerical methods for Linear Systems, Direct Methods for Linear Systems: Cramer's Rule, Gauss Elimination Method, Gauss Jordan Elimination Method, Pivoting Strategies, Gauss- Jordon Method, LU Decomposition Method, Tridiagonal Systems of Linear Equations, Iterative Methods for Solving Linear Systems, Jacobis Iteration Method, Gauss-Seidel Iterative Method, Convergence Criteria, EigenValues and EigenVectors.	10
4	**UNIT- IV-Relations and Functions** Cartesian products and Relations, Properties of Relations, Functions: Plain and One-to-One, Onto Functions: Stirling Numbers and the Second Kind, Special functions, The Pigeon-hole principle, Function composition and inverse functions.	10
5	**UNIT- V-Graph Theory** Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree: Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles, Graph Coloring and Chromatic Polynomials.	10

Reference Books

1. Sant Sharan Mishra, “Computer Oriented Numerical and Statistical Methods”, PHI Learning Private Limited, 2013.
2. Rizwan Butt, “Introduction to Numerical Analysis Using Matlab”, Infinity Science Press LLC, 2008
3. Ralph P Grimaldi, B.V.Ramana, “Discrete & Combinatorial Mathematics, An Applied Introduction” 5th Edition, Pearson Education, 2009.
4. D.S. Chandrasekharaiah, Discrete Mathematical Structures, 4th Edition, PRISM Pvt. Ltd. 2012.
5. Bondy and U.S.R.Murty: Graph Theory and Applications (Freely downloadable from Bondy's website; Google-Bondy)
6. S. Kumarsean, “Linear Algebra A geometric approach”, Prentice Hall of India Private Limited, 2001
7. Kenneth H Rosen, “Discrete Mathematics & its Applications" 7th edition, McGraw- Hill, 2010.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Business Systems</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of information technology and business principles.
2. Familiarity with management practices and organizational structures.
3. Introductory knowledge of e-business systems (Recommended).

COURSE OBJECTIVE: This course aims to enable learners:

1. Provide an in-depth understanding of the role of information systems (IS) in modern businesses.
2. Explore the integration of information technology (IT) with business strategies and processes.
3. Enable students to develop IT-driven solutions for business challenges and opportunities.
4. Introduce key concepts in e-business and the integration of business processes through information systems.
5. Equip students with skills to analyze and design business/IT strategies for improving business operations.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the fundamental role of information systems in enhancing business operations and decision-making.
CO2	Analyze the managerial challenges associated with implementing information technology in business environments.
CO3	Apply strategic planning techniques such as SWOT analysis and business/IT architecture planning to align IT solutions with business goals.
CO4	Design and evaluate cross-functional enterprise applications, including transaction processing systems and enterprise collaboration systems.
CO5	Explore the role of e-business systems in areas such as marketing, human resources, accounting, and financial management.

Course Content

Unit No	Content	Hours
1	**UNIT I - Information Systems in Business** The fundamental roles of information systems in business, Trends in information systems, The role of e-business in business, Types of information systems (Operations Support Systems, Management Support Systems), Other classifications of information systems.	12
2	**UNIT II - Information Technology in Business** Managerial challenges of information technology, Components of information systems, Information system resources, Information system activities, Recognizing information systems.	10
3	**UNIT III - Developing Business/IT Strategies** Planning fundamentals, Organizational planning: Scenario approach, SWOT analysis, Business models and planning, Business/IT architecture planning, Identifying business/IT strategies, Business application planning, Change management.	10
4	**UNIT IV - Designing Business/IT Solutions** Introduction to cross-functional enterprise applications, Enterprise application integration, Transaction processing systems, Enterprise collaboration systems.	10
5	**UNIT V - e-Business Systems** Information technology in business, Marketing systems, Manufacturing systems, Human resource systems, Accounting systems, Financial management systems.	10

Reference Books

1. James A. O'Brien and George M. Marakas, "Management Information System", Tenth Edition, Tata McGraw Hill, 2013.
2. W.S. Jawadekar, "Management Information Systems", Tata McGraw Hill Private Limited, New Delhi, 2009.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Data Mining and Data Warehousing</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of databases, SQL, and data management.
2. Familiarity with statistics and probability.
3. Introductory knowledge of machine learning (Recommended).

COURSE OBJECTIVE: This course aims to enable learners:

1. Introduce students to the fundamentals of data warehousing and data mining.
2. Equip students with data reduction skills like wavelets, attribute selection, and clustering.
3. Develop a deep understanding of association rule mining, including algorithms like Apriori and FP-Growth.
4. Explore classification techniques such as decision trees, Naive Bayes, and K-Nearest Neighbors for predictive modeling.
5. Provide exposure to cluster analysis and the real-world applications of data mining across various domains.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the foundational concepts of data mining and data warehousing, including data cubes and OLAP.
CO2	Apply data reduction techniques like wavelet transforms, clustering, and data transformation strategies to large datasets.
CO3	Implement association rule mining using algorithms such as Apriori and FP-Growth for frequent pattern generation.
CO4	Evaluate and apply various classification techniques like decision trees, Naive Bayes, and K-Nearest Neighbors to solve classification problems.
CO5	Perform cluster analysis and explore the emerging trends in data mining applications, including sequence data mining and visual data mining.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to Data Mining and Data Warehousing** Introduction to data mining and data warehousing, Data Cube and OLAP modeling, Data warehouse implementation, Data mining types, Data patterns, Data cleaning, Data integration.	12
2	**UNIT II - Data Reduction** Data reduction techniques: Wavelet transforms, Attribute subset selection, Histogram, Clustering, Sampling, Data cube aggregation. Data transformation: Overview and normalization strategies.	10
3	**UNIT III - Mining Frequent Patterns** Association rules: Problem definition, Frequent item set generation, Apriori principle, Support and confidence measures, Apriori algorithm, FP-Growth, Maximal and Closed frequent item sets.	10
4	**UNIT IV - Classification** Problem definition, General approaches to classification, Decision trees: Construction, Methods for attribute tests, Best split selection, Naive Bayes classifier, Bayesian belief networks, K-Nearest neighbor classification. Prediction: Accuracy measures, Ensemble methods.	10
5	**UNIT V - Cluster Analysis & Data Mining Applications** Requirements for cluster analysis, Clustering methods, Data mining applications & trends: Mining sequence data, Time series mining, Visual data mining.	10

Reference Books

1. Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 3rd Edition, 2011.
2. George M. Marakas, "Modern Data Warehousing, Mining and Visualization", Pearson Education, 2003.
3. W.H. Inmon, "Building the Data Warehouse", Wiley Dreamtech, 3rd Edition.
4. Michael J.A. Berry & Gordon S. Linoff, "Mastering Data Mining", Wiley Publications.
5. Sam Anahory & Dennis Murray, "Data Warehousing", Pearson Education.

SEMESTER I

<table>
<tr><td>Course Title</td><td colspan="8">Probability and Statistics</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners.

1. To analyze statistical data through different techniques.
2. To apply different statistical measures on data.
3. To identify, formulate and solve problems.
4. To understand the different types of probability and their applications.
5. To understand the different types of random variables.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Distinguish between quantitative and categorical data
CO2	Apply different statistical measures on data
CO3	Identify, formulate and solve problems
CO4	Classify different types of Probability and their fundamental applications
CO5	Explore the different types of random variables.

Course Content

Unit No	Content	Hours
1	**UNIT-I-Measures of Central Tendency & Measures of Dispersion:** Frequency Distribution, Histogram, Stem and leaf diagram, ogives, Frequency Polygon, Mean, Median, Mode, Range, Quartile Deviation, Mean Deviation, Box whisker plot, Standard Deviation, Coefficient of Variation.	12
2	**UNIT-II-Skewness, Correlation & Regression:** Karl Pearson's coefficient of Skewness, Bowley's coefficient of Skewness, Scatter Diagram, Karl Pearson's coefficient of correlation, Spearman's rank correlation coefficient , Linear Regression and Estimation, coefficients of regression.	10
3	**UNIT-III-Theory of Attributes:** Classes and Class Frequencies, Consistency of Data, Independence of Attributes, Association of Attributes **Testing of Hypothesis**: Hypothesis, Type I and Type II errors. Tests of significance – Student's t-test:Single Mean, Difference of means, paired t-test, Chi-Square test:Test of Goodness of Fit, Independence Test	10
4	**UNIT-IV-Introduction to Probability:** Random experiment, Sample space, Events, Axiomatic Probability, Algebra of events **Conditional Probability:** Conditional Probability, Multiplication theorem of Probability, Independent events, Baye's Theorem	10
5	**UNIT-V-Random variables**: Discrete random variable, Continuous random variable, Two-dimensional random variable, Joint probability distribution, Stochastic independence **Mathematical Expectation**: Expected value of a random variable, Expected value of a function of a random variable, Properties of Expectation and Variance, Covariance	10

Reference Books

1. Fundamentals of Mathematical Statistics – 1st Edition S.C.Gupta, V.K.Kapoor , S Chand.
2. Introduction to Probability & Statistics – 4th Edition J.Susan Milton, Jesse C. Arnold Tata McGraw Hill
3. Fundamentals of Statistics : 7th edition S C Gupta, Himalaya Publishing house
4. Probability and Statistics with Reliability, Queuing, And Computer Science Applications (English) 1st Edition: Kishore Trivedi, PHI
5. Schaum's Outlines Probability, Random Variables & Random Process 3rd Edition Tata McGraw Hill
6. Probability & Statistics for Engineers: Dr J Ravichandran, Wiley
7. Statistics for Business and Economics: Dr Seema Sharma, Wiley

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Java Programming</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">48</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>03</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: C++

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic concepts and fundamentals of platform independent object oriented language.
2. To develop skills to write programs using inheritance, packages and interfaces.
3. To acquire skills in writing programs using exception handling techniques and multithreading.
4. To understand the concept of handling strings and events.
5. To develop skills to create java applications using applets, swings and networking concepts.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Explain the basics of OOPs concepts.
CO2	Identify classes, objects, members of a class and use packages and interfaces appropriately.
CO3	Demonstrate for Java program formulate thread, synchronization and exception handling concepts.
CO4	Use the concept of string, event handling, simple data structures like arrays and members of classes of Java API in application development.
CO5	Design and develop Java based UI and applications using applets, swing components.

Course Content

Unit No	Content	Hours
1	**UNIT-I-Java Basics** History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, using objects as parameters, Returning objects, recursion, nested and inner classes, exploring string class	10
2	**UNIT-II-Inheritance, Packages and Interfaces** **Inheritance** - Inheritance Basics, Using 'super', Creating Multilevel hierarchy, Method Overriding, Dynamic method dispatch, Using Abstract Classes, Using final with Inheritance. **Packages and Interfaces** - Packages, Access Protection, Importing packages, Interfaces, Default interface methods, Use static methods in an interface. **Exploring java.io-** The I/O classes and interfaces, The Byte streams, The Character streams	8
3	**UNIT III- Exception Handling and Multithreaded Programming** Exception Handling Fundamentals, Exception Types, Uncaught Exception, Using try and catch, Multiple catch clauses, Nested try Statements, throw, throws, finally, Java Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exception, Three Recently Added Exception Features. The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread priorities, Synchronization, Inter Thread communication, Suspending, Resuming, and Stopping Threads, Obtaining A Thread's State, Using Multithreading. **Enumerations, Type Wrappers, Autoboxing, Annotations**	10
4	**UNIT-IV-String and Event Handling** String fundamentals, String Constructors, The Length () method, character extraction, String comparison, searching strings, Modifying a string, changing the case of the characters within the string, String buffer and String builder. Delegation event model, Event classes, Event Listener Interfaces, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, check box, checkbox groups, choices, lists, scroll bars, textfield, textarea, menu bars and menus, dialog boxes. layout manager – layout manager types – border, grid, flow, card and grid bag.	10
5	**UNIT-V- Applets, Swings and Networking with Java.Net** The Applet Class, Types of Applets, Applet Basics, Applet Architecture, An Applet Skeleton, Simple Applet Display Methods, Requesting Repaint, The HTML APPLET tag . Swings- Introduction, limitations of AWT, two key swing features, MVC architecture, components and containers, a simple swing application, exploring swing- ImageIcon and JLabel, JTextField, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, JTabbedPane, JScrollPane, Trees, and JTable. Exploring collection framework, Collection overview, Collection classes and interfaces, Array class.	10

Reference Books.

1. Herbert Schildt. Java - The Complete Reference, Ninth Edition. Oracle Press, McGraw Hill Education (India) Edition 2014.
2. Cay S. Horstmann, Gary Cornell. Core Java, Core Java Volume-1 – Fundamentals, 9th edition, Pearson Education, 2014.
3. Timothy Budd, “Understanding Object-oriented programming with Java”, Updated Edition, Pearson Education, 2000.

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Advanced Database Management System</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">48</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>03</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of Database Management Systems (DBMS)
2. Familiarity with SQL and relational database concepts
3. Understanding of basic programming concepts, preferably in C++ or Java
4. Exposure to NoSQL and Cloud Databases (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. To deepen understanding of database architecture and models.
2. To master advanced SQL and query optimization techniques.
3. To apply concepts of database indexing and optimization.
4. To understand and implement transaction management and concurrency control.
5. To explore distributed, NoSQL, and specialized databases.
6. To develop the skills to design and implement complex database systems that support real-world applications.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Demonstrate the fundamentals of data models and conceptualize and depict a database system. Make use of ER diagrams in developing ER Models.
CO2	Summarize SQL and relational database design.
CO3	Illustrate transaction processing, concurrency control techniques, and recovery.
CO4	Infer the database design in real-world entities.
CO5	Explore and evaluate modern databases, including NoSQL and cloud-based systems, and their applications in various domains.

Unit No	Course Content	Hours
1	**UNIT I - Introduction to DBMS Concepts & Architectures** Database and Need for DBMS, Characteristics of DBMS, Database 3-tier schema (ANSI/SPARC) and system architecture of DBMS, Views of data - Schemas and instances, Data Independence. Centralized systems, Client-Server systems, Transaction servers, Data servers, Cloud-based servers. Indexing and Hashing - Basic concepts of indexing, ordered index, B+ tree index (overview only), Hashing concepts, types of hashing.	10
2	**UNIT II - Data Modelling and Relational Database Design** Data Modelling using ER Diagram: Representation of Entities, Attributes, Relationships, and their Types, Cardinality, Generalization, Specialization. Relational Data Model: Structure of Relational Database Model, Types of Keys, Referential Integrity Constraints, Codd's Rules. Database Design: E-R to Relational Mapping, Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms based on Primary Keys - General Definitions of 1st, 2nd, and 3rd Normal Forms.	10
3	**UNIT III - Advanced SQL and Query Optimization** Overview of SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, SQL Data Types and Schemas. Basic Operations: Joins, Subqueries, Transactions, Views, Aggregate Functions. Integrity Constraints, Basic Query Optimization Techniques, Indexes in SQL.	8
4	**UNIT IV - Transaction Management and Concurrency Control** Concept of Transaction, ACID Properties, States of Transaction, Concurrency Control Problems, Lock-based Protocols, Deadlock Handling. Crash Recovery and Backup: Log-based Recovery, Checkpointing, Basic Backup Strategies.	10
5	**UNIT V - Modern Databases (NoSQL and Cloud Databases)** Introduction to NoSQL Databases: Types of NoSQL Databases (Key-Value, Document, Column-Family, Graph), Basic Use Cases. Cloud Databases: Introduction to Cloud Database Concepts, Benefits and Challenges, Overview of Popular Cloud DBMS (Amazon RDS, Google Cloud SQL, Azure SQL Database).	10

Reference Books

1. Henry F Korth, Abraham Silberschatz and S. Sudharshan, "Database System Concepts", Sixth Edition, McGraw Hill, 2011..
2. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Fifth Edition, Pearson Education/Addison Wesley, 2007.
3. Thomas Cannolly and Carolyn Begg, "Database Systems, A Practical Approach to Design, Implementation and Management", Third Edition, Pearson Education, 2007.

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Java Programming Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">02</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: C++

COURSE OBJECTIVE: This course aims to enable learners:

1. To learn the concepts of data types, operators and control statements.
2. To develop skills to write programs using inheritance, exception handling and multithreading.
3. To acquire skills in writing programs using applets, swings, JDBC and MYSQL.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Implement the programs to know the use of data types, operators and control statements.
CO2	Implement the OOPs Concept and write efficient Java programs using Inheritance, Exceptions and Multithreading.
CO3	Implement the concepts like Applet and Swings.

