Savitribai Phule Pune University, Pune

Faculty of Commerce and Management

Master of Computer Applications (MCA)

Programme Curriculum (Sem. I & II)
(2020-2022)

Preamble:

1. The name of the programme shall be Masters of Computer Applications (M.C.A)
2. The revised MCA Curriculum 2020 builds on the implementation of the Choice Based Credit System (CBCS) and Grading System initiated in the AY 2015. The curriculum takes the MCA programme to the next level in terms of implementing Outcome Based Education along with the Choice Based Credit System (CBCS) and Grading System.
3. The Institutes should organize placement programme for M.C.A. students by interacting with Industries and software consultancy.
4. At the end of each semester, appearing for various certifications is possible for each student enabling them to make their resume rich.
5. With the rapidly changing scenario industry and academia should identify possible areas of collaboration and work together. Institute's placement cell should focus on identifying industrial expectations and institutional preparation for meeting industrial needs.

Introduction:

1. Definition: Outcome Based Education:

1.1 Outcome Based Education (OBE) Approach: Outcomes are about performance, and this implies:

- **1.1.1** There must be a performer – the student (learner), not only the teacher
- **1.1.2** There must be something performable (thus demonstrable or assessable) to perform
- **1.1.3** The focus is on the performance, not the activity or task to be performed

1.2 Programme Educational Objectives (PEOs): Programme educational objectives are broad statements that describe the career and professional accomplishments that the programme is preparing graduates to achieve. Programme Educational Objectives are a set of broad future focused learner's performance outcomes that explicitly identify what learners will be able to do with what they have learned, and what they will be like after they leave institution and are living full and productive lives. Thus, PEOs are what the programme is preparing graduates for in their career and professional life (to attain within a few years after graduation).

1.3 Programme Outcomes (POs): Programme Outcomes are a set of narrow statements that describes what students (learners) of the programme are expected to know and be able to perform or attain by the time of graduation.

1.4 Course Outcomes (COs): Course Outcomes are narrower statements that describe what students are expected to know and be able to do at the end of each course. These relate to the skills, knowledge, and behavior that students acquire in their matriculation through the course.

1.5 Learning Outcomes: A learning outcome is what a student CAN DO because of a learning experience. It describes a specific task that he/she can perform at a given level of competence under a certain situation. The three broad types of learning outcomes are: a) Disciplinary knowledge and skills b) Generic skills c) Attitudes and values

1.6 Teaching and Learning Activities (TLAs): The set of pedagogical tools and techniques or the teaching and learning activities that aim to help students to attain the intended learning outcomes and engage them in these learning activities through the teaching process.

1.7 Assessment and Evaluation: Assessment is one or more processes, carried out by the institution, that identify, collect, and prepare data to evaluate the achievement of programme educational objectives and programme outcomes. Evaluation is one or more processes, done by the evaluation team, for interpreting the data and evidence accumulated through assessment practices. Evaluation

1.8 determines the extent to which programme educational objectives or programme outcomes are being achieved, and results in decisions and actions to improve the programme.

2. MCA Programme Focus:

The basic objective of the Master of Computer Applications (MCA) is to provide a steady stream of necessary knowledge, skills and foundation for acquiring a wide range of rewarding careers into rapidly expanding world of Information Technology

2.1 Programme Educational Objectives: PEOs are defined by institution. Following are the guidelines for defining PEOs

- **2.1.1** PEOs should be assessable and realistic within the context of the committed resources.
- **2.1.2** The PEOs should be consistent with the mission of the institution.
- **2.1.3** All the stakeholders should participate in the process of framing PEOs.
- **2.1.4** The number of PEOs should be manageable.
- **2.1.5** It should be based on the needs of the stakeholders.
- **2.1.6** It should be achievable by the programme.
- **2.1.7** It should be specific to the programme and not too broad.
- **2.1.8** It should not be too narrow and similar to the POs.

2.2 MCA Programme Outcomes (POs): At the end of the MCA programme the learner will possess the following Program Outcome:

PO1: Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

PO2: Identify, formulate, research literature, and solve *complex* Computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing sciences, and relevant domain disciplines.
PO3: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO4: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.

PO5: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

PO6: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.

PO7: Recognize the need, and have the ability, to engage in independent learning for continual development as a Computing professional.

PO8: Demonstrate knowledge and understanding of computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO9: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

PO10: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.

PO11: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

PO12: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

3. Admission Details:

3.1 Eligibility for Admission: The eligibility criteria for admission for the MCA course will be as decided by the All Indian Council of Technical Education (AICTE), New Delhi and Directorate of Technical Education (DTE), Government of Maharashtra. It will be published on their respective websites time to time.

3.2 Reservation of Seat: The percentage of seat reserved for candidates belonging to backward classes only from Maharashtra State in all the Government Aided, Un-aided Institutions/Colleges and University Departments is as per the norms given by Government of Maharashtra, time to time.

3.3 Selection Basis: The selection would be done as per the guidelines given by the Director of Technical Education, Maharashtra State, time to time.

4. Lecture-Practical/Project-Tutorial (L-P-T)

A course shall have either or all the three components, i.e. a course may have only lecture component, or only practical/project component or a combination of any two/three components

4.1 Lecture(L): Classroom sessions delivered by faculty in an interactive mode. It should be conducted as per the scheme of lectures indicated in respective course.

4.2 Practical/Project(P): Practical / Project Work consisting of Hands-on experience /Field Studies / Case studies that equip students to acquire the much-required skill component. Besides separate Practical/Project course, three courses in each semester include few practical assignments and it will be evaluated under internal evaluation

4.3 Tutorial(T): Session consisting of participatory discussion/ self-study/ desk work/ brief seminar presentations by students and such other novel methods that make a student to absorb and assimilate more effectively the contents delivered in the Lecture sessions

4.4 A Mini project is an assignment that the student needs to complete at the end of every semester in order to strengthen the understanding of fundamentals through effective application of the courses learnt. The details guidelines have been given in the course structure.

4.5 The Project Work to be conducted in the FINAL Semester and evaluated at the end of the semester. The detail guidelines have been in the respective course structure.

4.6 The teaching / learning as well as evaluation are to be interpreted in a broader perspective as follows:

i) Teaching – Learning Processes: Classroom sessions, Group Exercises, Seminars, Small Group Projects, Self-study, etc.

ii) Evaluation: Tutorials, Class Tests, Presentations, Field work, Assignments, competency-based Activity, Research papers, Term papers, etc.

The MCA programme is a combination of:

a. Three-Credit Courses (75 Marks each): 3 Credits each
b. Two-Credit Courses (50 Marks each): 2 Credits each
c. One-Credit Courses (25 Marks each) : 1 Credits each

Following are the session details per credit for each of L-P-T model

1) **Every <u>ONE-hour session per week</u> of Lecture(L) amounts to <u>1 credit</u> per semester,**
2) **Minimum of <u>TWO hours per week</u> of Practical(P) amounts to <u>1 credit</u> per semester,**
3) **Minimum of <u>ONE hours per week</u> of Tutorial(T) amounts to <u>1 credit</u> per semester**

5. Open Courses (OC):

Institute has to offer two open courses of 1 credit each per semester to the students from Semester I to Semester III. The motive behind keeping an open course is to make students aware of current/upcoming trends in Information Technology and other domains. Full autonomy is given to the Institute to plan and execute the open courses. It is expected to extend the autonomy to the student

also. Care must be taken to consider credit points and necessary contact hours assigned to it while finalizing any open course for the given semester. In each semester total 2 credits are reserved for open courses.

Suggestive List of OPEN Courses

FOR SEMESTER I		FOR SEMESTER II		FOR SEMESTER III	
1	Data Privacy and Protection	1	Software Agent	1	Speech Recognition
2	Linux system administration	2	Aptitude building -1	2	Sentiment Analysis
3	social media listening	3	Basics of Tableau	3	R Programming
4	Research Methodology	4	Fraud detection	4	Gesture recognition
5	Applied Statistical Methods	5	Ruby Basics	5	Aptitude building-2
6	Digital Marketing	6	LaTeX	6	Digital Image processing
7	G-Suite	7	Big data Analytics	7	Network Security
8	Joomla	8	Game Programming in Unity	8	big data Technologies
9	e-trading	9	Block Chain Technology	9	AWS Fundamentals
10	Scratch and MIT App Inventor Programming	10	Business Intelligence - be specific	10	Edge Computing
11	Random Forest using MS Excel	11	Design Thinking & Problem-solving skills		
12	WordPress	12	Green Computing		
13	MS-OFFICE	13	IoT		
14	Code ignitor				

6. Extra Reading and Certification:

Each Chapter in the course is added with the extra reading part which gives extra pointer to gain In-depth knowledge apart from basic knowledge imparted in the syllabus. Learners should be encouraged to complete this extra reading portion as regular practice. Also, each course (Where ever applicable) includes suggested certification which help learners to enrich themselves as per industry demands and requirements.

7. Evaluation and Assessment:

In total 112 credits represent the workload of a year for MCA program.