Course Content

1. Write a program to check if the entered number is ODD or EVEN.
2. Write a program to find the factorial of a given number using recursion.
3. Write a program to find the sum of the digits in a given number.
4. Write a program to swap two numbers without using a temporary variable.
5. Write a program that accepts a name and displays the name with a greeting message using Class.
6. Write a program to generate a salary for an employee using class, object, constructors, methods and access control. Different parameters to be considered are Emp_No, Emp_Name, Age, Basic, DA, HRA,PF, PT. Find Gross and Net salary.
7. Write a program to demonstrate Constructor Overloading and Method Overloading.
8. Implement Inner class and demonstrate its Access protection.
9. Write a program in Java for String handling which performs the following:
    a. Checks the capacity of StringBuffer objects.
    b. Reverses the contents of a string given on console and converts the resultant string in upper case.
    c. Reads a string from the console and appends it to the resultant string of (b).
10. Write a program to demonstrate Inheritance.
11. Write a program to implement Multiple inheritance using interfaces to calculate the area of a rectangle and triangle.
12. Write a JAVA program which has
    a. A Class called Account that creates an account with 500 Rs minimum balance, a deposit() method to deposit amount, a withdraw() method to withdraw amount and also throws Less Balance Exception if an account holder tries to withdraw money which makes the balance become less than 500 Rs.
    b. A Class called Less Balance Exception which returns the statement that says withdrawal amount (Rs) is not valid.
    c. A Class which creates 2 accounts, both account deposit money and one account tries to withdraw more money which generates a Less Balance Exception and takes appropriate action for the same.
13. Write a JAVA program using Synchronized Threads, which demonstrates the Producer Consumer concept.
14. Write a JAVA program to implement a Queue using user defined Exception Handling (also make use of throw, throws.).

15. Complete the following:
    a. Create a package named shape.
    b. Create some classes in the package representing some common shapes like Square, Triangle and Circle. Import and compile these classes in other programs.
16. Write a JAVA Program To Create an enumeration Day of Week with seven values SUNDAY through SATURDAY. Add a method called Workday( ) to the Day of Week class that returns true if the value on which it is called is MONDAY through FRIDAY. For example, the call Day Of Week SUNDAY is Workday ( ) returns false.
17. Write a JAVA program which has
    a. A Interface class for Stack Operations
    b. A Class that implements the Stack Interface and creates a fixed length Stack.
    c. A Class that implements the Stack Interface and creates a Dynamic length Stack.
    d. A Class that uses both the above Stacks through Interface reference and does the Stack operations that demonstrates the runtime binding.
18. Write a program to print a chessboard pattern.
19. Write a JAVA Program which uses File Input Stream / File Output Stream Classes.
20. Write a program to demonstrate utilities of Linked List Class.
21. Write a JAVA applet program, which handles keyboard events.
22. Write a JAVA Swing program, to design a form.

Reference Books

1. Herbert Schildt. Java - The Complete Reference, Ninth Edition. Oracle Press, McGraw Hill Education (India) Edition-2014.
2. Jim Keogh, J2EE The Complete Reference, Tata McGraw Hill Education Private Limited-2011.
3. Timothy Budd, “Understanding Object-oriented programming with Java”, Updated Edition, Pearson Education, 2000.
4. Java 8 Programming Black Book, Dreamtech Press-2012

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Advanced Database Management Systems Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">02</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. Design database schema for a given application.
2. Working on database systems, designing of databases, creating relational databases and analysis of table design.
3. Acquire skills in using SQL commands for data definition and data manipulation.

COURSE OUTCOME (CO)

COURSE OUTCOMES	
CO1	Implementation of data models, modeling notations, SQL queries.
CO2	Design and implement a database schema for a given problem.
CO3	Design and build a simple database system considering the different concepts of RDBMS.

Course Content

Pgm No.	Content	Hours
1	**Introduction Lab** Basic Concepts of DBMS will be covered Like: ● Types of DBMS ● DBMS Languages ● SQL-Structured Query Language ● Basic Instructions given with respect to Lab ● Installation Procedure given to students with respect to MySQL Software ● CREATE, USE, INSERT, DESC and SELECT commands will be explained..	3
2	**Database Number -1 : Banking Database** Consider the following relations: **BRANCH** (branch name: varchar , City : Varchar , Asset : Varchar) **ACCOUNT** (account Number : Varchar, Branch name : Varchar, balance: float) **CUSTOMER** (customer name : Varchar, Street : Varchar , City : Varchar) **DEPOSITOR** (customer name : varchar, account number : varchar) **LOAN** (loan number : varchar, branch name : varchar, amount : integer) **BORROWER** (customer name : varchar, loan number : varchar) ● The tables will be created using CREATE, DESC and tuples will be inserted using INSERT commands and displayed using SELECT Commands. ● Primary Key and Foreign Key relationships will be explained through a demo.	3
3	**Database Number -1 : Banking Database (Continuation) Queries :** ● List the loan number from loan having amount 10000 with a specific branch name. ● List the loan number with an amount between 1000 and 10000. ● List the cname with substring. ● List the number of tuples in the customer. ● List customer name, loan num and amount with specific branch name. ● Various Aggregate functions will be used to retrieve the data from the above constructed databases. ● Various Comparison operators will be used to retrieve the data from the above constructed databases ● Various Logical operators will be used to retrieve the data from the above constructed databases	3
4	**Database Number -2 : Accessories Database** Consider the following relations: **PRODUCT** (maker: varchar, model : varchar, type: varchar) **PC** (model : varchar , speed : varchar, ram : varchar, hdd: varchar , removable disk : int , price : int) **LAPTOP** (model : varchar , speed : varchar, ram : varchar, hdd: varchar , screen: varchar, price : int) **PRINTER** (model: varchar, color : blob , type : varchar, price : int) ● The tables will be created using CREATE, DESC and tuples will be inserted using INSERT commands and displayed using SELECT Commands. ● Primary Key and Foreign Key relationships will be explained through a demo.	3

	● Find the model,speed,RAM,HDD for all pc's, whose price is under 35000. ● Rename the speed column to mhz and HDD column to GB in pc. ● Find all manufacturers of laptops. ● Find all the tuples in the printer for color. ● Various Aggregate functions will be used to retrieve the data from the above constructed databases. ● Various Comparison operators will be used to retrieve the data from the above constructed databases ● Various Logical operators will be used to retrieve the data from the above constructed databases.	
5	**Database Number -3 : Order Processing Database** Consider the following relations: **CUSTOMER** (Cust id : varchar, cust name : varchar, city : varchar) **ORDER** (Order num : varchar, order date : date, cust id : varchar , order amount : int) **ITEM** (item id : varchar , unit price : int) **ORDERITEM** (order num: varchar, item id: varchar, quantity : int) **WAREHOUSE** (warehouse id : varchar, city: varchar) **SHIPMENT** (order num: varchar, warehouse id : varchar, shipdate : date) ● The tables will be created using CREATE, DESC and tuples will be inserted using INSERT commands and displayed using SELECT Commands. ● Primary Key and Foreign Key relationships will be explained through a demo. ● Produce a list in customer name, number of orders, average order amount where the middle column is the total number of orders by the customers and the last column is average order amount for the third column. ● List the order number for orders that were shipped from all the warehouses that the company asked in a specific city. ● Various Aggregate functions will be used to retrieve the data from the above constructed databases. ● Various Comparison operators will be used to retrieve the data from the above constructed databases ● Various Logical operators will be used to retrieve the data from the above constructed databases.	3
6	**Database Number - 4 : Supplier Database** Consider the following relations: **SUPPLIER** (Supplier id : varchar, name : varchar , address : varchar) **PART** (part id : varchar, partname : varchar, color : tinyblob) **CATALOG** (supplier id : varchar, part id : varchar, cost : int) ● The tables will be created using CREATE, DESC and tuples will be inserted using INSERT commands and displayed using SELECT Commands. ● Database Number -4 : Supplier Database (Continuation) Queries ● Primary Key and Foreign Key relationships will be explained through a demo. ● Find the names of suppliers who supply only red parts. ● Find supplier id of suppliers who supply red and green parts. ● Find the supplier id of the supplier who supplies some red part OR whose address is 'Mysuru'. ● Find the supplier id of suppliers who supply some red and some green parts. ● Find the supplier id of suppliers who supply every part.	3

7	**Database Number -5 : University Database -** Consider the following relations: **CLASSROOM** (building varchar(30), roomno varchar(30) primary key, capacity int) **DEPARTMENT** (deptname varchar(30) primary key,building varchar(30),budget decimal(10,2) **COURSE**(courseid varchar(30) primary key, title varchar(30),deptname varchar(30), credits int) **INSTRUCTOR** (iid varchar(30) primary key, name varchar(30), deptname varchar(30),salary int) **STUDENT**(stdid varchar(30) primary key,name varchar(30),deptname varchar(30), totalcredit int) **TEACHES**(iid varchar(30),courseid varchar(30), secid varchar(30),sem varchar(30), year varchar(30)) **SECTION** (courseid varchar(30),secid varchar(30) primary key,sem varchar(30), year varchar(30),building varchar(30),roomno varchar(30),timeid varchar(30)) **TIMESLOT**(timeid varchar(30) primary key,day varchar(30), starttime time, endtime time); **PREREQUISITE** (courseid varchar(30),preid varchar(30)) **ADVISOR** (stdid varchar(30), iid varchar(30)) **TEXT**(iid varchar(30), courseid varchar(30),secid varchar(30),sem varchar(30), year varchar(30), grade varchar(30)) ● The tables will be created using CREATE, DESC and tuples will be inserted using INSERT commands and displayed using SELECT Commands. ● Primary Key and Foreign Key relationships will be explained through a demo.	4
8	**Database Number -7 : University Database (Queries) Continuation** ● Retrieve the names of the instructors and order them by the departmental name. ● Retrieve instructors with salaries between specified ranges: ○ Between 10,000 and 20,000 ○ Between 50,000 and 70,000 (specific to the MBA department) ● Find salary statistics for instructors: ○ Average salary for a particular department ○ Minimum salary for a department ○ Average salary grouped by department, where the average is greater than 50,000 ● Display instructor details: ○ Instructor teaching in a specific building, e.g., "Golden Jubilee Block"	5
9	**Database Number -7 : University Database (Queries) Continuation** ● Retrieve student details: ○ Students who have taken at least one MCA course ○ Students with the maximum CGPA ● Department and enrollment queries: ○ Total number of students in each department, section-wise ○ Enrollment of sections offered in a specific year (e.g., 2015 and odd 2019) ● Course details: ○ Delete courses that have never been offered ○ List all course titles of a specific instructor ○ Retrieve all course titles taught in specific semesters (odd/even 2018) ○ Create a view listing MCA course sections for even 2016	5

10	**Database Number -7 : University Database (Queries) Continuation** ● Instructor and department associations: ○ Department names for all instructors ○ Instructor names and details, including department and building name ○ Instructor names that include specific letters (e.g., 'ma') and the courses they teach ○ Instructor names in descending order of salary ○ Instructor names not matching specified names (e.g., "Charlie" and "Deepika") ● Credits and CGPA: ○ Total credits offered by each department ○ Total CGPA scored by students of each department ● Additional data manipulations: ○ Increase the salary of each instructor by 25,000 and list their names and IDs ● Room and course queries: ○ Courses belonging to the MBA department running in specific rooms during 2015	

Text Books

1. A. Silberschatz, Henry.F.Korth, S.Sudharshan, “Database System Concepts”, 7th Edition,2017

Reference Books

1. Raghu Ramakrishnan and J Gehrke,” Database Management Systems”, 3rd Edition, 2016.
2. C.J.Date, AKannan, S..Swamynathan ,”An Introduction to Database System”, 8th Edition,2009.
3. RamezElmasri, Shamkant.B.Navathe, “Database Systems”, 7th Edition,2016.

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Analysis and Design of Algorithms</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of programming concepts
2. Familiarity with basic mathematical concepts, such as logic and set theory.
3. Problem Solving and Analytical skills
4. Basic programming Skills

COURSE OBJECTIVE: This course aims to enable learners:

1. Describe computational solutions to well known problems like searching, sorting etc.
2. Estimate the computational complexity of different algorithms.
3. Devise an algorithm using appropriate design strategies for problem solving.
4. Gain the ability to analyze and apply data structures and Algorithms in real-world scenarios, focusing on efficiency and Optimization.

COURSE OUTCOME (CO)

COURSE OUTCOMES	
CO1	Describe computational solutions to well known problems like searching, sorting etc.
CO2	Implement abstract data types (ADT) such as arrays, strings, stacks, and queues to solve algorithmic problems.
CO3	Estimate the computational complexity of different algorithms.
CO4	Devise an algorithm using appropriate design strategies for problem solving.
CO5	Gain the ability to analyze and apply data structures and Algorithms in real-world scenarios, focusing on efficiency and Optimization.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction and the fundamentals of the Analysis of Algorithm** Efficiency Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures. Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical analysis of Recursive and Non-recursive algorithms With Examples.	12
2	**UNIT -II- Brute Force and Divide and Conquer:** Selection Sort and Bubble Sort, Sequential Search and String Matching, Exhaustive Search, Merge- sort, Quick-sort, Binary Search, Binary tree Traversals and related properties.	10
3	**UNIT-III- Decrease-and-Conquer, Transform-and-Conquer** Insertion Sort, Depth First search and Breadth First Search, Topological sorting, Algorithms for Generating Combinatorial Objects. Presorting, Balanced Search Trees, Heaps and Heap sort, Problem Reduction.	10
4	**UNIT-IV - Space and Time Trade offs and Dynamic Programming** Sorting by Counting, Input Enhancement in String Matching, Computing a binomial coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions.	10
5	**UNIT V - Limitations of Algorithm Power** Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems. Coping with Limitations of Algorithm Power: Backtracking: n-Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem. Branch-and- Bound: Assignment Problem, Knapsack Problem, Traveling Salesperson Problem.	10

Reference Books

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
3. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education).

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Big Data Analytics</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of Database Management Systems (DBMS)
2. Familiarity with programming concepts and languages such as Java or Python
3. Understanding of data structures and algorithms
4. Introductory knowledge of distributed systems (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamental concepts of big data and its significance in modern data analysis.
2. To explore various tools and techniques for analyzing and processing large datasets, including Hadoop and MapReduce.
3. To gain practical knowledge of big data technologies such as Hadoop, HDFS, Pig, and Hive.
4. To develop skills in designing and implementing big data solutions for real-world applications.
5. To explore advanced topics like recommendation systems and social network analysis using big data.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the fundamental concepts of big data and its challenges, including the need for standards and effective management.
CO2	Analyze and apply data analysis techniques using MapReduce and understand the evolution of analytics.
CO3	Gain proficiency in using Hadoop Distributed File System (HDFS) and related tools to process and analyze large datasets.
CO4	Administer Hadoop and utilize platforms like Pig and Hive for big data management and query processing.
CO5	Apply big data techniques to build recommendation systems and analyze social network graphs, addressing real-world data challenges.

Unit No	Course Content	Hours
1	**UNIT I - Introduction to Big Data** What is big data? The importance of "big" and "data." How big data differs from traditional data. Risks associated with big data, the need to manage big data effectively. The structure of big data. Exploring and filtering big data, integrating big data with traditional data. The need for standards in big data management. Web data insights: what web data reveals and its applications. Overview of big data sources and the value they provide.	12
2	**UNIT II - Data Analysis** Evolution of analytic scalability. The convergence of analytic and data environments. Massively parallel processing systems. Cloud computing, Grid computing, MapReduce. Enterprise analytic sandbox, enterprise analytic data sets. Evolution of analytic tools and methods. Approaches to analysis: framing the problem, statistical significance versus business relevance. Enabling analytic innovation and overcoming traditional barriers.	10
3	**UNIT III - MapReduce and Hadoop Distributed File System (HDFS)** Weather dataset case study. Data analysis using Unix tools. Data analysis using Hadoop. Scaling out Hadoop, Hadoop Streaming, and Hadoop Pipes. HDFS design, HDFS concepts, and the Java interface. Data flow in Hadoop. Parallel copying with distcp. Introduction to Hadoop Archives.	10
4	**UNIT IV - Introduction to Hadoop and Its Operations** Administering Hadoop: HDFS monitoring and maintenance. Pig: Installing and running Pig, comparison with traditional databases, Pig Latin, user-defined functions. Hive: Installing and running Hive, comparison with traditional databases, HiveQL, querying data.	10
5	**UNIT V - Recommendation Systems and Mining Social Network Graphs** Introduction to recommendation systems: models, content-based recommendations, collaborative filtering, dimensionality reduction. The Netflix challenge. Mining social network graphs: social networks as graphs, clustering, partitioning, neighborhood properties of graphs.	10

Reference Books

1. "Big Data: Concepts, Technology, and Architecture" by Balamarugan Balusamy, Nandhini Abirami R, Seifedine Kadry, and Amir Gandomi (2024 Edition).
2. "Big Data Analytics for Cyber-Physical Systems" by Houbing Song, et al. (2019 Edition).
3. "Data Science and Predictive Analytics: Biomedical and Health Applications using R" by Ivo D. Dinov (2023 Edition).

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Foundations of Cyber Security</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Understanding of Computer Hardware and Software
2. Familiarity with Operating Systems
3. Familiarity of Computer networks and various protocols with their applications

COURSE OBJECTIVE: This course aims to enable learners:

1. Understand the fundamentals of the cybersecurity domain and related issues.
2. Understand the interdisciplinary nature of cyber security domain
3. Knowledge about various Cyber laws and regulations
4. Illustrate Tools and Methods used on Cybercrime , Understanding about Phishing and Identity Theft.
5. Justify the need of computer forensics.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the fundamentals of the cyber security domain and related issues.
CO2	Describe Cyber offenses and Botnets
CO3	Knowledge about various Cyber laws and regulations
CO4	Illustrate Tools and Methods used on Cybercrime , Understanding about Phishing and Identity Theft.
CO5	Justify the need of computer forensics

Course Content

Unit No	Content	Hours
1	**UNIT-I- Introduction to Cybercrime:** Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Who are Cybercriminals? Classifications of Cyber Crimes, Cybercrime: The Legal Perspectives, Cybercrimes: Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes, Cybercrime Era: Survival Mantra for the Netizens. Cyber offenses: How Criminals Plan Them: How Criminals Plan the Attacks, Social Cyber stalking Cyber cafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector.	12
2	**UNIT-II- Cybercrime: Mobile and Wireless Devices:** Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era.	10
3	**UNIT-III- Tools and Methods Used in Cybercrime:** Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks. Phishing and Identity Theft: Introduction, Phishing, Identity Theft (ID Theft).	10
4	**UNIT-IV- Understanding Computer Forensics:** Introduction, Historical Background of Cyber forensics, Digital Forensics Science, The Need for Computer forensics , Cyber Forensics and Digital Evidence, Digital Forensics Life Cycle, Chain of Custody Concept, Network Forensics, Approaching a Computer Forensics Investigation, Understanding the Requirements, Relevance of the OSI 7 Layer Model to Computer Forensics, Computer Forensics from Compliance Perspective, Challenges in Computer Forensics, Special Tools and Techniques, Forensics Auditing, Anti forensics.	10
5	**UNIT-V- Cybercrime and Cyberterrorism:** Social, Political, Ethical and Psychological Dimensions: Introduction, Intellectual Property in the Cyberspace, The Ethical Dimension of Cybercrimes, The Psychology, Mindset and Skills of Hackers and Other Cybercriminals, Sociology of Cybercriminals, Information Warfare: Perception or an Eminent Reality? Cybercrime: Illustrations, Examples and Mini-Cases: Introduction, Real Life Examples, Example 1: Official Website of Maharashtra Government Hacked, Example 2: Email Spoofing instances, Example 3: Email Bombing involving a foreigner. Mini Cases: Mini Case 1: Indian Cyber Defamation case of a young couple, Mini Case 2: Internet Time Stealing, Mini Case 3: Mini Case of Indian case of online gambling. Illustrations of Financial Frauds in Cyber Domain, Digital Signature-Related Crime Scenarios.	10

Reference Books

1. Fundamentals of Cyber Security (Principles Theory & Practices), Bhushan Mayank, Publisher: BPB Publications, 2020.
2. Cyber security , Nina Godbole , Sunit Belapure , Wiley India Pvt Ltd , Reprint 2023
3. Cyber Security Fundamentals, Rajesh Kumar Gautam, BPB Publications, 2019.
4. Mowbray, Copyright © 2014 by John Wiley & Sons, Inc, ISBN: 91-118 - 84965 -1. Dreamtech Press. ISBN: 9789351194736, 2015 Martti Lehto, Pekka Neittaanmäki, "Cyber Security: Analytics, Technology and Automation edited" Springer International Publishing Switzerland, 2015.

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Introduction to Cloud Computing</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. Apply Cloud Computing and Related Technologies.
2. To acquire the knowledge of cloud deployment models.
3. To acquire skills in various services available in cloud computing.
4. To understand the file systems to manage large datasets in cloud computing.
5. To develop skills to use tools and technologies of cloud computing platforms.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understanding of Cloud Computing and Related Technologies.
CO2	Acquiring the knowledge of different cloud deployment models.
CO3	Understanding the different services available in cloud computing.
CO4	Understanding the file systems designed to handle the storage and management of large datasets in cloud computing.
CO5	Understanding the tools and technologies for cloud computing platforms.

Course Content

Unit No	Content	Hours
1	**UNIT-I- Introduction to Cloud Computing:** Introduction to Cloud Computing, Definition, Characteristics, Components, Cloud provider, SLA, Virtualization, Types of virtualization, Server virtualization, storage virtualization, Network Virtualization and application virtualization, Importance of virtualization in cloud, Study of hypervisors.	12
2	**UNIT-II- Cloud deployment models:** Cloud deployment models: Public cloud, Private cloud and Hybrid cloud- Organizational scenarios of clouds, , Deploy application over cloud-Workload distribution, Resource pooling, dynamic scalability, elasticity, Service load balancing, Cloud bursting, Service Technology: SOAP and REST Web services, AJAX and mashups Web services, Service Middleware.	10
3	**UNIT-III- Overview:** Overview, Infrastructure as a Service (IaaS) Cloud Delivery Model, Platform as a Service (PaaS) Cloud Delivery Model, Software as a Service (SaaS) Cloud Delivery Model- Administering & Monitoring cloud services, benefits and limitations- Cloud computing platforms: Infrastructure as a service: Amazon EC2, Platform as a Service: Google App Engine, Microsoft Azure, Utility Computing, Elastic Computing.	10
4	**UNIT-IV- GFS and HDFS:** GFS and HDFS, Big Table, HBase and Dynamo, Map-Reduce: The Map-Reduce model-Cloud Workload	8
5	**UNIT-V- Tools and Technologies for Cloud:** Tools and Technologies for Cloud, Cloud Computing Platform: Eucalyptus, Nimbus, Open Nebula, Cloud Mashups, Cloud Tools: VMWare, Eucalyptus, Cloud Sim, Implementing real time application over cloud platform, QOS Issues in Cloud, data migration, streaming in Cloud, Concepts in Mobile Cloud Computing, Fog Computing, Dockers, Green Cloud, Cloud Computing, IoT Cloud.	12

Reference Books.

1. Thomas Erl, Zaigham Mahmood, and Ricardo Puttini,"Cloud Computing Concepts, Technology & Architecture", Prentice Hall, 2013.
2. A.Srinivasan,J.Suresh,"Cloud Computing, A practical approach for learning and implementation",Pearson,2014.
3. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley,2011

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Artificial Intelligence and Robotics</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of programming (preferably in Python or Java)
2. Understanding of algorithms and data structures
3. Familiarity with basic concepts of mathematics, particularly linear algebra, calculus, and probability
4. Introductory knowledge of logic and reasoning (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand fundamental AI concepts and their real-world applications, explore search strategies and problem-solving techniques, and gain knowledge in logical reasoning, machine learning, and robotics.
2. To explore various search strategies and problem-solving techniques in AI.
3. To gain knowledge of logical reasoning, constraint satisfaction problems, and adversarial search.
4. To develop skills in knowledge representation, machine learning, and reasoning systems.
5. To introduce the basics of robotics, including perception, planning, and control mechanisms.
6. To understand the current trends and future directions of AI and robotics.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the role of intelligent agents and problem-solving strategies, including both uninformed and informed search methods.
CO2	Analyze and apply techniques for constraint satisfaction, adversarial search, and logical reasoning in AI systems.
CO3	Gain proficiency in using first-order logic for knowledge representation and inference in AI.
CO4	Develop an understanding of knowledge representation techniques, learning methods, and the present and future trends in AI.
CO5	Acquire foundational knowledge in robotics, including hardware, perception, planning, and control, and understand their application domains.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to AI, Informed Search, and Exploration** Intelligent Agents: Agents and environment; Rationality; the nature of the environment; the structure of agents. Problem-solving: Problem-solving agents; Example problems; Searching for solutions; Uninformed search strategies. Informed search strategies: Heuristic functions; Online search agents and unknown environments.	12
2	**UNIT II - Constraint Satisfaction, Adversarial Search, Logical Agents** Constraint satisfaction problems; Backtracking search for CSPs. Adversarial search: Games; Optimal decision-making in games; Alpha-Beta pruning. Logical Agents: Knowledge-based agents; The wumpus world as an example world; Logic; Propositional logic; Reasoning patterns in propositional logic.	10
3	**UNIT III - First-Order Logic, Inference in First-Order Logic** First-Order Logic: Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Inference: Propositional versus first-order inference; Unification and lifting; Forward chaining; Backward chaining; Resolution.	10
4	**UNIT IV - Knowledge Representation and Learning, AI: Present and Future** Knowledge Representation: Ontological engineering; Categories and objects; Actions, situations, and events; Mental events and mental objects; The Internet shopping world; Reasoning systems for categories; Reasoning with default information; Truth maintenance systems. Learning: Forms of learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory. AI: Present and Future: Agent components and current trends.	10
5	**UNIT V - Introduction to Robotics** Introduction to Robotics: Robot hardware; Sensors and effectors; Robotic perception: Localization, mapping, and other types of perception. Planning: Configuration space; Cell decomposition methods; Skeletonization methods. Planning for Uncertain Movements: Robust methods for movement planning. Moving: Dynamics and control; Potential field control; Reactive control. Application Domains of Robotics..	10

Reference Books

1. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (4th Edition, 2020)
2. "Introduction to Robotics: Mechanics and Control" by John J. Craig (4th Edition, 2017)
3. "Artificial Intelligence: Foundations of Computational Agents" by David L. Poole and Alan K. Mackworth (2nd Edition, 2017)

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="9">Web Technologies</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: C++/Java/ Python Programming

COURSE OBJECTIVE: This course aims to enable learners:

1. To learn HTML tags and JavaScript Language programming concepts and techniques.
2. To develop the ability to logically plan and develop web pages.
3. To learn to write, test, and debug web pages using HTML and JavaScript.
4. To understand the concept of XAMPP server and node js.
5. To Implement basics of MEAN and FULL stack.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Apply the concepts and usage of web based programming techniques.
CO2	Demonstrate the development of XHTML documents using JavaScript and CSS
CO3	Demonstrate applications of AngularJs and JQuery for the given problem.
CO4	Design and implement user interactive dynamic web based applications.
CO5	Implement modern web applications using MEAN and FULL STACK.

Course Content

Unit No	Content	Hours
1	**OVERVIEW OF WEB TECHNOLOGIES:** HTML5 and CSS: Web browsers, web servers, MIME, HTTP Introduction to XHTML5 tags, Basic syntax and structure, text markups, images, lists, tables, Media tags-audio and video, forms, frames. CSS: Introduction to CSS, Levels of CSS, Selectors, Font, color and Text Properties, BOX Model, Span and Div tags. Introduction to JavaScript, controls statements, Arrays and functions, pattern matching, Element Access, Event Handling.	10
2	**XML and AJAX:** XML-Documents and Vocabularies-Versions and Declaration -Namespaces JavaScript and XML: Ajax-DOM based XML processing Event-oriented Parsing: SAX-Transforming XML Documents Selecting XML Data: XPATH-Template based Transformations: XSLT-Displaying XML Documents in Browsers - Evolution of AJAX -Web applications with AJAX -AJAX Framework.	10
3	**CLIENT-SIDE SCRIPTING :** Introduction to JQuery, Syntax, selectors, events, JQuery HTML, JQuery Effects, JQuery CSS. Introduction to Angular JS, Directives, Expressions, Directives, Controllers, Filters, Services, Events, Forms, Validations, Examples. Bootstrap: Introduction to Bootstrap, First example, containers, Bootstrap elements: colors, tables, images, buttons, button groups, progress bars, Forms, utilities, Classes, alerts, custom forms, Grid System.	12
4	**SERVER-SIDE SCRIPTING:** Essentials of PHP Installation of Web Server, XAMPP Configurations-PHP Forms- GET and POST method - Regular Expressions-Cookies-Sessions- Usage of Include and require statements- File: read and write from the file-PHP Filters-PHP XML Parser-Introduction to Node.js-Node.js Modules and filesystem-Node.js Events.	10
5	**MySQL and MEAN STACK:** PHP with MySQL- Performing basic database operation (DML) (Insert, Delete, Update, Select)-Prepared Statement- Uploading Image or File to MySQL- Retrieve Image or File from MySQL- Uploading Multiple Files to MySQL-Introduction to MEAN and FULL Stack-Real time example for modern web applications using MEAN-MEAN vs Full Stack.	10

Reference Books.

1. Christopher Murphy, Richardclark, OliStudholme, DivysManian, “Beginning HTML5 and CSS3”, Apress Publication,2012.
2. Grant, Andrew, “Beginning AngularJS”, Apress Publication,2014.
3. Matt Doyle, “Beginning PHP 5.3”, Wiley Publisher,2010.
4. Chris Bates, “Web Programming Building Internet Applications”, Wiley India, 3rd Edition, 2007.
5. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, Pete Warden, Big Data Glossary, O‟Reilly,2016.

SEMESTER II

<table>
<tr><td>Course Title</td><td colspan="8">Algorithm Design and Analysis Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">02</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Knowledge of Data Structures
2. Problem Solving and Analytical skills
3. Basic programming Skills
4. Knowledge of Programming Language
5. Familiarity with mathematical concepts

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamental design, analysis, and implementation of basic data structures.
2. To discuss the basic concepts in the specification and analysis of programs.
3. To understand the principles for good program design, by using various design techniques.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Analyze different algorithms and Implement design techniques to solve problems.
CO2	Implement dynamic programming and greedy technique algorithms to solve real time applications.
CO3	Analyze and Implement the string matching and graph - based algorithms.

List of Lab Programs

1. Write a program to sort a list of N elements using Selection Sort and Bubble sort Technique. Compute the time for very large values on N and plot the graph of time complexity.
2. Implement an algorithm for Linear search & Binary search Compute the time for very large values on N and plot the graph of time complexity.
3. Implement an algorithm for Quick sort and Merge sort based on Divide & Conquer Strategy & Compute the time for very large values on N and plot the graph of time complexity.
4. Implement an algorithm for finding Minimum value & Maximum value based on Divide & Conquer Strategy & Compute the time for very large values on N and plot the graph of time complexity.
5. Implement DFS and BFS algorithms for a graph.
6. Write a C program that accepts the vertices and edges for a graph and stores it as an adjacency matrix. Implement functions to print In-Degree, Out- Degree and to display that adjacency matrix.
7. Implement Knapsack problem using Greedy Solution
8. Implement Dynamic Programming algorithm for the 0/1 Knapsack problem.
9. Implement Optimal Binary Search Tree algorithm using Dynamic Programming.
10. Implement Backtracking Algorithm for solving N- Queens
11. Implement the Backtracking algorithm for the sum of subsets problem
12. Implement greedy algorithms for job sequencing with deadlines.

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Software Engineering</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic concepts of various software application domains.
2. To develop skills to apply appropriate models for software development.
3. To acquire skills in software testing.
4. To understand emerging technologies in software development.
5. To understand the principles of agile development

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Identify unique features of various software application domains and classify software applications.
CO2	Choose and apply an appropriate lifecycle model of software development.
CO3	Acquire the knowledge of software testing.
CO4	Understand the concept of security risk management and other emerging technologies.
CO5	Understand the principles of agile development and distinguish agile process models from other process models.

Unit No	Course Content	Hours
1	**UNIT -I – Introduction** Introduction, professional and ethical responsibility, emerging system properties, systems engineering, legacy systems, systems dependability, availability, reliability, safety and security. Software process models, process iteration, process activities, Project Management: Management activities, project planning, project scheduling, risk management. Software requirements, SRS.	10
2	**UNIT -II – Requirement engineering process** Requirement engineering process, System models and CASE workbenches. Software design: Architectural design- system structuring, control models, modular decomposition, domain specific architectures. Object oriented design: objects and object classes, an object oriented design process, design evolution.	10
3	**UNIT -III– Verification and Validation** Verification and validation, Software testing – Different types of testing, Software cost estimation: productivity, estimation techniques, algorithmic cost modeling, project duration and staffing. Fundamentals of software quality management.	10
4	**UNIT -IV – Emerging Technologies** Emerging Technologies: Security concepts, security risk management, design for security, system survivability. Service-oriented software engineering- services as reusable components, service engineering, software development with services. Aspect- oriented software development- The separation of concerns, aspects, join points and pointcuts, software engineering with aspects.	10
5	**UNIT-V- Introduction to Agile software Development** Overview, Objectives ,Three Perspectives on Software Engineering, The Agile Manifesto, Individuals and Interactions over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract, Negotiation, Responding to Change over Following a Plan, Application of Agile Software Development , Teamwork: Overview, Objectives, A Role Scheme in Agile Teams, Remarks on the Implementation of the Role Scheme, Human Perspective on the Role Scheme, Using the Role Scheme to Scale Agile Projects, , Teamwork in learning environments, Customers and Users: Overview, Objectives, The Customer, Customer Role, Customer Collaboration, The User, Combining UCD with Agile Development.	12

Reference Books

1. Software Engineering, Ian Sommerville, 8th Edition, Pearson Education Ltd.,
2. Software Engineering – A practitioner's approach, Roger. S. Pressman, Tata-McGraw Hill 6th Edition.
3. Fundamentals of software engineering, Rajib Mall, Phi learning Pvt. Ltd, 3rd edition.
4. Agile software development, Alistair Cockburn, Pearson Education India

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Python Programming</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">48</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>03</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Concepts of Object-Oriented Programming Language

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic concepts of python programming
2. To acquire the knowledge of writing programs using functions.
3. To write python programs using object oriented concepts.
4. To understand the skills of writing GUI programs in python.
5. To acquire the knowledge of writing applications using databases.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Acquire programming skills in core Python.
CO2	Develop the skills of writing modular programs using functions.
CO3	Acquire Object Oriented Skills in Python.
CO4	Develop the skill of designing Graphical user Interfaces in Python
CO5	Develop an ability to write database applications in Python.