Semester	Credit	IE	UE
Semester I	28	350	350
Semester II	28	350	350
Semester III	28	350	350
Semester IV	28	350	350
Total	**112**	**1400**	**1400**
			2800

The final total assessment of the candidate is made in terms of an internal (concurrent) evaluation and an external (university) examination for each course. In total the internal (concurrent) to external (university) marks ratio is maintained 50:50.

In general

1) For each course, 25 will be based on evaluation and 50 marks for semester end examination conducted by University, unless otherwise stated.
2) The internal evaluation of 25 marks further divided into Written Examination (Assignments/Unit test/written examination etc.), Practicals and Tutorials. The details have been specified in each course.
3) There will be one Practical course and one Mini Project course in each semester with 75 marks allotted for internal evaluation and 50 marks allotted for University examination. External assessment will be done by university appointed examiner. During external examination, examiner should ask the programs/practical ONLY from the work book of the students.
4) The internal marks will be communicated to the University at the end of each semester, but before the semester-end examinations. These marks will be considered for the declaration of the results.

Guidelines to conduct Mini-Project evaluation for Semester I, Semester II and Semester III of MCA – 2020 pattern

For Internal Evaluation

1. Internal evaluation will be of 75 Marks. It will be distributed as follows

Description	**Marks**
Project Report	35
Viva	15
Working Demo	25
Total	**75**

2. Project Report (including Project Diary) should be evaluated only during INTERNAL evaluation. Textual chapters should be given 10 marks while diagrams, test cases/validations, screen designs should be evaluated for 20 marks and 5 Marks should be given for Project Diary. Thus, totaling up to 35 marks.

For External Evaluation

1. Evaluation will be conducted by one Internal (Appointed by Institute) and one External examiner (Appointed by university).
2. External evaluation will be of 50 Marks. It will be distributed as follows

Description	**Marks**
Viva	15
Working Demo	35
Total	**50**

For Internal Evaluation and External Evaluation

1. VIVA should be conducted based on project domain and technologies used for developing the project. Every team member's individual contribution to the project may vary. Hence VIVA should be based on individual contribution pertaining to the project.
2. Working Demo is given maximum weightage to make sure that each group submits executable version of their project.
3. Examiners should evaluate efforts and contribution of every individual in the team (in case of group project).
4. Examiner may review code of the project while evaluating its working demo and modules.

Examination: Examinations shall be conducted at the end of the semester i.e. during November and in April/May. However supplementary examinations will also be held in November and April/May.

Concurrent Evaluation: A continuous assessment system in semester system (also known as internal assessment/comprehensive assessment) is spread through the duration of course and is done by the teacher teaching the course. The continuous assessment provides a feedback on teaching learning process. The feedback after being analyzed is passed on to the concerned student for implementation and subsequent improvement. As a part of concurrent evaluation, the learners shall be evaluated on a continuous basis by the Institute to ensure that student learning takes place in a graded manner. Concurrent evaluation components should be designed in such a way that the faculty can monitor the student learning & development and intervene wherever required. The faculty must share the outcome of each concurrent evaluation component with the students, soon after the evaluation, and guide the students for betterment Individual faculty member shall have the flexibility to design the concurrent evaluation components in a manner so as to give a balanced assessment of student capabilities across Knowledge, Skills & Attitude (KSA) dimensions based on variety of assessment tools.

Suggested components for Concurrent Evaluation (CE) are:

1. Case Study / Situation Analysis – (Group Activity or Individual Activity)
2. Class Test
3. Open Book Test
4. Field Visit / Study tour and report of the same
5. Small Group Project & Internal Viva-Voce
6. Learning Diary
7. Scrap Book
8. Group Discussion
9. Role Play / Story Telling
10. Individual Term Paper / Thematic Presentation
11. Written Home Assignment
12. Industry Analysis – (Group Activity or Individual Activity)
13. Literature Review / Book Review
14. Model Development / Simulation Exercises – (Group Activity or Individual Activity)
15. In-depth Viva
16. Quiz

Institute can decide the type, method and frequency of Concurrent Evaluation for each course and execute accordingly. Detailed record of the Concurrent Evaluation shall be maintained by the Institute. The same shall be made available to the University, on demand.

8. Choice based Credit System (CBCS) and Grading:

The detail document about Choice based Credit System for PG Programme is available on university website. The Grading methodology is also available on university website. University reserves rights to revise CBCS and grading system time to time.

9. Medium of Instruction:

The medium of Instruction will be English.

10.Clarification of Syllabus:

It may be necessary to clarify certain points regarding the course. The BOS should meet to study and clarify any difficulties from the Institutes, as and when required.

11.Revision of Syllabus:

As the computer technology is changing very fast, revision of the syllabus should be considered every 2 years.

12.Attendance:

The student must meet the requirement of 75% attendance per semester per course for grant of the term. The Director shall have the right to withhold the student from appearing for examination of a specific course if the above requirement is not fulfilled. Since the emphasis is on continuous learning and concurrent evaluation, it is expected that the student's study all-round the semester. Therefore, there shall not be any preparatory leave before the University examinations.

13.ATKT Rules:

The ATKT rules mention in CBCS handbook (available on university website) is application to MCA Programme.

14.Maximum Duration for completion of the Programme:

The candidates shall complete the MCA Programme WITHIN 5 YEARS from the date of admission, by earning the requisite credits. The student will be finally declared as failed if she/he does not pass in all credits within a total period of four years. After that, such students will have to seek fresh admission as per the admission rules prevailing at that time.

15.Structure of the Programme and detail syllabus of each course:

Semester I					
Sr. No.	**Course Title**	**Course Code**	**CP**	**EXT**	**INT**
1	Java Programming	IT11	3	50	25
2	Data Structure and Algorithms	IT12	3	50	25
3	Object Oriented Software Engineering	IT13	3	50	25
4	Operating System Concepts	IT14	3	50	25
5	Network Technologies	IT15	3	50	25
6	Open Course 1	OC11	1		25
7	Open Course 2	OC12	1		25
	*** Practicals**				
8	Practical	IT11L	5	50	75
9	Mini Project	ITC11	5	50	75
	Soft Skills				
10	Soft Skills - I	SS11	1		25
			28	**350**	**350**

Semester II					
Sr. No.	**Course Title**	**Course Code**	**CP**	**EXT**	**INT**
1	Python Programming	IT21	3	50	25
2	Software Project Management	IT22	3	50	25
3	Optimization Techniques	MT21	3	50	25
4	Advanced Internet Technologies	IT23	3	50	25
5	Advanced DBMS	IT24	3	50	25
6	Open Course 3	OC21	1		25
7	Open Course 4	OC22	1		25
	*** Practicals**				
8	Practical	IT21L	5	50	75
9	Mini Project	ITC21	5	50	75
	Soft Skills				
10	Soft Skills - II	SS21	1		25
			28	**350**	**350**

Semester III					
Sr. No.	**Course Title**	**Course Code**	**CP**	**EXT**	**INT**
1	Mobile Application Development	IT31	3	50	25
2	Data Warehousing and Data Mining	IT32	3	50	25
3	Software Testing and Quality Assurance	IT33	3	50	25
4	Knowledge Representation & Artificial Intelligence - ML, DL	IT34	3	50	25
5	Cloud Computing	IT35	3	50	25
6	Open Course 5	OC31	1		25
7	Open Course 6	OC32	1		25
	*** Practicals**				
8	Practical	IT31L	5	50	75
9	Mini Project	ITC31	5	50	75
	Soft Skills				
10	Soft Skills- III	SS31	1		25
			28	**350**	**350**

Semester IV					
Sr. No.	**Course Title**	**Course Code**	**CP**	**EXT**	**INT**
1	DevOps	IT41	3	50	25
2	PPM and OB	BM41	3	50	25
2	Project	ITC41	22	250	300
			28	**350**	**350**

Semester	**Credit**	**IE**	**UE**
Semester I	28	350	350
Semester II	28	350	350
Semester III	28	350	350
Semester IV	28	350	350
Total	**112**	**1400**	**1400**
			2800

Semester I

Course Code: IT-11
Course Name: Java Programming

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	25	-	-	50	75

Course Description:

Prerequisite:

Knowledge of programming structures like decision flows, loops, etc.