Unit No	Course Content	Hours
1	**UNIT-1- Introduction to Python** Python Basics: Data Types, Operators, Input/ Output Statements, Creating Python Programs, Python Flow Control statements: Decision making statements, Indentation, Conditionals, loops, break, continue, and pass statements. Data Structures Lists: Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences, Comprehensions.	8
2	**UNIT-II-Python Functions** Defining functions, DOC strings, Function parameters: default, keyword required and variable length arguments, key-word only parameters, local and global variables, pass by reference versus value, Anonymous functions, Recursion. Functional Programming: Mapping, Filtering and Reduction, Lambda Functions.	10
3	**UNIT-1II-Object Oriented Programming** Definition and defining a class, Constructor, Destructor, self and del keywords, Access to Attributes and Methods, getattr and hasattr attributes, Data Attributes and Class Attributes, Data Hiding, Inheritance, Static Members. Regular Expressions: Defining Regular Expressions and String Processing.	10
4	**UNIT-1V- File Handling and Python GUI Programming** File object attributes, Read and Write into the file, Rename and Delete a File, Exceptions Handling :Built-in Exceptions and User defined Exceptions GUI Programming, Introduction to Python GUI Programming, Tkinter Programming, Tkinter widgets, Events and Bindings Turtle Graphics Testing: Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.	10
5	**UNIT-V-Working with Django** Rendering Templates into HTML and Other Formats ,Understanding Models,Views, and Templates,Separating the Layers(MVC)-Models,Views,Templates,Overall Django Architecture, Defining and Using Models, Using Models, Templates and Form Processing, Setting up the Database, Using a Database Server, Using SQLite, Creating the Tables.	10

Reference Books

1. Charles Dierbach,"Introduction to Computer Science Using Python",1s tEdition, Wiley India Pvt Ltd.
2. MarkLutz,“Programming Python”, 4thEdition,O’ReillyMedia, 2011.
3. WesleyJChun,“Core Python Applications Programming”,3rd Edition,Pearson Education India, 2015.
4. Roberto Tamassia,Michael H Goldwasser,Michael TGoodrich,“DataStructures and Algorithms in Python”,1st Edition, Wiley India Pvt Ltd, 2016.
5. Reema Thareja, “Python Programming using problem solving approach”, Oxford university press, 2017.
6. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
7. Learning Python, MarkLutz, O'reilly

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Python Programming Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">24</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">02</td><td>Credits</td><td>01</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Concepts of Object-Oriented Programming Language

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic concepts of python programming.
2. To acquire the knowledge of writing programs using exception handling, multithreading and performing operations on files.
3. To write python programs by creating databases and to solve real world applications.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	To understand the concept of data types, control statements, functions, package, class and object.
CO2	To develop programs using exception handling, multithreading and performing operations on file.
CO3	Acquire Skills in creating databases in Python and to solve real world applications.

Course Content

1. Python installation and configuration with windows
2. Programs for understanding the data types, control flow statements, blocks and loops
3. Programs for understanding functions, use of built in functions, user defined functions
4. Programs to use existing modules, packages and creating modules, packages
5. Programs for implementations of all object-oriented concepts like class, method, inheritance, polymorphism etc.
6. Write a program to validate Password, email and URL.
7. Programs for Pattern finding.
8. Programs covering all the aspects of Exception handling, user defined exception, Multithreading.
9. Programs demonstrating the IO operations like reading from file, writing into file from different file types like data file, binary file, etc.
10. Programs to perform searching, adding, updating the content from the file.
11. Program for performing CRUD operation with MongoDB and Python
12. Basic programs with NumPy as Array, Searching and Sorting, date & time and String handling
13. Programs for series and data frames should be covered.
14. Programs to demonstrate data pre-processing and data handling with data frame
15. Program for data visualization should be covered.

Reference Books.

1. Charles Dierbach,"Introduction to Computer Science Using Python",1s tEdition, Wiley India Pvt Ltd.
2. MarkLutz,**"**Programming Python**"**, 4thEdition,O**'**ReillyMedia, 2011.
3. WesleyJChun,**"**Core Python Applications Programming**"**,3rd Edition,Pearson Education India, 2015.
4. Roberto Tamassia,Michael H Goldwasser,Michael TGoodrich,**"**DataStructures and Algorithms in Python**"**,1st Edition, Wiley India Pvt Ltd, 2016.
5. Reema Thareja, **"**Python Programming using problem solving approach**"**, Oxford university press, 2017.
6. Python Programming: A Modern Approach, Vamsi Kurama, Pearson Learning Python, MarkLutz, O'reilly

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="9">Machine Learning</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of programming (preferably in Python or R)
2. Understanding of algorithms and data structures
3. Familiarity with probability and statistics
4. Introductory knowledge of linear algebra and calculus

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamental concepts and frameworks of machine learning.
2. To explore and implement various supervised learning algorithms and their applications.
3. To gain knowledge of unsupervised learning techniques and their role in data analysis.
4. To comprehend probabilistic graphical models and their use in machine learning.
5. To introduce reinforcement learning and other advanced topics in machine learning, such as ensemble methods and genetic algorithms.
6. To develop the skills to design, analyze, and evaluate machine learning experiments.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the basic concepts, types, and significance of machine learning, and identify the challenges involved in designing learning systems.
CO2	Apply supervised learning algorithms, including decision trees, logistic regression, and neural networks, and evaluate their performance on various tasks.
CO3	Implement unsupervised learning techniques, including clustering and dimensionality reduction, and analyze their effectiveness in data exploration.
CO4	Develop and utilize probabilistic graphical models such as Bayesian networks and hidden Markov models for inference and learning.
CO5	Explore reinforcement learning, ensemble methods, and experiment design for model validation.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to Machine Learning** Introduction to machine learning, definition, importance of machine learning, machine learning framework, types of machine learning, relation to other fields, examples of machine learning applications, designing a learning system, issues in machine learning.	12
2	**UNIT II - Supervised Learning** Introduction to supervised learning, decision tree-based classifier, logistic regression, Bayesian theory-based classifier, neural network-based classifier, nearest neighbor classifier, support vector classifier, performance evaluation.	10
3	**UNIT III - Unsupervised Learning** Introduction to unsupervised learning, clustering methods, criteria functions for clustering, similarity measures, component analysis, low-dimensional analysis, and multidimensional scaling.	10
4	**UNIT IV - Probabilistic Graphical Models** Directed graphical models: Bayesian networks, exploiting independence properties, from distributions to graphs. Naïve Bayes classifiers, Markov models, hidden Markov models, inference, learning generalization.	10
5	**UNIT V - Reinforcement Learning** Reinforcement learning, genetic algorithms, analytical learning, ensemble of classifiers, design and analysis of machine learning experiments.	10

Reference Books

1. "Pattern Recognition and Machine Learning" by Christopher M. Bishop (2006)
2. "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy (2012)
3. "Machine Learning" by Tom M. Mitchell (1997).

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Cloud Computing Architecture</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Knowledge of Programming Skills
2. Familiarity with Databases
3. Basics Networking , Security and Privacy
4. Familiarity with Operating Systems
5. Knowledge of Agile Development
6. Understanding of Virtualization

COURSE OBJECTIVE: This course aims to enable learners:

1. With the fundamentals and essentials of Cloud Computing
2. A sound foundation of the Cloud Computing and Related Technologies
3. Appreciate the requirements of various service paradigms in Cloud Computing
4. Applying virtualization concept in reality
5. To start using and adopting Cloud Computing services and tools in their real life scenarios.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the fundamentals and essentials of Cloud Computing
CO2	Understand the subtle architectural difference in Public and Private Clouds.
CO3	Appreciate the requirements of various service paradigms in Cloud Computing.
CO4	Design, Develop & Demonstrate real-world applications from Cloud computing.
CO5	Applying virtualization concepts in reality.

Course Content

Unit No	Content	Hours
1	**UNIT- I - Introduction to Cloud Computing** Cloud Computing definition , Characteristics, Components , private, public and hybrid cloud. Cloud types; IaaS, PaaS, SaaS. Benefits and challenges of cloud computing, public vs private clouds, role of virtualization in enabling the cloud; Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery; next generation Cloud Applications- Cloud computing Architecture– Cloud containers.	10
2	**UNIT-II-Web Service Architecture:** Web Service Architecture – Web Service APIs – Web service Authentication - Web service authentication methods - Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages.	12
3	**UNIT-III-Cloud Economics:** Reliability, availability and security of services deployed from the cloud. Performance and scalability of services, tools and technologies used manage cloud services deployment; Cloud Economics: Cloud Computing infrastructures available for implementing cloud based services. Economics of choosing a Cloud platform or organization, based on application requirements, economic constraints and business needs (e.g Amazon, Microsoft and Google, Salesforce.com, Ubuntu and Redhat).	12
4	**UNIT-IV-Programming Models for Cloud Computing:** Programming Models for Cloud Computing - Software Development in Cloud - Service creation environments to develop cloud based applications. Development environments for service development; Amazon, Azure, Google App.	8
5	**UNIT-V- Analysis of Case Studies:** Analysis of Case Studies when deciding to adopt cloud computing architecture. How to decide if the cloud is right for your requirements. Cloud based service, applications and development platform deployment so as to improve the total cost of ownership.	10

Reference Books

1. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge University Press; 1 edition, *ISBN: 978-0521137355], 2010
2. TobyVelte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach”McGraw-Hill Osborne Media;1edition[ISBN:0071626948],2009.
3. Dimitris N.Chorafas,“Cloud Computing Strategies”CRCPress;1st edition 2010.

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Machine Learning in Cyber Security</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Knowledge of Probability and Statistics
2. Knowledge of Linear Algebra and Calculus
3. Cyber security , Network concepts , Knowledge of Computer Hardware and Software
4. Familiarity with Programming Languages and Good Programming Skills
5. Choosing Right Programming Language

COURSE OBJECTIVE: This course aims to enable learners:

1. Understand the importance of Machine learning in Cyber Security
2. Explain the use of Classification and Clustering algorithms in Cyber Security domain
3. Describe Anomaly Detection with Data and Algorithms
4. Apply Machine learning for Malware Analysis
5. Describe Machine Learning and Network Security for Network Traffic Analysis

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the importance of Machine learning in Cyber Security
CO2	Explain the use of Classification and Clustering algorithms in Cyber Security domain
CO3	Describe Anomaly Detection with Data and Algorithms
CO4	Apply Machine learning for Malware Analysis
CO5	Describe Machine Learning and Network Security for Network Traffic Analysis

Course Content

Unit No	Content	Hours
1	**UNIT-I- Why Machine Learning and Security?** Why Machine Learning and Security? Cyber threat Landscape, The Cyber Attackers Economy, A Marketplace for Hacking skills, Indirect Monetization, The Upshot. What is Machine Learning? What Machine Learning is not, Adversaries Using Machine Learning, Real-World uses of Machine Learning in Security, Spam-Fighting: An Iterative Approach, Limitations of Machine Learning in Security.	10
2	**UNIT-II – Classifying and Clustering** Machine Learning: Problems and Approaches, Machine Learning in Practice: A Worked Example, Training Algorithms to Learn : Model Families, Loss Functions, Optimization, Supervised Classification Algorithms: Logistic Regression, Decision Trees, Decision Forests, Support Vector Machines, Naive Bayes, k-Nearest Neighbors, Neural Networks	12
3	**UNIT-III- Anomaly Detection:** When to use Anomaly Detection Versus Supervised Learning, Intrusion Detection with Heuristics, Data-Driven Methods, Feature Engineering for Anomaly Detection: Host Intrusion Detection, Network Intrusion Detection, Web Application Intrusion Detection. Anomaly Detection with Data and Algorithms: Forecasting, Statistical Metrics, Goodness-of–Fit, Unsupervised Machine Learning Algorithms, and Density based Methods, Challenges of using Machine Learning in Anomaly Detection, Response and Mitigation, Practical System Design Concerns: Optimizing for Explainability, Maintainability of Anomaly Detection Systems, Integrating Human Feedback, Mitigating Adversarial Effects.	12
4	**UNIT-IV- Malware Analysis** Understanding Malware: Defining Malware Classification, Malware: Behind the scenes, Feature Generation : Data Collection, Generating Features, Feature Selection, From Features to Classification: how to get Malware Samples and Labels.	8
5	**UNIT-V- Network Traffic Analysis** Theory of Network Defense : Access Control and Authentication, Intrusion Detection, Detecting In-Network Attackers, Data-Centric Security, Honeypots, Machine Learning and Network Security : From Captures to Features, Threats in the Network, Botnets and You, Building a Predictive Model to Classify Network Attacks: Exploring the data, Data Preparation, Classification, Supervised Learning, Semi- Supervised Learning, Unsupervised Learning, Advanced Ensembling.	10

Reference Books

1. Machine Learning and Security, Clarence Chio and David Freeman, O Reilly Publications, 2018
2. Artificial Intelligence and Cyber Security, Sumeet Dua, Xian Du,CRC Press 2011
3. Machine Learning for Cyber Security Cookbook, Emmanuel Tsukerman, Packt Publications, 2020.

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Data Science</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of statistics and probability
2. Familiarity with programming concepts (preferably in Python or R)
3. Understanding of basic mathematical concepts (algebra, calculus)
4. Introductory knowledge of data structures and algorithms (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. Understand foundational data science concepts, including statistics, probability, and data preprocessing.
2. Explore NLP and data visualization to derive insights from various data types.
3. Grasp ethical considerations and best practices in data handling.
4. Apply feature selection, dimensionality reduction, and ensemble learning for model building.
5. Gain practical experience in data manipulation, plotting, and analysis using tools like ggplot2 in R.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand and apply statistical concepts, data preprocessing techniques, and attribute-oriented analysis to various data types.
CO2	Explore natural language processing techniques and data visualization principles to enhance data understanding.
CO3	Identify and implement ethical practices in data science, ensuring data privacy, security, and ethical reporting.
CO4	Apply feature selection, dimensionality reduction, and ensemble learning techniques to improve model performance.
CO5	Develop proficiency in data manipulation and visualization using R, including basic plotting and handling external data files.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to Data Science** Brief introduction to data science, Descriptive statistics, Probability and distributions (mean, variance, covariance, covariance matrix), Hypothesis testing. Facets of Data: Structured and unstructured data, natural language, machine-generated data, network data, audio, images, and video streaming data. Data Preprocessing: Data cleaning, transformation, and reduction, Attribute-oriented analysis (Attribute generalization, relevance, class comparison).	12
2	**UNIT II - NLP and Data Visualization** Introduction to NLP: Text processing and sentiment analysis. Data Visualization: Importance in data science, Exploratory data analysis, visualization tools, developing a visualization aesthetic (maximizing data link ratio, proper scaling and labeling, effective use of color, and shading).	10
3	**UNIT III - Ethics and Best Practices in Data Science** Ethical Considerations: Data privacy and security, ethical data handling, and reporting. Best Practices: Ensuring responsible data use and adhering to legal frameworks.	10
4	**UNIT IV - Feature Selection and Ensemble Learning** Feature Selection: Filters, Wrappers, Dimensionality reduction (PCA and LDA). Ensemble Learning: Bagging, Boosting, Gradient Boosting, Random Forest, Adaptive Boosting.	10
5	**UNIT V - Data Manipulation and Plotting with R** Basic Plotting: Using plot with coordinate vectors, graphical parameters, titles, and axis labels, setting appearance constants with Geoms, adding points, lines, and text to existing plots. Using ggplot2: Quick plots with qplot, reading and writing files, R-ready data sets, contributed data sets, reading external data files, writing data files and plots, ad hoc object read/write operations.	10

Reference Books

1. "Data Science from Scratch: First Principles with Python" by Joel Grus (2nd Edition, 2019)
2. "Python for Data Analysis" by Wes McKinney (3rd Edition, 2022)
3. "Practical Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce (2nd Edition, 2020)

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Digital Image Processing</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 0 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Knowledge of Probability and Statistics
2. Knowledge of Linear Algebra and Calculus
3. Basic Programming Skills (C++ , Math Lab)
4. Knowledge of Signals & Systems

COURSE OBJECTIVE: This course aims to enable learners:

1. Understand the fundamental concepts of a digital image processing system.
2. Analyze images in the frequency domain using various transforms
3. Evaluate the techniques for image enhancement and image restoration.
4. Categorize various compression techniques and Interpret Image Compression standards.
5. Interpret image segmentation and representation techniques.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Review the fundamental concepts of a digital image processing system
CO2	Analyze images in the frequency domain using various transforms
CO3	Evaluate the techniques for image enhancement and image restoration.
CO4	Categorize various compression techniques and Interpret Image Compression standards.
CO5	Interpret image segmentation and representation techniques.