Course Objectives:

1. To familiarize students with the concepts of OOPs.
2. To enable the students to understand the core principles of the Java Language and use AWT tools to produce well designed, effective applications.
3. Students will be able to develop server-side applications with database handling using servlets, JSP, JDBC

Course Outcomes:

Student will be able to

CO1 Understand Basic Concepts of OOPs, Java, Inheritance, Package. (Understand)
CO2: Understand Exception handling, arrays and Strings and multi-threading in Java (Understand.)
CO3: Understand collection framework (Understand)
CO4: Develop GUI using Abstract Windows Toolkit (AWT) and event handling (Apply)
CO5: Develop Web application using JSP and Servlet, JDBC (Apply)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Introduction -** 1.1 About Java 1.2 Flavours of Java 1.3 Java Installation 1.4 Java Program Development Environment **Extra reading**: docs oracle docs	3	2

2	**2. Object Oriented Programming** 2.1 Class Fundamentals 2.2 Object & Object reference 2.3 Object Life time & Garbage Collection 2.4 Creating and Operating Objects 2.5 Constructor & initialization code block 2.6 Access Control, Modifiers, Use of Modifiers with Classes & Methods. 2.7 Nested, Inner Class & Anonymous Classes, Abstract Class & Interfaces 2.8 Methods, Defining Methods, Argument Passing Mechanism, Method Overloading, Recursion, Dealing with Static Members, Finalize () Method, Native Method. 2.9 Use of "this "reference, 2.10 Design of Accessors and Mutator Methods 2.11 Cloning Objects, shallow and deep cloning 2.12 Generic Class Types. **Extra Reading** : OCA Java Programmer :I Exam Kathy Sierra	8	3
3	3. **Extending Classes and Inheritance** 3.1 Use and Benefits of Inheritance in OOP 3.2 Types of Inheritance in Java 3.3 Inheriting Data members and Methods 3.4 Role of Constructors in inheritance 3.5 Overriding Super Class Methods, Use of "super" 3.6 Polymorphism in inheritance 3.7 Type Compatibility and Conversion 3.8 Implementing interfaces **Extra Reading**: Understanding and practicing above concept in depth - OCA Java Programmer: I Exam Kathy Sierra	6	3
4	4. **Package** 4.1 Organizing Classes and Interfaces in Packages 4.2 Package as Access Protection 4.3 Defining Package 4.4 CLASSPATH Setting for Packages 4.5 Making JAR Files for Library Packages 4.6 Import and Static Import 4.7 Naming Convention for Packages. **Extra Reading**: Oracle Java tutorial	3	2
5	5. **Exception Handling**	6	3

	5.1 The Idea behind Exception 5.2 Exceptions & Errors 5.3 Types of Exception 5.4 Control Flow in Exceptions 5.5 JVM reaction to Exceptions 5.6 Use of try, catch, finally, throw, throws in Exception Handling 5.7 In-built and User Defined Exceptions Checked and Un-Checked Exceptions **Extra Reading**: Oracle Java tutorial		
6	**6. Array & String**: 6.1 Defining an Array 6.2 Initializing & Accessing Array 6.3 Multi –Dimensional Array 6.4 Operation on String, Mutable & Immutable String 6.5 Using Collection Bases Loop for String, Tokenizing a String 6.6 Creating Strings using StringBuffer, String Builder Extra Reading : Java arrays, tokenizer applications– Jenkov Tutorials	4	2
7	**7. Thread** 7.1 Understanding Threads 7.2 Needs of Multi-Threaded Programming 7.3 Thread Life-Cycle 7.4 Thread Priorities 7.5 Synchronizing Threads 7.6 Inter Communication of Threads 7.7 Critical Factor in Thread –Deadlock Extra Reading : Animation Using Thread	6	3
8	**8. A Collection of Useful Classes** 8.1 Utility Methods for Arrays 8.2 Observable and Observer Objects, 8.3 Date & Times, 8.4 Using Scanner 8.5 Regular Expression, 8.6 Input/output Operation in Java (java.io Package) 8.7 Streams and the new I/O Capabilities 8.7.1 Understanding Streams **8.7.2** The Classes for Input and Output **8.7.3** The Standard Streams 8.8 Working with File Object 8.8.1 File I/O Basics, **8.8.2** Reading and Writing to Files	6	3

	8.8.3 Buffer and Buffer Management **8.8.4** Read/Write Operations with File Channel 8.9 Serializing Objects Extra Reading : regex – Pattern matching, split examples, reading and writing Character Stream, Byte stream and Objects in java files.		
9.	**9. UI Programming** **9.1** Designing Graphical User Interfaces in Java, 9.2 Components and Containers, 9.3 Basics of Components 9.4 Using Containers 9.5 Layout Managers, 9.6 AWT Components 9.7 Adding a Menu to Window 9.8 Extending GUI Features Using Swing Components Extra Reading : Using Swing toolkit GUI –oracle java tutorial	12	5
10	**10. Event Handling** 10.1 Event-Driven Programming in Java 10.2 Event- Handling Process 10.3 Event Handling Mechanism 10.4 The Delegation Model of Event Handling 10.5 Event Classes, Event Sources, Event Listeners 10.6 Adapter Classes as Helper Classes in Event Handling. Extra Reading : Hierarchy of Event Classes, Event Sources, Event Listeners- Oracle java docs	10	4
11	**11. The Collection Framework** 11.1 Introduction to Java Frameworks 11.2 Collections of Objects 11.3 Collection Types, Sets, Sequence, Map 11.4 Understanding Hashing 11.5 Use of ArrayList & Vector 11.6 Java Utilities (java.util Package) Extra Reading : searching, sorting, insertion, manipulation, deletion of data using Java Collections	10	4
12	**12. Database Programming using JDBC** 12.1 Introduction to JDBC 12.2 JDBC Drivers & Architecture 12.3 CURD operation Using JDBC 12.4 Connecting to non-conventional databases	10	4

	Extra Reading: List of JDBC Drivers and Jars, Statement, Prepared Statement and Callable Statement.		
13	**13. Java Server Technologies** 13.1 Servlet Web Application Basics, 13.2 Architecture and challenges of Web Application 13.3 Introduction to servlet 13.4 Introduction to JSP 13.5 Servlet life cycle 13.6 Developing and Deploying Servlets, Exploring Deployment Descriptor (web.xml) 13.7 Handling Request and Response. Extra Reading : Session handling 4 methods, Request Dispatcher ,JSP Tags, JSP Implicit objects, Generic Servlet	16	7
	Total:	**100**	**45**

Course References:

Recommended Books:

Text Books:

1. Java Complete Reference Schildt Herbert, TMH.
2. Java Fundamentals (SIE), Schildt Herbert, TMH
3. The Complete Reference JSP, Phil Hanna, TMH
4. JDBC, Servlet and JSP, Black Book, Santosh Kumar K. Dremtech publication

Reference Books:

1. Head First Servlets and JSP, 2nd Edition by Bert Bates, Bryan Basham, Kathy Sierra
2. OCJP Oracle Certified Programmer for Java Study Guide by Kathy Sierra and Bert Bates.
3. A Programmer's Guide to Java OCJP Certification (A Comprehensive Primer) by Khalid A. Mughal and Rolf W. Rasmussen.
4. Java Server Programming Java Ee&(J2EE 1.7), Black Book, Wiley publications

Recommended Learning Material:

1. www.javatpoint.com
2. www.oracle.com
3. www.tutorialspoint.com
4. www.geeksforgeeks.org/java

Recommended Certification:

1. OCA- Oracle Certified Associate
2. OCP- Oracle Certified Professional

Course Code: IT-12
Course Name: Data Structure and Algorithms

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	25	-	-	50	75

Course Description:

Prerequisite

Loops, Functions, Pointers, Arrays, Memory Allocation, Recursion

Course Objectives:

1. To understand basics data structure and algorithms
2. To solve problems using data structures such as linked lists, stacks, queues, hash tables, trees, heaps and graphs
3. To understand various programming techniques such as brute force, greedy, dynamic programming, divide-conquer and backtracking

Course Outcomes:

Student will be able to

CO1: demonstrate linear data structures linked list, stack and queue (apply)
CO2: implement tree, graph, hash table and heap data structures (apply)
CO3: apply brute force and backtracking techniques (apply)
CO4: demonstrate greedy and divide-conquer approaches (apply)
CO5: implement dynamic programming technique (apply)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**Linked List** 1.1 Singly Linked List 1.2 Doubly Linked List Extra Reading: Circular Linked list and Circular doubly linked list	8	2
2	**Stack and Queues** 2.1 Linked List implementation of Stack 2.2 Linked List implementation of Queue 2.3 Circular Queue 2.4 Priority Queue	10	4

	Extra Reading: Dqueue, Application of Stack		
3	**Tree** 3.1 Tree 3.2 Binary Search Tree 3.3 AVL Tree 3.4 Red-Black Tree 3.5 Segment Tree - with min/max/sum range queries examples 3.6 Fenwick Tree (Binary Indexed Tree) Extra Reading: Application of Tree, B* tree	12	5
4	**Graph** 4.1 Directed and Undirected Graph 4.2 Graph Representations 4.2.1 Adjacency Matrix 4.2.2 Adjacency List 4.3 Graph Traversals 4.3.1 BFS 4.3.2 DFS Extra Reading: Application of Graph in Maps	8	2
5	**Hash Table and Heaps** 5.1 Hash Table 5.1.1 Hash Function 5.1.2 Hash function approaches 5.1.3 Handling the collisions 5.2 Heap 5.2.1 Min heap and Max heap Extra Reading: Hashing used in File handling	7	2
6	**Brute Force** 6.1 Linear Search 6.2 Rain Terraces 6.3 Recursive Staircase 6.4 Maximum Subarray 6.5 Travelling Salesman Problem 6.6 Discrete Fourier Transform Extra Reading: Application in Cryptography	10	5
7	**Greedy** 7.1 Jump Game 7.2 Unbound Knapsack Problem 7.3 Dijkstra Algorithm 7.4 Prim's Algorithm 7.5 Kruskal's Algorithm	10	5

	Extra Reading: Huffman's Tree		
8	**Divide and Conquer** 8.1 Binary Search 8.2 Tower of Hanoi 8.3 Pascal's Triangle 8.4 Euclidean Algorithm 8.5 Merge Sort 8.6 Quicksort 8.7 Fast Powering Extra Reading: Cooley–Tukey Fast Fourier Transform (FFT) algorithm	10	8
9	**Dynamic Programming** 9.1 Fibonacci Number 9.2 Unique Paths 9.3 Longest Common Subsequence (LCS) 9.4 Longest Common Substring 9.5 Longest Increasing Subsequence 9.6 Shortest Common Super sequence 9.7 0/1 Knapsack Problem 9.8 Integer Partition 9.9 Regular Expression Matching Extra Reading: Painting Fence Algorithm, Moser-de Bruijn Sequence, Newman-Conway Sequence	15	7
10	**Backtracking** 10.1 Power Set 10.2 Hamiltonian Cycle 10.3 N-Queens Problem 10.4 Knight's Tour 10.5 Combination Sum Extra Reading: Word Break Problem using Backtracking	10	5
Total:		**100**	**45**

Note: Course should be taught independent of any programming language.

Course References:

Recommended Books:

Text Books

1. Jean Paul Tremblay, Paul G. Sorensons, “AN Introduction to Data Structures with Application”, McGraw Hall Publication (INDIAN edition)
2. A. V. Aho and J.D. Ullman, “Design and Analysis of Algorithms”, Addison Wesley
3. Thomas H Cormen and Charles E.L Leiserson, “Introduction to Algorithm” PHI

Reference Books

1. Lipschutz Schaum’s, “Data Structure”, Outline Series, MH
2. D. Samanta, “Classical Data Structure”, PHI,
3. Practical Approach to Data Structures by Hanumanthappa.
4. Data Structure and Algorithms in C++ by Joshi Brijendra Kumar
5. Data Structures with C++: Schaum’s Outlines by Hubbard JohnBressard,
6. Horowitz/Sahani, Fundamental of Algorithm. PHI, Galgotia.
7. Magnifying Data Structures, Arpita Gopal, PHI Publications

Course Code: IT-13
Course Name: Object Oriented Software Engineering

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	10	10	5	50	75

Course Description:

Prerequisite:

Basic System Analysis and Design Concept

Course Objectives:

1. To study basic concepts of software engineering
2. To study phases of SDLC and different process models
3. To learn & understand the Requirement analysis and system Design.
4. To get acquainted with the agile software development methodology

Course Outcomes:

Student will be able to

CO1: Distinguish different process model for a software development. (Understand)
CO2: Design software requirements specification solution for a given problem definitions of a software system. (Analyze)
CO3: Apply software engineering analysis/design knowledge to suggest solutions for simulated problems (Analyze)
CO4: Design user interface layout for different types of applications (Apply)
CO5: Recognize and describe current trends in software engineering (Understand)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Introduction to development approach SSAD and OOAD** 1.1. Overview of Software Development with SSAD 1.1.1. Basic System Development Life Cycle with different users and their role in SDLC. 1.1.2. Different Approaches and Models for System Development. 1.1.2.1. Waterfall Model 1.1.2.2. Spiral Model 1.1.2.3. Prototyping Model 1.1.2.4. RAD	10	4

	1.1.2.5. Rational Unified Process Extra Reading: Object oriented concepts		
	2. Requirement Engineering 2.1. Types of Requirements – Functional and Non-functional 2.2. Four Phases of Requirement Engineering 2.3. Software requirement Specification (SRS) 2.3.1. Structure and contents of SRS 2.3.2. IEEE standard format for SRS Note: Case studies based on SRS Extra Reading: Fact finding techniques, Feasibility study and reports	20	9
	3. Use-case Driven Object-Oriented Analysis 3.1. Introduction to oops concepts 3.1.1. Class and object 3.1.2. Abstraction and encapsulation 3.1.3. Method and messages 3.1.4. Interface, Inheritance and polymorphism 3.1.5. Structural Diagram - Class Diagram and Object diagram 3.1.6. Associations and links 3.1.7. Aggregation, Composition and containment 3.1.8. Inheritance, Sub Types and IS-A hierarchy 3.2. Behavioral Diagram 3.2.1. Use case Diagram 3.2.1.1. Identify Actors 3.2.1.2. Identify Use cases: describing how the user will use the system 3.2.1.3. Develop use-case Model 3.2.1.4. Description of Use case Diagram. 3.2.2. Activity Diagram 3.2.3. Sequence diagram 3.2.4. Collaboration Diagram. 3.2.5. State Transition Diagram Note: Case studies should be covered on the above topic Extra Readings: UML diagram drawing tools such as draw.io, Star UML, etc. , Documentation associated with UML diagrams	40	20

	4. User Interface Design 4.1. Elements of good design 4.2. Eight golden rules for design 4.3. Features of modern GUI, Menus, Scroll bars, windows, buttons, icons, panels, error messages etc. Note: Case studies should be covered on the above topic Extra Readings: UI/UX software, Interactive UI design	10	4
	5. Current trends in Software Engineering 5.1. Introduction to Web Engineering 5.2. Agile Process 5.2.1. Agile Process Models 5.2.1.1. Extreme Programming (XP) 5.2.1.2. Adaptive Software Development (ASD) 5.2.1.3. Dynamic Systems Development Method (DSDM) 5.2.1.4. Scrum 5.2.1.5. Crystal 5.2.1.6. Feature Driven Development (FDD) Extra Readings: Comparative analysis of traditional process models and agile, Agile methodology in testing	20	8
	Total:	**100**	**45**

List of Practical (if any)

Case studies will be given to the student as a part of tutorial. The same diagram should be drawn using UML diagram drawing tool as practical.

Course References:

Recommended Books:

Text Books:

1. Software Engineering by Roger Pressman (6th edition)
2. Object-Oriented Software Engineering: A Use Case Driven Approach by Ivan Jacobson
3. Software Engineering by Sommerville, Pearson,8th Ed
4. Analysis & Design of Information System James Senn, TMH, 2nd Ed
5. Object Oriented System Development - Ali Bahrami McGraw-Hill International Edition
6. Object-Oriented Software Engineering - Ivar Jacobson Pearson Education INC
7. Agile Software Engineering with visual studio by Sam Guckenheimer, Neno Loje.
8. UML Instant – Thomas A Pendar – Wiley Publication

9. UML in Nutshell, O'reilly Pub

Reference Books:

1. Software Requirements by Karl Wiegers
2. Object Oriented Modeling and Design with UML by James Rumbaugh, Michael Blaha
3. Object Oriented Systems and Techniques with UML & Java by Udit Agarwal
4. Software Engineering by Chandramouli Subramanian, Saikat Dutt
5. Object Oriented Systems Analysis and Design using UML by Simon Bennett
6. UML 2 Bible by Tom Pender
7. The Unified Modeling Language user guide by Grady Booch, James Rumbaugh, Ivar Jacobson

Recommended Learning Material:

5. https://www.mooc-list.com/course/object-oriented-design-coursera
6. https://nptel.ac.in/courses/106101061/

Course Code: IT-14
Course Name: Operating Systems Concepts

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	15	10	-	50	75

Course Description:

Prerequisite:

Basics of Operating System

Course Objectives:

1. To learn the fundamentals of Operating Systems and handle processes and threads and their communication
2. To learn the mechanisms involved in memory management in contemporary OS
3. To know the functionality of Multiprocessor OS and Mobile OS.
4. To gain knowledge on distributed operating system concepts.
5. To learn about Basics of Linux.
6. To learn programmatically to implement Linux OS mechanisms.