Course Content

Unit No	Content	Hours
1	**UNIT-I- Introduction to digital image processing:** Introduction to digital image processing, Stages, Application areas, components, electromagnetic spectrum, image sensing and acquisition, image sampling and quantization, relationship between pixels, Enhancement in spatial domain: Intensity transformation functions.	12
2	**UNIT-II – Spatial filtering, Frequency domain Enhancement** Spatial filtering, Frequency domain enhancement: DiscreteFourier transform (DFT) properties of the 2D discrete Fourier transform, filtering in the frequency domain, Introduction to Color image processing.	10
3	**UNIT-III- Segmentation** Segmentation -Intensity based – point, line and edge. Region based – Boundaries, region growing, Thresholding, splitting and merging, segmentation by morphological watersheds, the use of motion in segmentation.	10
4	**UNIT-IV- Morphological operations** Morphological operations: Preliminaries, opening and closing, the hit-or-miss transformation, some basic morphological algorithms, Gray-scale images. Image representation. Some applications: Document image processing, Biometrics, robot vision and medical applications.	10
5	**UNIT-V–Image Compression** Introduction, coding Redundancy, Inter-pixel redundancy, Image compression model, Lossy and Lossless compression, Huffman Coding, Arithmetic Coding, LZW coding, Transform Coding, Sub-image size selection, blocking, DCT implementation using FFT, Run length coding.	10

References Books

1. R.C.Gonzalez, R.E.Woods, DigitalImageProcessing,3-red.Prentice Hall,Pearson publication.
2. Anil K Jain, Digital Image Processing, PHI Publication
3. Milan Sonka, Image Processing, Analysis, and Machine Vision, 3rd Edition, CL Engineering(2013)

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Internet of Things</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of computer networks and communication protocols
2. Understanding of basic programming concepts (preferably in Python or C)
3. Familiarity with embedded systems or microcontroller programming (Recommended)
4. Introductory knowledge of cloud computing and data analytics.

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamental concepts, architecture, and protocols of the Internet of Things (IoT).
2. To explore the design methodology and development tools required to create IoT solutions.
3. To gain insights into data analytics platforms and techniques relevant to IoT, including real-time data processing and big data integration.
4. To examine domain-specific IoT applications, including smart cities, industrial IoT, and healthcare.
5. To identify and address real-world design constraints in IoT implementations, including hardware limitations and integration challenges.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the basic concepts, architecture, and enabling technologies of IoT, and differentiate between IoT and M2M communication.
CO2	Apply IoT platform design methodologies and utilize relevant tools and technologies, including Python and cloud services, for IoT development.
CO3	Analyze and implement data analytics techniques for IoT, including batch, real-time processing, and big data platforms.
CO4	Explore and apply IoT solutions across various domains such as smart cities, healthcare, and industrial automation.
CO5	Tackle IoT design constraints like hardware, data representation, and system integration.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to IoT** Definition and characteristics of IoT, Physical design of IoT (devices, sensors, actuators), Logical design of IoT (communication models, APIs), IoT enabling technologies (wireless sensor networks, cloud computing). M2M and IoT: Overview of M2M (Machine-to-Machine communication) and key differences between IoT and M2M. IoT Protocols: Introduction to key IoT communication protocols such as IEEE 802.15.4, Zigbee, and 6LoWPAN.	12
2	**UNIT II - IoT Platforms and Development** IoT Platform Design: Fundamentals of designing IoT platforms, including hardware and software components. IoT Development Tools: Overview of Python packages and tools (e.g., Arduino, Raspberry Pi) for IoT development. IoT and Cloud: Basic understanding of integrating IoT with cloud services, including IoT physical devices and endpoints.	10
3	**UNIT III - Data Analytics for IoT** Data Analytics Fundamentals: Introduction to data analytics in IoT, focusing on batch and real-time data processing. Big Data and IoT: Overview of big data platforms like Hadoop and Spark, and their applications in IoT. Fog Computing: Introduction to fog computing as a platform for IoT and real-time analytics.	10
4	**UNIT IV - Domain-Specific IoTs** Selected IoT applications in specific domains: **Home Automation:** Basics of smart home systems. **Smart Cities**: IoT applications in urban planning and infrastructure. **Healthcare:** Introduction to IoT in healthcare, including health information systems and wearable devices.	10
5	**UNIT V - Real World IoT Design Constraints** Design Constraints: Overview of key design challenges in IoT, including hardware limitations, data representation, and security. Case Studies: Simplified case studies on IoT in industrial automation and building automation.	10

Reference Books

1. "Internet of Things: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti (2014)
2. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, and Jerome Henry (2017)
3. "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz (2016)

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Mobile Application Development</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of programming (preferably in Dart, Java, or any object-oriented programming language)
2. Familiarity with basic concepts of software development, including control flow, data types, and OOP principles
3. Understanding of basic web technologies (HTML, CSS, JavaScript) is helpful.

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamentals of Dart programming and its application in mobile development.
2. To gain proficiency in using Flutter for building cross-platform mobile applications.
3. To explore key Flutter widgets and layout techniques for designing responsive user interfaces.
4. To learn how to integrate RESTful APIs and manage state in Flutter applications.
5. To gain practical knowledge of deploying mobile applications to Play Store and App Store, including Firebase integration and basic socket programming.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Demonstrate proficiency in Dart programming, including OOP concepts and basic programming constructs.
CO2	Understand and apply Flutter concepts, including the use of basic and layout widgets for building user interfaces.
CO3	Implement navigation, routing, and utilize third-party packages to enhance Flutter applications.
CO4	Integrate RESTful APIs, manage state, and apply asynchronous programming techniques in Flutter.
CO5	Understand and implement Firebase integration, basic socket programming, and the deployment process for mobile applications.

Course Content

Unit No	Content	Hours
1	**UNIT I: Introduction to Dart Programming** Dart Programming Language: Introduction, Variables, operators, standard input/output, Data types and their methods, Control flow, Key functions, OOP concepts, Basic programming examples.	8
2	**UNIT II: Flutter Basics and Widgets** Flutter Introduction: Overview of Flutter, Benefits and scope of Flutter, Applications made using Flutter, Tools used (Android Studio, VS Code, Postman) with explanations. Flutter Concepts: Basic Flutter widgets and layouts, Understanding widgets (stateless vs. stateful widgets, setState method, runApp, materialApp, and its properties), Entry point to the app. Basic Widgets: Scaffold, Text, Icon, Image, AppBar, Column, Row, Container. Layout Widgets: Padding, Align, Center, Expanded, SizedBox, Stack, ListView, GridView. Handling User Inputs: TextField, Button, Switch, Checkbox, Slider. Other Flutter Widgets.	14
3	**UNIT III: Navigation, Routing, and Packages in Flutter** Navigation and Routing in Flutter: Basic navigation using Navigator push and pop, Named routes. Responsive Designs in Flutter: Importance, MediaQuery. Basic Packages in Flutter: Understanding packages, How to use packages in Flutter with examples (pub.dev), Basic packages (image_picker, url_launcher, loading_animation_widget, flutter_svg, http, intl, google_maps_flutter, geolocator).	10
4	**UNIT IV: RESTful API Integration and State Management** RESTful API Integration: Understanding RESTful APIs, Making HTTP requests (http package, introduction to dio), Handling responses (JSON parsing, Error handling), Asynchronous programming, Data modeling (Purpose, Creating model classes, Serialization and deserialization), Shared Preferences. State Management in Flutter: Overview of state management, Purpose and benefits, Overview of common methods (Provider, Riverpod, BLoC).	10
5	**UNIT V: Firebase Integration, Socket Programming, and Deployment** Firebase Integration in Flutter: Introduction to Firebase, Project creation, Firebase notification handling, Overview of Firebase Crashlytics and Firebase Analytics. Socket Programming: Basic introduction. Deployment Procedure: Overview of deploying applications on Play Store and App Store	10

Reference Books

1. "Flutter Complete Reference 2.0" by Alberto Miola (2023).
2. "Flutter Cookbook: 100+ Step-by-Step Recipes for Building Cross-Platform, Professional-Grade Apps with Flutter 3.10.x and Dart 3.x, 2nd Edition" by Simone Alessandria (2023).
3. "Flutter for Beginners - Third Edition" by Thomas Bailey and Alessandro Biessek (2023).

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="9">Ethical Hacking</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(1 : 1 : 2)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Understanding of Computer Hardware and Software
2. Familiarity with Operating Systems
3. Familiarity of Computer networks and various protocols with their applications
4. Linux and Database Expertise
5. Cryptography Expertise
6. Reverse Engineering

COURSE OBJECTIVE: This course aims to enable learners:

1. Introduces the concepts of Ethical Hacking
2. Gives the students the opportunity to learn about different tools and techniques in Ethical hacking and security Practically
3. Apply Ethical hacking tools to perform various activities.
4. Demonstrate how Intruders escalate privileges

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the concepts of ethical hacking
CO2	Understand the concepts of System hacking
CO3	Perform TCP/IP and Port scanning
CO4	Identify desktop and server OS vulnerabilities
CO5	Describe network protection systems

Course Content

Unit No	Content	Hours
1	**UNIT-I- Introduction** Ethical hacking process, Hackers behavior & mindset, Maintaining Anonymity, Hacking Methodology, Information Gathering, Active and Passive Sniffing, Physical security vulnerabilities and countermeasures. Internal and External testing. Preparation of Ethical Hacking and Penetration Test Reports and Documents.	12
2	**UNIT-II- Social Engineering attacks and countermeasures** Social Engineering attacks and countermeasures. Password attacks, Privilege Escalation and Executing Applications, Network Infrastructure Vulnerabilities, IPspoofing, DNSspoofing, Wireless Hacking: Wireless footprint, Wireless scanning and enumeration, Gaining access (hacking 802.11), WEP, WPA, WPA2.	10
3	**UNIT-III- DoS attacks** DoS attacks. Web server and application vulnerabilities, SQL injection attacks, Vulnerability Analysis and Reverse Engineering, Buffer overflow attacks.Client-side browser exploits, Exploiting Windows Access Control Model for Local Elevation Privilege. Exploiting vulnerabilities in Mobile Application.	10
4	**UNIT-IV- System Hacking** Cracking passwords, escalating privileges, executing applications, hiding files and covering tracks – Steganography application and classification, tools, methods/attacks on Steganography, Steganography detection tools. Practical: Foot Printing & Reconnaissance, Scanning Networks, Enumeration, System Hacking	10
5	**UNIT-V-Introduction to Metasploit** Introduction to Metasploit: Metasploit framework, Metasploit Console, Payloads, Metrpreter, Introduction to Armitage, Installing and using Kali Linux Distribution, Introduction to penetration testing tools in Kali Linux. Case Studies of recent vulnerabilities and attacks.	10

Reference Books

1. Council,Ec.,Computer Forensics:Investigating Network Intrusions and Cybercrime, Cengage Learning, SecondEdition,2010
2. Baloch,R. Ethical Hacking and Penetration Testing Guide, CRCPress,2015.
3. Beaver,K., Hacking for Dummies,3rd ed. JohnWiley&sons., 2013.
4. McClureS.,Scambray.,and Kurtz G, Hacking Exposed.TataMcGraw-Hill Education,6theEdition,2009
5. International Council of E-Commerce Consultants by Learning, Penetration Testing Network and Perimeter Testing Ec-Council/ Certified Security Analyst Vol. 3 of Penetration Testing, Cengage Learning, 2010

SEMESTER III

<table>
<tr><td>Course Title</td><td colspan="8">Mobile Application Development Laboratory</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(2 : 1 : 1)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic programming skills in Dart, Java, or any object-oriented language.
2. Familiarity with software development concepts such as control flow, data types, and OOP principles.
3. Basic understanding of mobile development and app deployment processes.

COURSE OBJECTIVE: This course aims to enable learners:

1. Develop hands-on proficiency in building mobile applications using Dart and Flutter.
2. Implement practical knowledge of Flutter widgets, layouts, and user interface design.
3. Practice integrating RESTful APIs and managing application state in a mobile environment.
4. Gain experience in deploying mobile applications on Android and iOS platforms.
5. Explore advanced features like Firebase integration, local storage, and real-time communication.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Demonstrate the ability to set up a Flutter development environment and create basic Flutter applications.
CO2	Build responsive user interfaces using various Flutter widgets and layout techniques.
CO3	Implement navigation, routing, and utilize third-party libraries to extend app functionality.
CO4	Integrate RESTful APIs and manage state in Flutter applications using Provider or Bloc.
CO5	Develop, test, and deploy mobile applications to Play Store and App Store, including Firebase integration and basic socket programming.

Mini Project

1. **Student Management System**
   Create a program that manages student information using a class-based structure. It will include methods to add students, calculate their average scores, and display the information. This program demonstrates the use of classes, objects, functions, data types, and control flow statements.

2. **Fibonacci Series Generator**
   Create a program that generates a Fibonacci series based on user input. The program will prompt the user to specify how many terms of the Fibonacci sequence they want to see and will then calculate and display those terms.

3. **Palindrome Checker**
   Write a Dart program that checks if a given string is a palindrome (reads the same backward as forward). Ignore case and non-alphanumeric characters.

4. **Word Frequency Counter using Dart**
   Develop a program that counts the frequency of each word in a given text. The program will take a text input from the user and output each word along with its corresponding frequency. This will demonstrate string manipulation, the use of maps, and data processing.

5. **Binary Search Implementation**
   Implement a binary search algorithm to find the position of a target value within a sorted array. The program will prompt the user for a sorted list of numbers and the target number to search for, and it will return the index of the target if found, or indicate if the target is not present.

6. **Quick Sort Implementation**
   Create a program that implements the Quick Sort algorithm to sort an array of integers. The program will accept an unsorted list from the user and output the sorted list. This will showcase algorithmic thinking, recursion, and array manipulation.

7. **Shopping Cart Management System**
   Develop a shopping cart management system using Dart that utilizes a List<Map> to store product information. The program should enable users to perform the following operations:
   a. Add Products: Allow users to input new product details, including name, price, and quantity, and store them in a list.
   b. View Products: Display all available products in the inventory.
   c. Search Products: Implement functionality to search for products by name and display matching results.
   d. Sort Products: Enable sorting of the product list by product name.
   e. Add Products to Cart: Allow users to select products from the inventory and add them to their shopping cart.
   f. Delete Products: Provide functionality to remove products from the inventory.
   g. View Cart: Display the contents of the user's shopping cart, including product details.

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Major Project</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">128</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">16</td><td>Credits</td><td>8</td></tr>
<tr><td>(L : T : P)</td><td>(0 : 2 : 6)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">Continuous Evaluation</td><td></td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td></td><td></td><td></td><td></td></tr>
</table>

COURSE PREREQUISITE: NIL

COURSE OBJECTIVE: This course aims to enable learners:

1. Provide students with in-depth knowledge in their chosen area of technology for project development.
2. Enable students to identify, discuss, and justify the technical aspects of their project using a comprehensive and systematic approach.
3. Facilitate the design and development of web-based or research-oriented projects focused on software systems.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Apply knowledge gained both within and beyond the curriculum to solve real-world problems.
CO2	Formulate and investigate current, trending problems using appropriate research methodologies.
CO3	Apply software engineering principles to design and implement solutions for real-world challenges.

Project Guidelines

1. Introduction

The Department of Computer Applications aims to impart value-based technical education with a focus on computational skills, analysis, design, and project management. The project work is a major component in most professional programs and serves as both a fulfillment of MCA requirements and a demonstration of students' skills, abilities, and areas of specialization.

This project provides students with the opportunity to showcase the practical implementation of the knowledge gained throughout the previous five semesters, integrating both theoretical and practical aspects of computer applications. The project work must include the Software Development Life Cycle (SDLC), proper documentation, and a demonstration, all to be completed within the stipulated 16-week period.

2. Aim

The primary aim of the project is to familiarize students with the process of implementing computer-based solutions and applications using their domain knowledge. It provides real-time solutions to problems in industry, academia, and research.

3. Objective

The project is structured to help students develop their ability to apply theoretical knowledge and practical tools/techniques. It prepares them to handle real-world challenges through systematic and comprehensive project implementation.