Course Outcomes:

Student will be able to

CO1: Understand structure of OS, process management and synchronization. (Understand)
CO2: Understand multicore and multiprocessing OS. (Understand)
CO3: explain Realtime and embedded OS (Understand)
CO4: understand Windows and Linux OS fundamentals and administration. (Understand)
CO5: solve shell scripting problems (Apply)

Course Structure:

Serial No.	Topics Details	Weightage in %	No of Sessions
1	**1. Overview** 1.1. Overview of operating systems 1.2 Functionalities and Characteristics of OS 1.3 Hardware concepts related to OS 1.4 CPU states 1.5 I/O channels 1.6 Memory Management 1.6.1 Memory Management Techniques 1.6.2 Contiguous & Non-Contiguous allocation 1.6.3 Logical & Physical Memory –	15	7

	Conversion of Logical to Physical address 1.7 Paging 1.7.1 Demand Paging 1.7.2 Page Replacement Concept 1.8 Segmentation - Segment with paging 1.9 Virtual Memory Concept 1.10 Thrashing Extra Reading: Type of OS, Batch OS, Time Sharing OS, Network OS, Multiprogramming OS, Multiprocessing OS, Evolution of Operating System., Computer System Organization Operating System Structure and Operations- System Calls, System Programs, OS Generation and System Boot		
2	2. Process Management and Synchronization 2.1 PCB 2.2 Job and processor scheduling 2.3 Scheduling Concept 2.4 Process hierarchies 2.5 Problems of concurrent processes 2.6 Critical sections 2.7 Mutual exclusion 2.8 Synchronization 2.9 Deadlock 2.10 Device and File Management 2.10.1 Overview 2.10.2 Techniques 2.10.3 File Systems Extra Reading: Threads- Overview, Multithreading models, Threading issues, Process Synchronization – The critical-section problem, Synchronization hardware, Mutex locks, Semaphores, Classic problems of synchronization, Critical regions, Monitors; Deadlock – System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock., Banker's Algorithms	17	8
3	3. Multiprocessor and Multicore Operating Systems 3.1 Introduction 3.1.1 Advantages and Disadvantages 3.1.2 Multicore System Vs. Multiprocessor System. 3.2 Types of Multiprocessors 3.2.1 Symmetric Multiprocessors 3.2.2 Asymmetric Multiprocessors 3.3 Basic Multicore Concepts: Memory Sharing Styles	17	8

	3.3.1 Uniform Memory Access (UMA) 3.3.2 Non-Uniform Memory Access (NUMA) 3.3.3 No Remote Memory Access (NORMA) 3.4 Cache Coherence, Inter-Process (and inter-core) Communication: 3.4.1 Shared Memory 3.4.2 Message Passing 3.5 Mobile Operating Systems 3.5.1 Concept Need and Features 3.5.2 Types of Mobile OS 3.5.3 Overview of Android OS 3.5.4 Applications of Mobile OS 3.6 Distributed Operating Systems 3.6.1 Concept Need and Features 3.6.2 Examples of Distributed OS with brief introduction 3.6.3 Applications of Distributed OS Extra Reading: Virtual Machine, Cache Memory and Catching Concept, Multi-Processor and Distributed Operating System: – Introduction, – Architecture, – Organization, – Resource sharing, – Load Balancing, – Availability and Fault Tolerance, – Design and Development Challenges, – Inter-process Communication		
4	4. Real Time OS 4.1 Introduction and use of RTOS 4.1.2 Components of RTOS 4.1.3 Types of RTOS 4.1.4 Features of RTOS 4.1.5 Factors for selecting in RTOS 4.1.6 Applications of RTOS 4.1.7 Disadvantages of RTOS 4.2 Embedded OS 4.2.1 Concept Need and Features of embedded OS 4.2.2 Examples of embedded OS with brief introduction 4.2.3 Applications of embedded OS Extra Reading: Real Time and Embedded Operating Systems: – Introduction, – Hardware Elements, – Structure Interrupt Driven, Interrupt Driven, Nanokernel, Nanokernel, Microkernel and Microkernel and Monolithic kernel-based models. Monolithic kernel-based models. – Scheduling – Periodic, Periodic, Aperiodic and Aperiodic and Sporadic Tasks, Sporadic	10	4

	Tasks, – Introduction to Energy Aware CPU Scheduling.		
5	5.Windows OS and Windows Server Architecture 5.1 Windows OS 5.1.1 Introduction 5.1.2 Windows OS Installation 5.1.3 Process Management 5.1.4 Control Panel Overview 5.1.5 Users, Security and Privacy Settings 5.1.6 Identify Accessibility Settings 5.1.7 Service Management 5.1.8 Syncing Devices and File Sharing 5.1.9Windows Utilities (Accessories, Disk Management, Resource Monitor, Backup and Recovery), Basic Troubleshooting (Networking, Security, Device Driver). 5.2 Introduction to Ubuntu 5.2.1 Introduction 5.2.2 Overview of Kernel 5.2.3 Installation of Ubuntu 5.2.4 File system 5.2.5 Basic Commands of Linux 5.2.6 Managing Processes in Linux 5.2.7 Installing and deleting software packages 5.2.8 User Management 5.2.9 File and Device Management 5.2.10 Backup and recovery 5.2.11 Introduction to Graphical Environment (GNOME), Ubuntu Utilities (VirtualBox, Evolution, Gimp, Bleach Bit, Unity Tweak Tool etc.), SAMBA Overview Extra Reading: Deploying and Managing Windows Server 2012 and 2016, Introduction to Active Directory Domain Services, Managing Active Directory Domain Services Objects, Automating Active Directory Domain Services Administration, Implementing IPv4, Implementing DHCP, Implementing DNS, Implementing Local Storage, Implementing File and Print Services, Implementing Group Policy.	25	12
6	6. Linux Shell Scripting 6.1 Introduction 6.2 Variables 6.3 Flow Controls 6.4 Loops 6.5 Functions	16	6

	6.6 Lists 6.7 Manipulating Strings 6.8 Reading and Writing Files 6.9 Positional Parameters 6.10 Case statement 6.11Real time scripts for different system administration activities Extra Reading: Shell Script Programming Concepts, Sequential Flow and Components of Shell Scripting, Decision Structures, Decision-Structure Theory, Statements & Operators, Looping Structures, Loop Theory & Statements, Functions and Arrays, Functions Parts/Libraries & Arrays, Advanced Shell Programming, File Access, Sorts & Techniques, Advanced Tech & Tools , Awk &Sed ,Script Design and Management Issues		
	Total:	**100**	**45**

List of Practicals (if any)

Minimum 5 Practicals to be conducted based on Shell Scripting

Course References:

Recommended Books:

Text Books:

1. “Operating System Concepts”, Abraham Silberschatz; Peter Baer Galvin; Greg Gagne, Seventh Edition, John Wiley & Sons, 2004.
2. “Advanced Concepts in Operating Systems– Distributed, Database, and Multiprocessor Operating Systems”, Mukesh Singhal and Niranjan G. Shivaratri, Tata McGraw-Hill, 2001.
3. “Understanding the Linux kernel”, 3rd edition, Daniel P Bovet and Marco Cesati, O’Reilly, 2005.
4. “Real-Time Systems: Theory and Practice”, Rajib Mall, Pearson Education India, 2006.
5. “iPhone iOS 4 Development Essentials – Xcode”, Neil Smyth, Fourth Edition, Payload
6. media, 2011.
7. “Microsoft Windows Server Administration Essentials”, Tom Carpenter
8. ”The Official Ubuntu Book” Eighth Edition, Matthew Helmke, Elizabeth K. Joseph, José Antonio, Rey Philip Ballew, With Benjamin Mako Hill

Reference Books:

1. " Operating Systems: Internals and Design Principles" by William Stallings.
2. " Operating Systems: A Concept-Based Approach" by D M Dhamdhere.
3. System Concepts, 9th Edition, John Wiley & Sons, Inc. by Avi Silberschatz, Peter Baer Galvin, Greg Gagne,
4. D.M Dhamdhere: Operating systems - A concept-based Approach, 3rd Edition, Tata McGraw- Hill, 2012.
5. Operating Systems: Internals and Design Principles, 8th edition Pearson Education Limited, 2014 by William Stallings.
6. Modern Operating system by Andrew Tenenbaum.
7. Distributed Operating System by Andrew Tanenbaum
8. P.C.P. Bhatt: Introduction to Operating Systems Concepts and Practice, 3rd Edition, PHI, 2010.
9. Harvey M Deital: Operating systems, 3rd Edition, Pearson Education, 2011

Recommended Certification:

Windows Server certifications

1. Entry level: IT Infrastructure: Microsoft Technology Associate (MTA) certification.
2. Associate level: Windows Server 2016: Microsoft Certified Solutions Associate (MCSA): Windows Server 2016 certification.
3. Associate level: Windows Server 2012: Microsoft Certified Solutions Associate (MCSA): Windows Server 2012 certification.
4. Expert level: Server infrastructure: Microsoft Certified Solutions Expert (MCSE): Server Infrastructure certification.

Linux Certifications

1. LINUX+ CompTIA
2. RHCE- RED HAT CERTIFIED ENGINEER
3. GCUX: GIAC CERTIFIED UNIX SECURITY ADMINISTRATOR
4. ORACLE LINUX OCA & OCP
5. LPI (LINUX PROFESSIONAL INSTITUTE) CERTIFICATIONS

    LPIC- 1: Linux Administrator
    LPIC- 2: Linux Engineer
    LPIC- 3: Linux Enterprise Professional Certification

Course Code: IT-15
Course Name: Network Technologies

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	10	10	5	50	75

Course Description:

Prerequisite:

Students should have fundamental knowledge of computer network.