4. Outcome

Upon completion of the project, students will be capable of:

- Understanding and applying the Systems Development Life Cycle (SDLC) effectively.
- Identifying and documenting system requirements.
- Using data-gathering techniques and performing in-depth analysis.
- Conducting feasibility studies, including cost-benefit, technical, time, and operational feasibility.
- Designing and developing system architectures and data flow diagrams.
- Evaluating and using structured methods like decision tables and decision trees.
- Creating and evaluating graphical tools such as system flowcharts and state transition diagrams.
- Defining software and hardware requirements
- Planning the system design phase of the SDLC.
- Distinguishing between logical and physical design requirements.
- Designing and evaluating system inputs and outputs.
- Estimating storage requirements and planning data management.
- Explaining the file update processes based on standard file organizations.

- Constructing and evaluating Entity-Relationship (ER) diagrams for RDBMS projects.
- Performing normalization for unnormalized tables in RDBMS projects.

5. Project Selection Process

Students can implement projects in two streams: **Application Stream** or **Research Stream.**

1. Application Stream

Projects in the Application Stream focus on developing software solutions to accomplish specific tasks or address domain-specific requirements. Students must thoroughly understand the project's applicability and develop the application with the necessary features and functionalities. Projects can be undertaken either in college or within an industry setting. Students may select from the following application domains (though not limited to these):

- Database, Data Mining, and Data Warehousing Applications
- Networking Applications (e.g., Mobile, SAN, Security, MANet, etc.)
- Business/Enterprise Applications (e.g., Supply Chain, ERP, CRM)
- Web-based Applications
- Multimedia Applications
- Software Engineering Applications
- Grid and Cloud Computing Applications
- Parallel and Distributed Computing Applications
- E-learning and E-commerce Applications
- Gaming Applications
- Hardware Driver Applications
- Mobile Applications
- Healthcare Applications
- Banking and Finance Applications
- Insurance Applications

2. Research Stream

Projects in the Research Stream aim to solve significant problems relevant to various domains. Students must clearly define the problem formulation, ensuring it is well understood and pertinent to a specific field or audience. The selected research area can include the following or any related field where software plays a critical role:

- Public Health, Epidemiology & Health Services
- Biomedical Research
- Clinical Sciences and Health Practice
- Neurosciences & Behavioral Sciences
- Cloud Services
- Big Data
- Data Science
- Image Processing
- Internet of Things (IoT)
- Any Engineering Topics Involving Software Solutions

A key early step in any research project is the literature survey. Conducting a thorough literature survey helps identify prior research and situates the current project within the relevant conceptual and theoretical framework.

6. Problem Formulation

Students are encouraged to follow these steps when formulating the problem for their project. Guides should ensure students adhere to the following:

1. **Tailoring the Idea:**
   Adapt the project idea to fit the targeted topic or work programme.
2. **Objectives and Applications:**
   Clearly define the objectives and practical applications of the project, specifying the goals of the selected topic.
3. **Project Implementation Phases**:
   Outline the phases of project implementation and the structure required. Identify the necessary activities to achieve the objectives and the expertise/knowledge needed.
4. **End Results and Impact:**
   Describe the expected project outputs and their potential impact.
5. **Innovation and Added Value:**
   Highlight the innovative aspects and added value of your project.

6. **Project Concept Questions:**

   Ensure the project concept addresses these key questions:

   - **Why**? – The problem or need being addressed.
   - **What**? – The objectives and scope of the project.
   - **How?** – The technical approach to solving the problem.
   - **Who?** – The intended users or beneficiaries.
   - **Where?** – The domain or environment where the solution will be applicable.

Statement of Problem: The "Why?"

- Provide a concise summary of the original problem statement.
- Include background information such as:
  - A brief description of the company (if applicable) and the relevance or importance of the problem.
  - Literature Review: Include credible sources and previous related work by others.
  - Patent Search: If relevant, conduct a patent search on related problems.

Objectives: The "What?"

- Translate the customer's quantitative and qualitative needs into clear, objective design specifications.
- Define the scope of work and clearly state the specific objectives of the project.

Technical Approach: The "How?"

- Provide an initial design concept for solving the problem, even if the solution details are still in progress.
- Explain your first approach or ideas on how to address the problem and present possible design concepts.

Intended Users: The "Who?"

- Identify the intended users or beneficiaries of the system being developed.

Environment: The "Where?"

- Specify the domain or area where the application or work will be useful and applicable.

7. Evaluation Process

The project duration is 16 weeks: 14 weeks for project development and 2 weeks for preparation of the project4 report. Students are required to maintain a project diary, regularly consult with their internal guide, update the status of their work, and have it reviewed weekly. Failure to provide regular updates may result in forfeiture of the project work claim.

The evaluation process consists of the following steps:

1. Project Screening and Finalization

- Student Activity: Submission of the project synopsis for review and finalization.

2. Evaluation of System Study and Analysis

- Student Activity: Submission of the System Requirement Specification (SRS) document.

3. Evaluation of Project Design

- Student Activity: Submission of the project design document.

4. Evaluation of Coding, Testing, and Implementation

- Student Activity: Submission of the draft final project report.

5. Final Report Submission

- Student Activity: Submission of the final project report.

Project Screening

Student projects will be screened based on the following criteria:

a) Relevance to current technology

b) Sufficient workload: Ensuring the project involves 16 weeks of man-hours.

c) Place of execution: Whether the project is conducted in an industry, research organization, or college setting.

d) Relevance to Computer Science and Applications

e) Student's understanding of the project work.

Finalization of Synopsis

The finalization of the synopsis follows these steps:

a) Finalization is based on the evaluation at the previous stage.

b) Recommendations from the panel of department staff members are incorporated.

c) Verification of whether previous evaluation suggestions have been properly implemented.

The format of the synopsis can be referred to in **Annexure-I**.

Evaluation of System Study and Requirement Analysis

The evaluation in this phase will be based on the following criteria:

a) **Understanding of the Problem:** Whether the student has comprehended the problem in-depth.

b) **Scope and Objectives:** Whether the scope and objectives of the project are clearly and accurately defined.

c) **System Study:** Whether the student has thoroughly studied and fully understands the system.

d) **Functional and Non-Functional Requirements**: Whether the functional requirements and non-functional requirements are properly defined. The SRS document can be referred to in **Annexure-II.**

Evaluation of Project Design

At this stage, the project design will be evaluated based on the following criteria:

a) Architectural Design

b) Detailed Design

c) Database Design

d) External Interface Design

e) Algorithm Design

The design document format can be referred to in **Annexure-III.**

Evaluation of Coding, Testing, and Implementation

The evaluation at this stage will be based on the following criteria:

a) Pseudo Code

b) Test Case Generation

c) Actual Testing

d) User's Manual

e) Implementation Strategy

f) Draft Report Submission

g)Verification against SRS: Whether the implementation meets the requirements specified in the SRS document.

h) Demo: Demonstration of the final working model of the system.

The final report format should follow the guidelines in **Annexure-VI.**

Annexure I: Format of the Format of the Synopsis

Application stream

- Title of the Project.
- Problem statement.
- Objective.
- Scope of the project.
- Brief description.
- Project Category (RDBMS/OOPS/Networking/E-learning. Etc.).
- Software requirements.
- Hardware requirements

Research stream

- Title of the Project.
- Problem statement.
- Objective of the project.
- Related work done(list of literature)
- Brief description of proposed work.
- Software requirements.

Note: Students doing internship programs outside the college need to submit the following details.

- Name and address of the company
- External guide name and designation
- Email and contact number of the external guide.

Annexure II: Format of the SRS document

Application stream

- ☐ **Introduction**
  - ☐ Scope of SRS document
  - ☐ Definitions, acronyms and abbreviations
    - ☐ Definitions, acronyms and abbreviations used
  - ☐ References
  - ☐ Overview
- ☐ **Overall description**
  - ☐ Product perspective
    - ☐ About your project work
  - ☐ Product functions
  - ☐ User characteristics
    - ☐ Characteristics of the intended user
  - ☐ General constraints
  - ☐ Assumptions and Dependencies
- ☐ **Specific requirements**
  - ☐ External Requirements
    - ☐ (User Interface/Hardware/Software)
  - ☐ Functional Requirements
  - ☐ Non-functional requirements
  - ☐ Other requirements.
- ☐ **Architectural overview**
- ☐ **Data flow diagrams**
- ☐ **Design constraints**

Research stream

- ☐ Introduction
- ☐ Detailed literature review of related work
- ☐ Detailed analysis of the proposal work
- ☐ Conclusion

Annexure III: Format of Design Document

Application Stream

- ☐ **Introduction**
  - ☐ Scope of the design document
  - ☐ Definitions, acronyms and abbreviations references
  - ☐ References
  - ☐ Overview
- ☐ **System Architecture**
  - ☐ Architectural design
  - ☐ User interface design
  - ☐ Forms design
  - ☐ Database or file design
  - ☐ Reports design
  - ☐ Workflows and algorithms design
  - ☐ Detailed design of DFDs.
  - ☐ Activity diagrams
  - ☐ Use case diagrams
  - ☐ Sequence diagrams
- ☐ **Component Interfaces**
  - ☐ Component design description
  - ☐ Software requirements mapping Critical requirements

Research stream

- ☐ **Introduction**
- ☐ **Algorithms/logic used in the related work.**
- ☐ **Algorithms/logic steps involved in the proposed method.**
- ☐ **Conclusion**

Annexure IV: Coding detail

Application and Research stream:

- ☐ Introduction coding convention
- ☐ Pseudo code of the important algorithms of the project

Annexure V: Testing report

Application and Research stream:

- Introduction
- Test process done for the project
- Test cases

Test Case No	Positive Scenario	Required Input	Expected Output	Actual Output	Test Pass / Fail

Test Case No	Negative Scenario	Required Input	Expected Output	Actual Output	Test Pass / Fail

Annexure VI: Report format

TITLE PAGE, CERTIFICATE by the Department, CERTIFICATE by the EXTERNAL ORGANIZATION DECLARATION, ACKNOWLEDGEMENTS ABSTRACT, CONTENT PAGE, LIST OF FIGURES, LIST OF TABLES

CHAPTER 1

- ☐ Introduction to the Topic of the Project Problem definition and Premises Objectives of the Study / Project Methodology

Organization of the Dissertation

CHAPTER 2

- History of the Organization (If carried out in an organization)

CHAPTER 3

- Justification for doing this project

CHAPTER 4

- Requirements analysis and specifications document. Refer Annexure II

CHAPTER 5

- Design
- Document Refer
- Annexure III

CHAPTER 6

- Coding
- Refer Annexure IV

CHAPTER 7

- Testing
- Refer Annexure V

CHAPTER 8

- Experimental results/findings.
- Conclusion
- Scope for future work (If any)
- Limitations of the study (If any)
- References [Papers, books, thesis web references (urls)]
- List of Publications (if any)
- Attach a hard copy of the publication.

Note

- Page numbers from “Abstract” to “List of Tables” to be given in Roman Numerals
- Page numbers to be given in numerals from Chapter 1 to Chapter 8.
- All pages should contain a footer indicating the Department of Master of Computer Applications, St. Philomena’s College (Autonomous), Mysuru and header indicated the title of the project and Month-Year of submission.

Font size should be:

- 16 bold for chapter no and chapter title.
- 14 bold times new roman for sub headings.
- 12 bold times new roman for sub-sub headings.
- 12 times new roman normal for running text.
- Table numbers, figure numbers and photograph numbers should be according to the Chapter with appropriate title. Caption should contain figure no and title below the figure. Table no and title should be above the table.
- Spacing should be 1.5 between lines and 2.0 between paragraphs.
- Cross references should be clearly indicated and proper citation should be given wherever references have been made.
- Project report should be hard bound in Pink color to be submitted. Make 3 Copies. Submit one to the Department, one to the internal guide and one to yourself.

Marks allotment in evaluation process

- The evaluation process is done in four stages.

First Evaluation

- Evaluation is conducted at the time of acceptance of the project i.e., during synopsis submission. Students need to prepare and present the selected project before the panel. Evaluation carries 20 marks. The main components to be evaluated are,

Sl. No	Evaluation of Synopsis	Panel Marks	Guide Marks
1	Role of Innovation / Creativity, Research Element, challenges in the Problem.	**2**	**2**
2	Literature survey / Existing system analysis.	**2**	**2**
3	Clarity about the problem	**2**	**2**
4	Attendance	**2**	**2**
5	Question and Answer	**2**	**2**
Note: If total marks is less than 50% of the maximum mark, then after a week students need to re- present the project before the panel for reconsideration of the project or may propose a different project. After the panel clearance students can start the project.			

Second Evaluation

Evaluation is conducted after one month of submission of the synopsis. Students need to prepare and present on the methodology, analysis and possible outcome of the problem in front of the panel. Evaluation carries 20 marks. The main components to be evaluated are, understanding of the problem, system study and analysis done.

Sl. No	Evaluation of Synopsis	Panel Marks	Guide Marks
1	Analysis of the Problem and understanding	**2**	**2**
2	Specification of the requirements	**2**	**2**
3	Progress of the project	**2**	**2**
4	Attendance	**2**	**2**
5	Question and Answer	**2**	**2**

Third Evaluation

Evaluation is conducted after one month of second evaluation. Students need to present the design concepts of the project in front of the panel. Evaluation carries 20 marks. The main components to be evaluated are,

Sl. No	Evaluation of Synopsis	Panel Marks	Guide Marks
1	Architectural Design	**2**	**2**
2	Detailed Design	**2**	**2**
3	Work implementation Status	**2**	**2**
4	Attendance	**2**	**2**
5	Question and Answer	**2**	**2**

Final Evaluation

Evaluation is conducted after one month of third evaluation to check coding, testing and implementation stages. Students need to prepare a draft final report with a demo of the project for the presentation.

Sl. No	Evaluation of Synopsis	Panel Marks	Guide Marks
1	Coding and implementation efficiency	**4**	**4**
2	Work completion	**4**	**4**
3	Testing efficiency	**4**	**4**
4	Attendance	**4**	**4**
5	Question and Answer	**4**	**4**

Overall Marks Distribution		
Total Internal Assessment Marks	**Panel Member**	**Guide**
100	**50**	**50**

Total Marks Distribution

Project	Internal Assessment (Guide (50%) + Panel (50%))	Report Evaluation		Report Evaluation	Viva Voce	Total Marks
		Internal (I)	External (E)	Avg. (Internal and External)		
Major (Max)	100	100	100	100	50	250
Min	50				25	125

Credits distribution for the evaluation for major project

Evaluation Steps	Credits
First Evaluation	1
Second Evaluation	1
Third Evaluation	1
Final Evaluation	1
Report Evaluation (Average of internal and external examiners)	3
Viva Voce	1
Total	**8**

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Research Methodology</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic Understanding of Computer Hardware and Software
2. Familiarity with Operating Systems
3. Familiarity of Computer networks and various protocols with their applications

COURSE OBJECTIVE: This course aims to enable learners:

1. Introduce various technologies for conducting research.
2. Choose an appropriate research design for the chosen problem.
3. Explain the art of interpretation and the art of writing research reports.
4. Explore the various forms of intellectual property, its relevance and business impact in the changing global business environment.
5. Discuss leading International Instruments concerning Intellectual Property Rights

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand various technologies for conducting research
CO2	Choose an appropriate research design for the chosen problem
CO3	Explain the art of interpretation and the art of writing research reports.
CO4	Explore the various forms of intellectual property, its relevance and business impact in the changing global business environment.
CO5	Discuss leading International Instruments concerning Intellectual Property Rights

Unit No	Course Content	Hours
1	**UNIT-I- Research Methodology** Introduction, Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, and Problems Encountered by Researchers in India. Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, An Illustration.	12
2	**UNIT-II- Reviewing the literature** Place of the literature review in research, Bringing clarity and focus to your research problem, Improving research methodology, Broadening knowledge base in research area, Enabling contextual findings, How to review the literature, searching the existing literature, reviewing the selected literature, Developing a theoretical framework, Developing a conceptual framework, Writing about the literature reviewed. Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs, Important Experimental Designs.	10
3	**UNIT-III- Design of Sampling** Introduction, Sample Design, Sampling and Non-sampling Errors, Sample Survey versus Census Survey, Types of Sampling Designs. Measurement and Scaling: Qualitative and Quantitative Data, Classifications of Measurement Scales, Goodness of Measurement Scales, Sources of Error in Measurement Tools, Scaling, Scale Classification Bases, Scaling Technics, Multidimensional Scaling, Deciding the Scale. Data Collection: Experimental and Surveys, Collection of Primary Data, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Case Study Method.	10
4	**UNIT-IV : Testing of Hypotheses** Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Testing of Hypothesis, Test Statistics and Critical Region, Critical Value and Decision Rule, Procedure for Hypothesis Testing, Hypothesis Testing for Mean, Proportion, Variance, for Difference of Two Mean, for Difference of Two Proportions, for Difference of Two Variances, P-Value approach, Power of Test, Limitations of the Tests of Hypothesis. Chi-square Test: Test of Difference of more than Two Proportions, Test of Independence of Attributes, Test of Goodness of Fit, Cautions in Using Chi Square Tests	10
5	**UNIT-V: Methods of Data Collection and Interpretation, Report Writing:** Introduction, experiments and surveys, Collection of Primary Data: Observation Method, Interview Method, Collection of Data through Questionnaires, Difference between Questionnaires and Schedule guidelines for constructing questionnaire/schedule, Some Other Methods of Data Collection, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Case Study Method. Meaning of Interpretation, Why Interpretation?, Technique of Interpretation, Precaution in Interpretation,Significance of Report Writing, Different Steps in Writing Report, Layout of the Research Report, Types of Reports, Oral Presentation, Mechanics of Writing a Research Report.	10

Reference Books

1. Research Methodology: Methods and Techniques, C.R. Kothari, Gaurav Garg, New Age International, 4th Edition, 2018.
2. Research Methodology a step-by-step guide for beginners.RanjitKumar, SAGE Publications, 3rd Edition, 2011.
3. Research Methods: the concise knowledge base, Trochim, Atomic Dog Publishing, 2005.
4. Conducting Research Literature Reviews: From the Internet to Paper, Fink A, Sage Publications, 2009.

Web links and Video Lectures (e-Resources):

1. https://www.youtube.com/watch?v=A7oioOJ4g0Y&list=PLVf5enqoJ-yVQ2RXUl6mCfLPf3J_JUfoc

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">BlockChain Technologies</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of cryptography and networking principles
2. Familiarity with programming concepts (preferably in Python or C++)
3. Introductory knowledge of distributed systems (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the fundamental concepts of blockchain technology and its underlying cryptographic principles.
2. To explore the structure and functioning of Bitcoin and other cryptocurrencies, including the mining process and decentralized networks.
3. To examine consensus models and their importance in blockchain technology.
4. To learn about smart contracts, their lifecycle, and how to interact with them using Solidity.
5. To critically analyze the limitations and misconceptions surrounding blockchain technology.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the core concepts and need for blockchain technology.
CO2	Analyze the structure and impact of Bitcoin and other cryptocurrencies.
CO3	Explore and compare different consensus models in blockchain.
CO4	Develop smart contracts using Solidity and understand blockchain forking.
CO5	Assess blockchain limitations, misconceptions, and security challenges.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to Blockchain Technology** What is blockchain? The need for distributed records, introduction to blockchain-based cryptocurrency. Blockchain Categories: Overview of permissioned and permissionless blockchains, types of blockchains (Public, Private, Hybrid). Core Concepts: Public key infrastructure, cryptographic hash functions, cryptographic nonce, transactions, ledgers, block creation, chaining blocks. Technologies in Blockchain: Brief overview of hash pointers, consensus, Byzantine fault-tolerant computing, and digital cash.	10
2	**UNIT II - Bitcoin and Cryptocurrencies** Bitcoin: Introduction, The Bitcoin Network, Bitcoin mining process, Bitcoin wallets. Key Concepts: Decentralization, hard forks, Ethereum Virtual Machine (EVM), Merkle Tree, and the double-spend problem. Introduction to Hyperledger: Overview of Hyperledger and its components like Fabric and Composer.	12
3	**UNIT III - Consensus Mechanisms** Introduction to Consensus: Why consensus is essential in blockchain. Types of Consensus: Focus on Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT). Consensus Comparison: Brief comparison of major consensus models.	10
4	**UNIT IV - Smart Contracts** Smart Contracts: Introduction, importance, and basic life cycle. Solidity Programming: Basics of Solidity, Ethereum accounts, and transactions. Forking in Blockchain: Overview of soft and hard forks, their implications in blockchain networks.	10
5	**UNIT V - Blockchain Challenges and Misconceptions** Blockchain Limitations: Immutability, 51% attack, governance issues, resource usage, and trust challenges. Misconceptions and Security: Pseudo-anonymity vs. anonymity, Zcash, zk-SNARKs, and common vulnerabilities in blockchain.	10

Reference Books

1. "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications" by Imran Bashir (4th Edition, 2023)
2. "Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher (1st Edition, 2017)
3. "Blockchain and the Law: The Rule of Code" by Primavera De Filippi and Aaron Wright (1st Edition, 2018)

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Theory of Languages and Automata</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of Discrete Mathematics, including logic, set theory, and functions.
2. Familiarity with mathematical proofs and algorithms.
3. Prior exposure to programming concepts and data structures would be beneficial.

COURSE OBJECTIVE: This course aims to enable learners:

1. Introduce the fundamental concepts of automata theory and formal languages.
2. Provide students with a strong foundation in regular languages, context-free languages, and their limitations.
3. Equip students with the ability to analyze and design finite automata, regular expressions, context-free grammars, and pushdown automata.
4. Enable students to apply formal methods and tools to solve computational problems and recognize their relevance to real-world applications.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the core concepts of automata theory and formal languages.
CO2	Analyze the power and limitations of regular languages and context-free languages.
CO3	Differentiate and manipulate formal descriptions of languages, automata, and grammars, with a focus on regular and context-free languages, finite automata, and regular expressions.
CO4	Design and evaluate finite automata and pushdown automata for various computational problems.
CO5	Apply the concepts of automata and formal languages to solve practical problems in fields like compiler design, artificial intelligence, and complexity theory.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction** Strings, alphabets, and languages. Graphs and trees. Inductive proofs, set notation, relations. Finite state systems: basic definitions. Non-deterministic finite automata (NDFA).	10
2	**UNIT II - Finite Automata and Regular Expressions** Finite automata with ϵ-moves. Regular expressions and their equivalence with finite automata. Two-way finite automata. Finite automata with output (Moore and Mealy machines). Applications of finite automata.	10
3	**UNIT III - Properties of Regular Sets** Pumping lemma for regular sets. Closure properties of regular sets (under union, intersection, etc.). Decision algorithms for regular sets (emptiness, finiteness, membership).	10
4	**UNIT IV - Context-Free Grammars** Motivation and introduction to context-free grammars (CFGs). Derivation trees and parsing. Simplification of context-free grammars. Chomsky normal form (CNF). Greibach normal form (GNF). Existence of inherently ambiguous context-free language.	12
5	**UNIT V - Pushdown Automata** Definition and introduction to pushdown automata (PDA). PDA and context-free languages. Pumping lemma for context-free languages (CFLs). Closure properties of context-free languages. Decision algorithms for context-free languages.	10

Reference Books

1. Introduction to Automata Theory, Languages, and Computation by J. E. Hopcroft, J. D. Ullman – Narosa Publishing House.

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Cloud Security</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td colspan="2" rowspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic knowledge of network security principles.
2. Familiarity with cloud computing fundamentals, including cloud service models (IaaS, PaaS, SaaS).
3. Understanding of cybersecurity threats and measures like encryption, firewalls, and intrusion detection systems.

COURSE OBJECTIVE: This course aims to enable learners:

1. Provide students with a comprehensive understanding of cloud security architectures from the perspective of cloud service providers, brokers, carriers, and auditors.
2. Equip students with the skills to implement and orchestrate secure cloud ecosystems, mitigating potential threats and risks.
3. Explore how cloud computing transforms traditional enterprise security practices compared to on-premise environments, with a focus on securing data and resources in the cloud.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Describe cloud security architectures, addressing the roles and responsibilities of providers, brokers, carriers, and auditors.
CO2	Develop and apply a methodology for orchestrating a secure cloud ecosystem, ensuring data and system protection.
CO3	Understand and evaluate how cloud computing changes traditional enterprise security compared to on-premise solutions.
CO4	Implement cloud security measures such as encryption, identity management, and network security controls to safeguard cloud resources.
CO5	Analyze the impact of compliance and regulatory frameworks on cloud security, and ensure adherence to best practices for data protection in cloud environments.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to Cloud Computing and Security** Introduction to Cloud Computing and Security: Understanding Cloud Computing - The IT Foundation for Cloud, Overview of Security Architecture, Cloud Computing Architecture: Cloud Reference Architecture, Control over Security in the Cloud Model, Cloud Deployment & Services Models, Key Examples.	10
2	**UNIT II - Cloud Computing: Security Concerns:** Security Concerns in Cloud Computing: Risk Tolerance, Legal and Regulatory Issues, Security Requirements for Cloud Architecture, Security Patterns and Architectural Elements, Cloud Security Architecture, Key Strategies for Secure Operation.	10
3	**Overview of Data Security in Cloud Computing** Overview of Data Security in Cloud Computing: Common Risks with Cloud Data Security, Data Encryption: Applications and Limits, Errors in Data Encryption, Cloud Data Security: Sensitive Data Categorization, Cloud Data Storage (Roach Motel Syndrome), Managing Risks Effectively, Overview of Security Controls, Limits of Security Controls, Best Practices in Security Monitoring.	10
4	**UNIT IV - Private Clouds** Private Clouds: Motivation and Overview, Security Implications: Shared vs. Dedicated Resources, Security Criteria for Ensuring a Private Cloud, Network Considerations, Data Center Considerations, Operational Security Considerations, Regulations, Selecting a Cloud Service Provider (CSP): Assurance and Risk Overview, Revisiting Defense-in-Depth, Additional Security Criteria.	12
5	**UNIT V - Evaluating Cloud Security** Evaluating Cloud Security: Checklists for Evaluating Cloud Security - Foundational Security, Business Considerations, Defense-in-Depth, Operational Security. Operating a Cloud: From Architecture to Secure Operations, Bootstrapping Secure Operations, Security Operations Activities, Business Continuity, Backup, and Recovery, Managing Changes in Operational Environments, Information Security Management, Vulnerability and Penetration Testing, Security Monitoring and Response.	10

Reference Books

1. Vic (J.R.) Winkler, “Securing the Cloud: Cloud Computer Security Techniques and Tactics”, Elsevier, 2011.
2. Sushil Jajodia, Krishna Kant, “Secure Cloud Computing”, Elsevier, 2014.
3. Curtis Franklin, Jr., Brian J. S. Chee, “Securing the Cloud: Security Strategies for the Ubiquitous Data Center”, CRC Press, 2019.

SEMESTER IV

Course Title	**Software Testing**						
Course Type	Soft Core	***Total Hours***	52	Hours / Week	04	***Credits***	4
						(L : T : P)	(3 : 0 : 1)
Course Code		***Evaluation***	***Internal***	C1 + C2 = 15 + 15		30 Marks	100 Marks
			External	C3	***Duration*** 03 Hours	70 Marks	

COURSE PREREQUISITE

1. Basic Understanding of Software Development process
2. Familiarity with Operating Systems and Programming Skills
3. Familiarity of Test Data and Test Case Management

COURSE OBJECTIVE: This course aims to enable learners:

1. Gain knowledge on basics of Software Testing, Test case selection and creation
2. Illustrate various perspectives of testing with examples.
3. Differentiate boundary value testing, Equivalence class testing, Decision table based testing
4. Implement Path testing and Data flow testing based on the requirements
5. Comprehend different levels of testing, Integration testing and Fault based testing

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Gain knowledge on basics of Software Testing, Test case selection and creation
CO2	Illustrate various perspectives of testing with examples.
CO3	Differentiate boundary value testing, Equivalence class testing, Decision table based testing
CO4	Implement Path testing and Data flow testing based on the requirements
CO5	Comprehend different levels of testing, Integration testing and Fault based testing

Unit No	Content	Hours
1	**UNIT-I-Basics of Software Testing** Humans, Errors and Testing, Software Quality; Requirements, Behavior and Correctness, Correctness Vs Reliability; Testing and Debugging; Test Metrics; Software and Hardware Testing; Testing and Verification; Defect Management; Execution History; Test Generation Strategies; Static Testing; Test Generation from Predicates. Basic Principles, Test case selection and Adequacy : Sensitivity, Redundancy, Restriction, Partition, Visibility and Feedback, Test Specification and cases, Adequacy Criteria, Comparing Criteria.	12
2	**UNIT-II- A perspective on Testing, Examples** Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Level of testing, Examples: Generalized pseudocode, The triangle problem, the Next Date function, The commission problem, The SATM (Simple Automation Teller Machine) problem, The currency converter, Saturn windshield wiper.	10
3	**UNIT-III- Boundary value, Equivalence class and Decision table based testing** Boundary value analysis, Robustness testing, Worst-case testing, special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for triangle problem, Next Date function and commission problem, Guidelines and observations, Decision tables, Test cases for triangle problem.	10
4	**UNIT-IV- PathTesting, Data flow testing, Levels and Integration Testing** DD Paths, Test coverage metrics, Basis path testing, guidelines and observations, Definition Use testing, Slice based testing, Guidelines and observations. Traditional view of testing levels, Alternative life cycle models, the SATM systems, separating integration and system testing, Guidelines and observations.	10
5	**UNIT-V- Fault Based Testing** Assumptions in fault-based testing, Mutation Analysis, Fault-based Adequacy Criteria; Variations on mutation Analysis; From Test case specification to Test Cases, Scaffolding, Generic vs specific Scaffolding, Test Oracles, Self checks as oracles, Capture and Replay. Agile Testing: Definition and description, how is it different from traditional testing, ten principles for testers, business-facing the test that support the testing.	10

Reference Books

1. Adithya P.Mathur“Foundations of Software Testing– Fundamental Algorithms and Techniques”,PearsonEducation India, 2011
2. MauroPezze,MichaelYoung,Softwaretesting and Analysis-Process,Principles and Techniques”, Wiley India, 2012
3. Paul C Jourgensen, “Software Testing A Craftmans Approach”, Aueredach publications, 3rd edition, 2011
4. Kshirasagara Naik, Priyadarshi Tripathy: Software Testing and Quality Assurance, Wiley India 2012
5. M.G.Limaye: Software Testing- Principles, Techniques and Tools–McGrawHill,2009
6. BrainMarick:The Craft of Software Testing, Pearson Education India,2008
7. RonPatton: Software Testing, 2nd Edition, Pearson Education, India, 2013

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">E-Commerce and E-Governance</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of information technology and internet concepts
2. Familiarity with fundamental business principles and practices
3. Introductory knowledge of web technologies and security (Recommended)

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the key concepts, technologies, and secure protocols in electronic commerce.
2. To explore consumer and business models in e-commerce, including web advertising and e-retailing.
3. To examine the principles, models, and implementation strategies of e-governance.
4. To assess e-government initiatives, with a focus on India's approach and infrastructure.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand the environment and secure technologies of electronic commerce.
CO2	Analyze e-commerce models for consumers and businesses.
CO3	Explore e-governance models and ICT’s role in implementation.
CO4	Evaluate e-government initiatives and policies in India.
CO5	Address security and privacy issues in e-commerce and e-governance.

Course Content

Unit No	Content	Hours
1	**UNIT I - Introduction to E-Commerce** Introduction to E-Commerce: Overview of the electronic commerce environment and opportunities. E-Commerce Technologies: Electronic data interchange (EDI) and web commerce basics. Secure E-Commerce: Introduction to secure transport protocols, Secure electronic payment protocol (SEPP), Secure electronic transaction (SET), and online payment systems. Payments and Security: Overview of electronic cash, Internet payment processing, MasterCard/Visa secured electronic transactions	12
2	**UNIT II - Consumer and Business-Oriented E-Commerce** Consumer-Oriented E-Commerce: E-retailing and features, Developing consumer-oriented systems, The PASS model. Business-Oriented E-Commerce: B2B commerce features, Business models, and integration strategies. Web Advertising and Publishing: Key internet advertising strategies, Business models for advertising, Revenue streams, Measuring advertising effectiveness, and basic web publishing techniques.	10
3	**UNIT III - Introduction to E-Governance** E-Governance Overview: Introduction, types, and scope of e-governance. E-Governance Models: Public, corporate, and urban e-governance; E-democracy and E-business. ICT in E-Governance: Role of ICT in e-governance, Basic infrastructure and standards, Overview of the India Development Gateway (InDG).	10
4	**UNIT IV - E-Government in India** E-Government Initiatives: Core policies, infrastructure, and HRD/training for e-governance in India. National E-Governance Strategy: Implementation approach and governance structure, Case studies of major initiatives like Mission Mode Projects.	10
5	**UNIT V - Security and Best Practices in E-Commerce and E-Governance** Security in E-Governance: Addressing privacy, security, and accessibility in e-governance. Mobile Security: Overview of security challenges in mobile e-governance. Best Practices: Implementing CRM in public services, Standards and guidelines for secure e-governance, Case studies of successful implementations.	10

Reference Books

1. "E-Commerce 2023: Business, Technology, Society" by Kenneth C. Laudon and Carol Guercio Traver (17th Edition, 2023)
2. "E-Governance: Concepts and Case Studies" by C.S.R. Prabhu (2nd Edition, 2013)
3. "E-Governance and Citizen Engagement: New Directions in Public Administration" by Sangita Dhal (1st Edition, 2021)

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Communication Skills and Professional Management</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Active Listening
2. Being Assertive
3. Using positive body language

COURSE OBJECTIVE: This course aims to enable learners:

1. Aware of their communication skills and know their potential to become successful managers.
2. About the mechanics of writing and can compose the business letters in English precisely and effectively.
3. Get exposure in drafting business proposals to meet the challenges of a competitive environment.
4. Aware of the managerial communication practices in business
5. Trained in the art of Interpersonal communication and technological advancement and social media usage in communications, with emphasis on analyzing business situations.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Communicate, interact and present the ideas to the other professionals.
CO2	Understand the importance, role and contents of soft skills through instructions, knowledge acquisition, demonstration and practice.
CO3	Have the right attitude in all behavioral aspects, and build the same through activities.
CO4	Possess right professional and social ethical values.
CO5	Trained in the art of Interpersonal communication and technological advancement and social media usage in communications, with emphasis on analyzing business situations.