Course Objectives:

1. To understand various computer networks and technologies behind networks
2. To study TCP/IP protocol suite, IP addressing schemes and link layer communication
3. To study routing concept along with Routing protocols
4. To study application layer protocols
5. To understand basics of cryptography and socket programming

Course Outcomes:

Student will be able to

CO1: Understand the basic concepts of Computer Network, and principle of layering (Understand)
CO2: Apply the error detection and correction techniques used in data transmission (Apply)
CO3: Apply IP addressing schemes and sub netting (Apply)
CO4: Understand the concept of routing protocols, Application layer protocols and Network Security (Understand)
CO5: Apply the socket programming basics to create a simple chat application (Apply)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Introduction to Data Communication and Computer Networks** 1.1. Internet basics and network components. [Transmission Media-Guided, Unguided, Network Devices] 1.2. Various types of Networks (only overview) 1.2.1. Connection Oriented N/Ws Vs Connectionless N/Ws, 1.2.2. Ethernet- Ethernet standards ZigBee, WiFi, Access Technique -CSMA-CD, Negotiation technique Overview	**6**	**3**

	1.2.3. Wireless Network 1.3 Unified Communication –VOIP Extra Reading: Switching Techniques, CSMA/CA, CSMA/CD, Unified Communication		
2	**2. Principle of Layering concept** 2.1 Need for layering 2.2 ISO-OSI 7 Layer Model 2.3 TCP/IP model 2.4 OSI Model vs TCP/IP mode Extra Reading: Data Encapsulation, PDU Formation, network devices	**12**	**8**
3	**3. Link Layer Communication** 3.1 Error detection and correction techniques 3.2 Framing and its types 3.3 Flow and error control 3.4 HDLC protocol 3.5 P2P Protocol Note: Examples based on 3.1 to be covered Extra Readings: DLL protocol examples, IEEE 802.2 MAC protocol	**25**	**10**
4	**4. IP Addressing** 4.1 Internet Protocol and IPv4 Packet format, 4.2 Addressing, Physical Addresses, Logical Addresses Port Addresses, Specific Addresses 4.3 IP Address- Network Part and Host Part 4.4 Network Masks, Network Addresses and, Broadcast Addresses, Loop Back Address 4.5 Address Classes 4.6 TCP and UDP Connections 4.7 TCP Performance in wireless network 4.8 Overview of IPv6 4.9 IP Routing - Types of routing protocol, Border Gateway Protocol (BGP), Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Routing Table concept Notes: Examples based on IP addressing and sub netting to be covered <u>Extra Reading</u>: Network Monitoring Tools –Open NMS, Putty, Wireshark, Nagios core, Cacti	**25**	**10**

5	**5. Application Layer Protocols** 5.1 DHCP – DHCP Client, DHCP server, DHCP scope 5.2 DNS – Resolution process, Resource Records, DNS protocol structure 5.3 HTTP – WWW architecture, HTTP: Request and Response Message 5.4 Email protocols – SMTP, POP3, IMAP4 & MIME 5.5 FTP, Telnet Extra Reading: Practical on FTP, Telnet, DNS, Putty	**15**	**6**
6	**6. Network Security** 6.1Active and Passive attacks 6.2Cryptography (Symmetric and Asymmetric) 6.3Firewall Extra Reading: Examples on symmetric and asymmetric algorithms	**5**	**2**
7	**7. Socket Programming** 7.1 Introduction 7.2 Berkeley Sockets 7.3 Specifying A Protocol Interface 7.4 The Socket Abstraction 7.4.1 System Data Structures for Sockets 7.5 Specifying an Endpoint Address 7.6 A Generic Address Structure 7.7 Major System Calls Used with Sockets 7.8 Utility Routines for Integer Conversion 7.9 Using Socket Calls in A Program (The socket can be created in any language) Extra Reading: Client-Server Architecture and its implementation using Socket programming	**12**	**6**
	Total:	**100**	**45**

List of Practical assignments (Socket Programming):

1. Write the client and server programs for establishing termination of connection between client and server using TCP. Assume the server can handle only one client.
2. Write the client and server programs for simple data (hello) transfer between client and server using UDP. Client will send hello server message to the server program. In its reply the server will send hello client message. The server and client programs should reside on different computers in a network.
3. Write the client and server programs for connectionless communication between two different computers in the same TCP/IP network. The server process receives a byte from the client process should and send back an should acknowledgement to the client process.
4. Write program for implementing the sliding window protocol of window size 5.
5. Write the client and server program for implementing the broadcasting in the local network.

Course References:

Recommended Books:

Text Books:

1. Network Essential Notes GSW MCSE Study Notes
2. Internetworking Technology Handbook CISCO System
3. Data and Computer Communication 8th Edition – William Stallings
4. Official Certification guide CCNA 200-301
5. TCP/IP Sockets in JAVA, Practical Guide for Programmers, Kenneth L Calvert, Michael J Donahoo

Reference Books:

1. Data Communication and Networking Behroz A.Forouzan, TMH, 4th Edition
2. Computer Networks and Internets with Internet Applications Douglas Comer
3. Cryptography and Network Security Atul Kahate, TMH 2nd Edition
4. Internetworking With TCP/IP Vol III: Client-Server Programming and Applications BSD Socket Version Second Edition

Recommended Learning Material:

1. https://docs.oracle.com.javase/tutorial/networkingindex.html
2. https://docs.oracle.com/javase/tutorial/networking/overview/networking.html

Recommended Certifications:

1. CISCO Networking Basics Specialization (Coursera) -
2. Network Protocols and Architecture (Coursera)
3. Data Communications and Network Services (Coursera)
4. Computer Networking-Digital Network Security (www.alison.com/course)
5. CCNA (200-301)
6. CCNP Enterprise (300-401 ENCOR)
7. CCNP Security (300-700 SCOR +concentration exam)
8. CCIE Enterprise Infrastructure(300-401 ENCOR + Infrastructure lab v1.0)
9. CCIE Enterprise Wireless CCIE +(300-401 ENCOR +wireless lab v1.0)
10. Microsoft Networking Fundamentals (98-366)
11. Microsoft Security Fundamentals (98-367)

Course Code: IT-11L
Course Name: Practicals

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
-	10 Hrs./Week	5	-	75	-	50	125

Course Description:

This Practical course contains 3 sections. –

1. JavaScript Syllabus
2. GITHUB Syllabus (Not for Evaluation)
3. List of Practicals – Java Programming
4. List of Practicals – Data Structure and Algorithm

Course Outcomes:

Student will be able to

CO1: Demonstrate Collection framework (Apply)
CO2: Develop GUI using awt and swing (Apply)
CO3: Develop Web application using JSP and Servlet, JDBC (Apply)
CO4: Apply Data Structure to solve problems using JavaScript (Apply)

Course Structure:

Syllabus for JavaScript

Sr. No.	Topics Details
1	**Introduction to JavaScript -** History, Features, Application of JavaScript, JavaScript Basics –Data Types, Variables, Identifiers, Constants, Comments, Operators in JavaScript
2	Control and looping structure
3	Array – Concept, Types and Methods
4	Java Script Objects – Object ,Date ,String, Array , Math, Number, Boolean
5	Event handling-Mouse, Keyboard, Form, Window
6	JavaScript BOM –Browser object, Window, Location, Navigator, History Object
7	JavaScript DOM –Document Object and its Methods
8	Form Validations in JavaScript
9	Exception Handling in JavaScript using JavaScript

Syllabus for GIT/GITHUB

Sr. No.	Topics Details
1	Introduction to versioning systems
2	creating repositories [local & cloud based]
3	git commands
4	branching & merging
5	conflict resolution (3-way merge)
6	rebasing
7	version control [tagging]

Session on GIT/GITHUB is not for evaluation. It may be conducted by using Demonstration method. The objective of these sessions is "*Students must able to submit their mini-project on GITHUB as project repositories*"

List of Practicals – Java Programing

1. Installation of jdk enviorment & following utilities. What is javac , javap and javadoc.
2. Design an application by using array.
3. Implementation of package, Interface and abstract class
4. Design application using String, StringBuilder, StringTokenizer
5. Test any five of standard exception and user Defined Custom Exceptions in java
6. Threads creation and design applications by using Extending the Thread class/ Implementing the Runnable Interface. Application of multithreading in java.
7. Design java application using Collection in java such as Array List, Link List
8. Design GUI based java application using AWT, Swing with Event Handling.
9. Design a and implement JDBC applications.
10. Design and implement servlet applications.
11. Design and implement JSP applications

List of Practicals – Data Structure and Algorithm Practicals

Following practical must be implemented using JavaScript

1. Demonstrate singly and doubly linked list
2. STACK implementation using Array with PUSH, POP operations
3. Reverse a string using stack
4. Check for balanced parentheses by using Stacks
5. Implement Stack using Linked List
6. Demonstration of Linear Queue, Circular Queue, Priority Queue
7. Reverse stack using queue
8. Practical based on binary search tree implementation with its operations
9. Graph implementation and graph traversals
10. Implementation of Hashing
11. Practical based on Brute Force technique
12. Practical based on Greedy Algorithm-Prim's/Kruskal's algorithm

13. Practical based on Divide and Conquer Technique-Binary Search, Tower of Hanoi
14. Implementation of Dynamic Programming- LCS, Regular Expression Matching
15. Practical based on backtracking- N Queen's problems

Course Code: ITC11
Course Name: Mini Project

Credit Scheme			Evaluation Scheme				
Lecture	Practical	Credit	Internal			External	Total
			Written	Practical	Tutorial		
-	10 Hrs./Week	5	-	75	-	50	125

Course Description:

A mini project is an assignment that the student needs to complete at the end of every semester to strengthen the understanding of fundamentals through effective application of the subjects learnt.

Course Outcomes:

Student will be able to

CO1: Create working project using tools and techniques learnt in this semester (Create)

Course Structure:

Guidelines for Mini Project

1. Students are expected to undertake one mini project starting from first semester till third semester.
2. The student may take up the mini project in first semester based on the courses learnt in that semester and for every next semester the mini project may be based on the courses learnt in the current semester along with all the subjects learnt in earlier semesters.
3. The student may take up the project individually or in group. However, if project is done in group, each student must be given a responsibility for distinct modules.
4. Selected project/module must have relevant scope as per the marks assigned and must be carried out in the Institute.
5. Internal guide should monitor and evaluate the progress of the project on individual basis through handwritten workbook (Project Diary) maintained by students containing various project milestones with learnings and remarks from internal guide for concurrent evaluation.
6. The Project Synopsis should contain an Introduction to Project clearly stating the project scope in detail justifying enough scope for 125 marks. The project work will carry 75 marks for internal assessment and 50 marks for external assessment.
7. Students are expected to show working demo of the project during final evaluation.
8. **Students are expected to upload mini-project on GITHUB as project repository of the institution.**
9. Students are expected to submit the soft copy of mini project report as a part of final submission.

10. The project will be assessed internally as well as externally by the examiners appointed by University. University may appoint Industry Experts as an external examiner

Semester II

Course Code: IT-21
Course Name: Python Programming

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	25	-	-	50	75

Course Description:

Prerequisite:

Object oriented Concepts.

Course Objectives:

1. To understand and use the basic of python.
2. To understand advance concepts of python and able to apply it for solving the complex problems.
3. To understand the reading and writing data through file handling.
4. To understand basic database concepts in python.
5. To develop the critical thinking and analytical approach by using python libraries.

Course Outcomes:

Student will be able to

CO1: Understand Demonstrate the concepts of python and modular programming. (Understand)
CO2: Apply the concepts of concurrency control in python (Apply)
CO3: Solve the real-life problems using object-oriented concepts and python libraries (Apply)
CO4: Demonstrate the concept of IO, Exception Handling, database (Apply)
CO5: Analyze the given dataset and apply the data analysis concepts and data visualization. (Analyze)

Course Structure:

Unit No.	Topics Details	Weightage in %	No of Sessions
1	1. Introduction & Components of Python 1.1. Understanding Python 1.2. Role of Python in AI and Data science 1.3. Installation and Working with Python 1.4. The default graphical development environment for Python - IDLE 1.5. Types and Operation 1.6. Python Object Types-Number, Strings, Lists, Dictionaries, Tuples, Files, User Defined Classes	15	7

	1.7. Understanding python blocks 1.8. Python Program Flow Control 1.9. Conditional blocks using if, else and elif 1.10. Simple for loops in python 1.11. For loop using ranges, string, list and dictionaries 1.12. Use of while loops in python 1.13. Loop manipulation using pass, continue, break and else 1.14. Programming using Python conditional and loops block Extra Reading: Python installation with windows, Linux and MAC OS, creating virtual environment, configuring python on EC2 instance, understanding python IDE –[VSCode, PyCharm, Spyder], Installing Anaconda and setting up environment for python		
2	2. Python Functions, Modules & Packages 2.1. Function Basics-Scope, nested function, non-local statements 2.2. built-in functions 2.3. Arguments Passing, Anonymous Function: lambda 2.4. Decorators and Generators 2.5. Module basic usage, namespaces, reloading modules. – math, random, datetime, etc. 2.6. Package: import basics 2.7. Python namespace packages 2.8. user defined modules and packages Extra Readings: GUI framework in python	15	7
3	3. Python Object Oriented Programming 3.1. Concept of class, object and instances, method call 3.2. Constructor, class attributes and destructors 3.3. Real time use of class in live projects 3.4. Inheritance, super class and overloading operators, 3.5. Static and class methods 3.6. Adding and retrieving dynamic attributes of classes 3.7. Programming using OOPS 3.8. Deligation and container Extra Readings: Integrating GUI framework with OOP	15	6
4	4. Python Regular Expression 4.1. Powerful pattern matching and searching 4.2. Power of pattern searching using regex in python 4.3. Real time parsing of data using regex 4.4. Password, email, URL validation using regular expression 4.5. Pattern finding programs using regular expression	10	4

	Extra Readings: Web scrapping and pattern matching with regex		
5	5. Python Multithreading and Exception Handling 5.1. Exception Handling 5.2. Avoiding code break using exception handling 5.3. Safe guarding file operation using exception handling 5.4. Handling and helping developer with error code 5.5. Programming using Exception handling 5.6. Multithreading 5.7. Understanding threads 5.8. Synchronizing the threads 5.9. Programming using multithreading Extra Readings: Multiprocessing, deadlock, synchronization, monitors and messaging queue	10	5
6	6. Python File Operation 6.1. Reading config files in python 6.2. Writing log files in python 6.3. Understanding read functions, read(), readline() and readlines() 6.4. Understanding write 6.5. functions write() and writelines() 6.6. Manipulating file pointer using seek 6.7. Programming using file operations Extra Readings: Reading and writing the files on AWS S3 bucket	5	2
7	7. Python Database Interaction 7.1. Introduction to NoSQL database 7.2. Advantages of NoSQL database 7.3. SQL Vs NoSQL 7.4. Introduction to MongoDB with python 7.5. Exploring Collections and Documents 7.6. Performing basic CRUD operations with MongoDB and python Extra Readings: Graph database like Neo4j with python	10	5
8	8. Python for Data Analysis 8.1. NumPy: 8.2. Introduction to NumPy 8.3. Creating arrays, Using arrays and Scalars 8.4. Indexing Arrays, Array Transposition 8.5. Universal Array Function 8.6. Array Input and Output 8.7. Pandas: 8.8. What are pandas? Where it is used? 8.9. Series in pandas, pandas DataFrames, Index objects, ReIndex	20	9

	8.10. Drop Entry, Selecting Entries 8.11. Data Alignment, Rank and Sort 8.12. Summary Statics, Missing Data, Index Hierarchy 8.13. Matplotlib: 8.14. Python for Data Visualization 8.15. Introduction to Matplotlib 8.16. Visualization Tools Extra Readings: Text analytics with NLP and python		
	Total:	**100**	**45**

Course References:

Recommended Books:

Text Books:

Introduction to Python Programming, By Gowrishankar S, CRC Press

Reference Books:

1. Learning Python 5th ed. by Mark Lutz
2. Python: The Complete Reference by Martin C. Brown
3. Python Data Analytics: With Pandas, NumPy, and Matplotlib 2nd ed. Edition by Fabio Nelli
4. Core Python Programming by Wesley J. Chun Publisher: Prentice Hall
5. Python Programming: A modular approach by Taneja Sheetal, Kumar Naveen
6. Beginner's Guide to Python Programming: Learn Python 3 Fundamentals, Plotting and Tkinter GUI Development Easily by Serhan Yamacli
7. Programming Python, O'reilly, by Mark Lutz
8. Learning Python, O'reilly, Mark Lutz
9. Head First Python, O'reilly, By Paul Barry

Recommended Certifications:

1. Programming, Data Structures and Algorithms Using Python https://swayam.gov.in/nd1_noc19_cs40/preview
2. Data Analytics with Python https://swayam.gov.in/nd1_noc20_cs46/preview

Course Code: IT-22
Course Name: Software Project Management

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	10	10	5	50	75

Course Description:

Prerequisite:

Basic Software Engineering process

Course Objectives:

1. To learn process of Software Project Management.
2. To Study role of Project Manager in Project Management.
3. To learn Agile Project Management Framework.
4. To study various role of Agile Team and Tools.
5. To understand project planning and tracking.