Course Content

Unit No	Content	Hours
1	**UNIT-I- Concepts of Communications:** Definition, Objectives of Communication, Characteristics of Communication, Process of Communication, Forms of communication, Roles of a Manager, Communication Roadblocks and Overcoming them, Overcoming Communication Barriers, Effectiveness in Managerial Communication Case Study: - Goodwill Corporation Ltd. Role of Verbal & Non-verbal Symbols in communication : Forms of Nonverbal Communication, Interpreting Non-verbal messages, Tips for effective use of non-verbal Communication Case Study:- Charisma Corporation Case Study:- Charisma Corporation	12
2	**UNIT-II - Written communication**–writing style, importance of writing skills, book review and disadvantages over oral communication. Listening: Definition, Anatomy of poor Listening, Features of a good Listener, Meaning of EL, Types of Listening skills, strategies, Barriers to effective Listening. Spoken Communication : Oral Presentation: Planning presentation, Delivering presentation, Developing & displaying visual aids, Handling questions from the audience , Telephone, Teleconferencing, Challenges and etiquette Case Study:- The Farewell Speech	10
3	**UNIT-III- Forms of Communication in Written mode:** Written Business Communication, Basic Principles, Tips for effective writing, The Seven Cs of Letter writing, Planning steps for effective writing , Persuasive written messages , Writing Business Reports (Short & Long), Kinds of Business Letters, Tone of writing, inquiries, orders & replying to them, sales letters, Job application Letters, Writing Effective Memos, Format and Principles of writing Memos.	10
4	**UNIT- IV- Group Discussion & Interviews:** Methodology of Group, Role Functions in Group Discussions, From of Group, Characteristics of Effective Groups, Group Decision –Making , Group Conflict, Types of Non-functional Behavior, Fundamental principles of Interviewing, Types of Interviewing Questions, Important Non-Verbal Aspects, Types of Interviews, Style of Interviewing. Mock Interviews, Introduction, Greetings and Art of Conversation, Dressing and Grooming, Norms of Business Dressing. Case Study:- Career Counseling	10
5	**UNIT-V- Support by word processing systems, LOTUS, Graphics software for Professional Management.** Job applications & Resume: Identifying potential career opportunities, Planning a Targeted Resume, Preparing Resumes, Supplementing a Resume, Composing Application Messages Writing E-mail, Business Reports, Business Proposals : Effective E-mail, E-mail Etiquettes, Writing Business Reports and Proposals, Purpose of Business Reports, Meetings: Ways and Means of conducting meeting effectively, Planning a Meeting, Meeting Process, How to Lead Effective Meeting, Evaluating Meeting, Writing Agenda and Minutes of meetings , Web Conferencing Case Study:- A Special Meeting of the Executive Committee	12

Reference Books:

1. Effective Communication made simple – Rupa & Co.
2. Instrument of Communication – P Meredith
3. Basic Management skills for all–EH McGrath
4. Managerial Communication – P M Timm
5. Professional Communication Aruna Koneru Tata McGraw-Hil
6. Basic business Communication, Raymond V. Lesikar & M. E. Flatley, TMH
7. Cases will be provided from Meenakshi Raman and Prakash Singh, Business Communication, Oxford University Press

SEMESTER IV

<table>
<tr><td>Course Title</td><td colspan="8">Soft Computing</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>4</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Knowledge of Programming Skills in C , C++ , java Or Mathlab
2. Proficiency With Algorithms
3. Strong Mathematics Background
4. Critical Thinking and Problem Solving Skills

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic principles, techniques, and applications of soft computing.
2. To acquire knowledge on Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithms.
3. To learn the mathematical basis for carrying out the optimization associated with Neural Network learning.
4. To Identify and select a suitable Soft Computing technology to solve the problem

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Explain the fundamental concepts of fuzzy logic, neural networks, and genetic algorithms.
CO2	Discuss and design different supervised learning networks and back propagation algorithms.
CO3	Explore un-supervised learning networks and its architecture.
CO4	Analyze the fuzzy logic and concept of fuzziness involved in various systems.
CO5	Design hybrid systems to revise the principles of soft computing in various applications.

Course Content

Unit No	Content	Hours
1	**UNIT-I- Introduction to Fuzzy logic, Neural Networks and Genetic Algorithm** Neural Networks, Application Scope of Neural Networks, Fuzzy Logic, Genetic Algorithm, Hybrid Systems, Soft Computing. Artificial Neural Network – Fundamental Concept, Evolution of Neural Networks, Basic Models of ANN.	10
2	**UNIT-II - Artificial Neural Network:** An introduction and Supervised Learning Network: McCulloch-Pitts Neuron, Hebb Network, Perception Networks, Adaptive Linear Neuron, Back Propagation Network-Theory, Architecture, Flowchart, Training Algorithm, Testing Algorithm, Learning factors of back-propagation network.	12
3	**UNIT-III - Unsupervised Learning Network and Special Networks** Introduction-FixedWeight Competitive Nets- Maxnet, Mexican Hat Net, Hamming Network, Kohonen self- organizing feature maps- Theory, Architecture, Flowchart, Training Algorithm, Kohonen Self-Organizing Motor Map, Learning Vector Quantization-Theory, Architecture, Flowchart, Training Algorithm, Variants, Probabilistic Neural Net.	12
4	**UNIT-IV- Introduction to Fuzzy Logic, Classical Sets and Fuzzy Sets:** Introduction to Fuzzy Logic, Classical Sets, Fuzzy Sets, Features of membership function, Fuzzification, methods of membership value assignments. Defuzzification-Lambda-cuts for fuzzy sets, Lambda-cuts for fuzzy relations, Defuzzification methods.	8
5	**UNIT- V- Hybrid Soft Computing Techniques and Applications of Soft Computing:** Introduction : Neuro- Fuzzy Hybrid Systems, Genetic Neuro Hybrid Systems. Applications: A fusion approach of multispectral images with SAR, optimization of traveling salesman problem using genetic algorithm-based internet search technique.	10

Reference Books

1. S.N. Sivanandam and S.N. Deepa, “Principles of Soft Computing”, Wiley India Pvt. Ltd., 3rd Edition, 2019.
2. S. Rajashekaran and G.A. Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis & Applications”, Prentice-Hall of India Pvt. Ltd., 2011.
3. David E. Goldberg, “Genetic Algorithm in Search Optimization and Machine Learning”, Pearson Education India, 2013.

OPEN ELECTIVE COURSES

<table>
<tr><td>Course Title</td><td colspan="8">Python Programming</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Concepts of Object-Oriented Programming Language

COURSE OBJECTIVE: This course aims to enable learners:

1. To understand the basic concepts of python programming
2. To acquire the knowledge of writing programs using functions.
3. To write python programs using object oriented concepts.
4. To understand the skills of writing GUI programs in python.
5. To acquire the knowledge of writing applications using databases.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Understand and write simple Python programs.
CO2	Use Python functions to organize code efficiently.
CO3	Understand basic object-oriented programming concepts.
CO4	Design simple graphical user interfaces (GUIs) using Python.
CO5	Connect Python programs to basic databases.

Unit No	Course Content	Hours
1	**UNIT-1: Introduction to Python Programming** Python Basics: What is Python? Installing Python, Writing and Running Simple Programs. Basic Concepts: Variables, Simple Data Types (numbers, strings), Basic Input/Output using print() and input(). Control Flow: Simple if statements, Loops (for and while), Using break and continue.	10
2	**UNIT-2: Working with Data Structures** Lists: Creating Lists, Accessing Elements, Basic List Operations (adding, removing, looping through lists). Tuples and Dictionaries: Simple Uses of Tuples, Creating and Accessing Dictionaries. Basic Data Handling: Using Lists and Dictionaries for Simple Data Management.	10
3	**UNIT-3: Writing Functions** Functions: What are Functions? Defining and Calling Simple Functions, Using Parameters and Return Values. Built-in Functions: Using Common Python Functions for Basic Tasks. Introduction to Modules: Using Standard Python Libraries like math and random.	10
4	**UNIT-4: Working with Files** File Basics: Reading from and Writing to Text Files. Simple File Operations: Opening, Closing, Reading, and Writing Data to Files. Error Handling: Understanding Simple Exceptions and How to Manage Them.	10
5	**UNIT-5: Basic Web Concepts** Introduction to Web Development: What is a Web Framework? Simple Overview of Django. Setting Up Django: Creating a Basic Project and Understanding How It Works. Creating Simple Web Pages: Using Django to Display Basic HTML Content.	10

Reference Books

1. "Python Crash Course, 3rd Edition" by Eric Matthes
2. "Automate the Boring Stuff with Python, 2nd Edition" by Al Sweigart
3. "Head-First Python, 2nd Edition" by Paul Barry

<table>
<tr><td>Course Title</td><td colspan="9">Artificial Intelligence</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Hard Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE

1. Basic understanding of data structures, algorithms, and discrete mathematics.
2. Familiarity with programming languages like Python or Java.
3. Prior exposure to logic, probability, and search algorithms will be beneficial.

COURSE OBJECTIVE: This course aims to enable learners:

1. Provide a solid understanding of Artificial Intelligence (AI) principles, including intelligent agents, search algorithms, and heuristics.
2. Enable students to implement and apply various AI problem-solving techniques and search strategies in real-world scenarios.
3. Introduce students to knowledge representation and inference systems, including first-order logic and propositional logic.
4. Develop skills in machine learning algorithms, including supervised learning and decision tree learning.
5. Equip students with basic knowledge in robotics, focusing on the integration of AI in robotic perception and movement planning

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Convert real-world problems into AI problems by selecting appropriate problem-solving strategies and search algorithms.
CO2	Formulate and implement informed search algorithms and heuristic methods to solve real-time and complex problems.
CO3	Represent and manipulate knowledge using first-order logic and propositional logic, understanding the fundamentals of knowledge representation.
CO4	Choose and implement machine learning algorithms (such as decision trees and inductive learning) for real-world supervised learning problems.
CO5	Understand and apply the principles of robotics, including robotic perception, sensors, and movement planning, integrating AI to solve problems in robotics.

Unit No	Course Content	Hours
1	**UNIT-1: Introduction to AI and Problem-Solving** Introduction to AI: What is AI? Basic concepts of agents and environments, Types of AI agents. Problem-Solving: Simple strategies for solving problems, Overview of search techniques (e.g., breadth-first and depth-first), Introduction to heuristics.	10
2	**UNIT-2: Games and Logical Reasoning** Game Playing: Understanding basic concepts of AI in games, Simple strategies for decision-making, Introduction to Alpha-Beta pruning. Logic Basics: What is logic? Simple propositional logic, Basic examples of reasoning.	10
3	**UNIT-3: Simple Logic and Inference** First-Order Logic: Basic understanding of first-order logic, Differences from propositional logic. Inference Techniques: Simple forward and backward reasoning, Introduction to unification and resolution.	10
4	**UNIT-4: Basics of Learning and Knowledge** Knowledge Representation: How to represent knowledge simply, Basic categories and examples. Machine Learning: Introduction to learning, Simple decision trees, Examples of AI applications today.	10
5	**UNIT-5: Introduction to Robotics** Robotics Basics: What are robots? Understanding sensors and simple robot actions. Movement and Planning: How robots move, Simple examples of planning and control, Basic applications of robots in real life.	10

Reference Books

1. "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell
2. "Artificial Intelligence: The Basics" by Kevin Warwick
3. "Introduction to Artificial Intelligence" by Wolfgang Ertel

<table>
<tr><td>Course Title</td><td colspan="8">Web Technologies</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Soft Core</td><td rowspan="2">Total Hours</td><td rowspan="2">52</td><td rowspan="2" colspan="2">Hours / Week</td><td rowspan="2">04</td><td>Credits</td><td>04</td></tr>
<tr><td>(L : T : P)</td><td>(3 : 1 : 0)</td></tr>
<tr><td rowspan="2">Course Code</td><td rowspan="2"></td><td rowspan="2">Evaluation</td><td>Internal</td><td colspan="3">C1 + C2 = 15 + 15</td><td>30 Marks</td><td rowspan="2">100 Marks</td></tr>
<tr><td>External</td><td>C3</td><td>Duration</td><td>03 Hours</td><td>70 Marks</td></tr>
</table>

COURSE PREREQUISITE: C++/Java/ Python Programming

COURSE OBJECTIVE: This course aims to enable learners:

1. To learn HTML tags and JavaScript Language programming concepts and techniques.
2. To develop the ability to logically plan and develop web pages.
3. To learn to write, test, and debug web pages using HTML and JavaScript.
4. To understand the concept of XAMPP server and node js.
5. To Implement basics of MEAN and FULL stack.

COURSE OUTCOME (CO): At the end of the course, the student will be able to

COURSE OUTCOMES	
CO1	Apply the concepts and usage of web based programming techniques.
CO2	Demonstrate the development of XHTML documents using JavaScript and CSS
CO3	Demonstrate applications of AngularJs and JQuery for the given problem.
CO4	Design and implement user interactive dynamic web based applications.
CO5	Implement modern web applications using MEAN and FULL STACK.

Course Content

Unit No	Content	Hours
1	**UNIT-1: Building Web Pages** HTML Basics: Understanding how web pages work, Creating simple web pages using HTML5 tags (text, images, lists, tables, audio, and video). Styling with CSS: Introduction to CSS, Basic styling of text and elements, Using the Box Model for layout, Simple use of div and span.	10
2	**UNIT-2: Making Web Pages Interactive** JavaScript Basics: Simple programming concepts using JavaScript (variables, control statements, functions). Adding Interactivity: Handling basic events like button clicks, Changing content dynamically, Simple form validation.	10
3	**UNIT-3: Useful Web Tools and Libraries** Introduction to jQuery: Simplifying JavaScript with jQuery, Using selectors and events to make web pages interactive. Getting Started with Bootstrap: Basics of Bootstrap for quick and easy page layouts, Using grids and common components (buttons, forms, navigation bars).	12
4	**UNIT-4: Working with Data** Introduction to AJAX: Understanding how to load data without refreshing the page, Simple AJAX examples to fetch and display data. XML Basics: What is XML? Simple examples of data formatting and reading XML	10
5	**UNIT-5: Introduction to Web Servers and Databases** Setting Up a Web Server: Using XAMPP to run a local server, Writing simple server-side code with PHP. Working with Databases: Connecting to MySQL, Performing simple operations like adding and retrieving data. Overview of Full Stack Concepts: What is Full Stack Development? A brief look at modern tools like Node.js and MEAN stack.	10

Reference Books.

1. "HTML and CSS: Design and Build Websites" by Jon Duckett, 1st Edition
2. "JavaScript and JQuery: Interactive Front-End Web Development" by Jon Duckett, 1st Edition
3. "Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5" by Robin Nixon, 5th Edition

Question Paper Blue Print

ST. PHILOMENA'S COLLEGE (AUTONOMOUS), MYSURU

II Semester C3 Examination November/December 2024

Subject: **Computer Application**

Title: _____

Time: 3 hours **Maximum Marks: 70**

	PART – A	
	Answer FOUR of the following	**4 × 5 = 20**
1.	Atleast One Question from each Unit	
2.		
3.		
4.		
5.		
	PART – B	
	Answer ONE of the following	**1 × 10 = 10**
6.	Unit 1 – can have sub questions (a) (b) for (5+5), (6+4), (7+3) or any suitable weightage	
7.		
	PART – C	
	Answer ONE of the following	**1 × 10 = 10**
8.	Unit 2 – can have sub questions (a) (b) for (5+5), (6+4), (7+3) or any suitable weightage	
9.		
	PART – D	
	Answer ONE of the following	**1 × 10 = 10**
10.	Unit 3 – can have sub questions (a) (b) for (5+5), (6+4), (7+3) or any suitable weightage	
11.		
	PART – E	
	Answer ONE of the following	**1 × 10 = 10**
12.	Unit 4 – can have sub questions (a) (b) for (5+5), (6+4), (7+3) or any suitable weightage	
13.		
	PART – F	
	Answer ONE of the following	**1 × 10 = 10**
14.	Unit 5 – can have sub questions (a) (b) for (5+5), (6+4), (7+3) or any suitable weightage	
15.		
	**********	