Course Outcomes:

Student will be able to

CO1: Understand the process of Software Project Management Framework and Apply estimation techniques. (Apply)

CO2: Learn the philosophy, principles and lifecycle of an agile project. (Understand)

CO3: Demonstrate Agile Teams and Tools and Apply agile project constraints and trade-offs for estimating project size and schedule (Apply)

CO4: Explain Project Tracking and Interpretation of Progress Report (Understand)

CO5: Analyze Problem statement and evaluate User Stories (Analyze)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	1. Linear Project Management Framework 1.1 Overview of project Management 1.2 Project management life cycle-IEEE Life Cycle 1.3 Project Management Process 1.4 Role of Project Manager 1.5 Quality Metrics 1.6 Risk Management Process (Case Study Based) 1.6.1 Risk Identification 1.6.2 Risk Analysis 1.6.3 Risk Mitigation 1.6.4 RMMM 1.7 Hands on MS Project Tool– Resource	15	6

	Allocation, Scheduling, Gannt chart Note: Case studies based on Risk Management, MS Project tool & Gannt Chart Extra Reading: Different software project management, Types of Risk, Risk Information sheet (RIS), CPM and PERT		
2	**2. Linear Software Project Estimation** 2.1 Different methods of Cost estimation 2.1.1 COCOMO-I & II model (Problem Statement) 2.1.2 Delphi cost estimation 2.2 Function Point Analysis (Problem Statement) 2.3 The SEI Capability Maturity Model CMM 2.4 Software Configuration management Note: Case studies/Numerical Problems based on COCOMO-I and FPA Extra Reading: KLOC, Rayleigh Curve, Change Management, Configuration management tool - SVN Tool or Redmine	20	8
3	**3. Agile Project Management Framework** 3.1 Introduction and Definition Agile, Agile Project Life Cycle 3.2 Agile Manifesto: History of Agile and Agile Principles 3.3 Key Agile Concepts: 3.3.1 User stories, Story points 3.3.2 Product Backlog 3.3.3 Sprint Backlog, 3.3.4 Sprint Velocity 3.3.5 Swim lanes 3.3.6 Minimum Viable Product (MVP) 3.3.7 Version and Release 3.4 Agile Project Management v/s Traditional Project Management Note: Case studies based on agile vs. traditional project Extra Reading: Study Scrum Agile Framework, Agile project management delivery & methodology framework, Software project team management and different team structures	15	10
4	**4 Agile Teams, Size and Schedule** 4.1 Dynamic System Development Method	15	10

	4.2 Value-Driven Development 4.3 Team and roles of an Agile Team 4.3.1 Scrum Master 4.3.2 Product Owner 4.3.3 Development Team 4.4 Product Vision and Product Roadmap 4.5 Project Objective and Key Metrics 4.6 Introduction to User Stories 4.7 Estimate the Product Backlog 4.8 Techniques for estimating Story Points 4.9 Plan Product Releases 4.10 Product Prioritization Note: Case studies based on Estimation of Product backlog & Story points, design your team and Add screenshots with the caption, Design User stories, log efforts and task in detail Extra Reading: Personnel Management, Release & iteration planning, eXtreme Programming (XP), Values and Principles, Team Dynamics and Collaboration		
5	**5.Tracking Agile Project and Reports** 5.1 Introduction 5.2 Plan and Execute Iteration 5.3 Facilitate Retrospective, Making Team Decisions and Closing out Retrospective 5.4 Agile Reports 5.4.1 Daily Reports 5.4.2 Sprint Burn down Chart and Reports 5.5 Benefits of Agile Project Management Note: Case studies based on No. of iterations and Project Report, Sprint Chart Extra Reading: Use of MS Project to track agile project, Agile project management tools, Feature-Driven Development, Agile Metrics	20	5
6	**6. Implementation with Agile Tools** 6.1 Introduction of Agile Tools 6.2 Hands on GitHub 6.2.1 Create Project using Kanban 6.2.2 Project Repositories 6.2.3 Continuous Integration 6.2.4 Project Backlog 6.2.5 Team Management 6.2.6 Progress Tracking	15	6

	6.2.7 Releases 6.3 Implementation of Problem statement with Agile Tools- GitHub 6.3.1Designing Product Vision, Product Backlog, 6.3.2 Sprint Backlog, Estimate Story Points 6.3.3 Iteration Release Note: Case study on design of product vision & backlog with features and user stories, Estimation of story points, Design Iteration Plan, Iteration progress and close iteration in detail Extra Reading: Agile modeling, Explore various Agile Tools		
	Total:	**100**	**45**

List of Practical's (if any)

1. Design Project Management plan template by using MS-Project tool. (Resource allocation, Scheduling, Cost Calculation and Gantt Chart)
2. Create project plan using agile methodologies for the development of web page of Library Management System as a minimum viable product using 3 resources as per sprint planning.
3. Calculate the effort to execute the task and prioritize the task to execute in the current sprint and keep rest of the task in backlog.
4. Demo of the task developed by the developer in the Sprint.
5. Retrospective to discuss about the short coming and improvement of the design and execution of the Sprint task.
6. Check in the developed code in the GitHub repository.

Course References:

Recommended Books:

Text Books:

1. Software engineering principles and practice, McGraw-Hill, Waman S. Javadekar
2. Software Engineering by Pressman
3. Agile Project Management for Dummies, 2nd Edition
4. Coaching Agile Teams: A Comparison for ScrumMasters, Agile Coaches, and Project Managers in Transition, Lyssa Adkins
5. Agile Project Management: Creating Innovative Products (2nd Edition) by Jim Highsmith, Addison-Wesley Professional

Reference Books:

1. Mark C. Layton, Steven J. Ostermiller
2. Agile Estimating and Planning by Mike Cohn Robert C Martin Series
3. Introduction to Software Project Management by Adolfo Villafiorita, CRC Press
4. Agile Project Management with Scrumby Ken Schwaber, Microsoft Press © 2004
5. Agile Project Management QuickStart Guide : The Simplified Beginners Guide to Agile Project Management by ClydeBank Business
6. Agile Product Management with Scrum: Creating Products that Customers Love by Roman Pichler.
7. Scrum Mastery: From Good to Great Servant-Leadership by Geoff Watts
8. Agile Project Management for Dummies by Mark C. Layton
9. The Agile Enterprise: Building and Running Agile Organizations by Mario E. Moreira
10. Scrum: The Art of Doing Twice the Work in Half the Time by Jeff Sutherland
11. Essential Scrum: A Practical Guide to the Most Popular Agile Process by Kenneth S. Rubin
12. Agile Project Management with Kanban By Eric Brechner
13. Agile Constraints: Creating and Managing Successful Projects with Scrum, Multiple authors

Recommended Learning Material:

1. https://learning.tcsionhub.in/
2. https://www.agilealliance.org
3. http://www.pmi.org
4. https://github.com/topics/kanban
5. https://www.opensourcescrum.com/
6. https://www.scrum.org/resources
7. https://www.tutorialspoint.com/agile/index.htm
8. https://www.atlassian.com/agile
9. https://www.javatpoint.com/agile
10. https://www.guru99.com/agile-testing-course.html
11. https://www.visual-paradigm.com/tutorials/agile-tutorial/

Recommended Certifications:

1. Project Management Professional (PMP)
2. PMI-ACP(Agile Certified Practitioner)
3. Associate in Project Management
4. BVOP Certified Project Manager
5. Certified Associate in Project Management (CAPM)
6. Certified Project Director

7. Certified Project Management Practitioner (CPMP)
8. Certified Project Manager (CPM)
9. Certified ScrumMaster (CSM)
10. CompTIA Project+
11. Master Project Manager (MPM)
12. PRINCE2 Foundation/PRINCE2 Practitioner
13. Professional in Project Management (PPM)
14. Project Management in IT Security (PMITS)
15. APMG International
16. Strategyex Certificate (Associate or Master's) in Agile
17. International Consortium for Agile (ICAgile)
18. Agile Certification Institute
19. Scaled Agile Academy
20. Scrum Alliance
21. Certified Agile Project Manager (IAPM)

Course Code: MT-21
Course Name: Optimization Techniques

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	10	10	5	50	75

Course Description:

Prerequisite:

Basic mathematical knowledge is essentials.

Course Objectives:

1. To understand the role and principles of optimization techniques in business world.
2. To understand the process of problem statement formulation of the business scenario.
3. To understand the implementation of various decision-making techniques in the process of decision making.
4. To gain the techniques and skills on how to use optimization techniques to support the decision making in business world.

Course Outcomes:

Student will be able to

CO1: Understand the role and principles of optimization techniques in business world (Understand)

CO2: Demonstrate specific optimization technique for effective decision making (Apply)

CO3: Apply the optimization techniques in business environments (Apply)

CO4: Illustrate and infer for the business scenario (Analyze)

CO5: Analyze the optimization techniques in strategic planning for optimal gain. (Analyze)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Linear Programming** 1.1. Various definitions, statements of basic theorems and properties, Advantages and Limitations, 1.2. Application areas of Linear programming 1.3. Linear Programming – Concept 1.4. Simplex Method and Problems 1.5. Two Phase Simplex Method and problems, Note: Case study-based problems	20	10

	Extra Readings: Formulation of Linear programming, Solution of LPP using Graphical method		
2	**2. Markov Chains & Simulation Techniques:** 2.1 Markov chains: Applications related to technical functional areas, 2.2 Steady state Probabilities and its implications, 2.3 Decision making based on the inferences Monte Carlo Simulation. Extra Readings: Application of Markov chain in Queuing theory, Simulation techniques used in Machine learning and bioinformatics	15	7
3	**3. Sequential model and related Problems** 3.1 Processing n jobs through 2 machines 3.2 Processing n jobs through 3 machines 3.3 Processing n jobs through m machine Extra Readings: Processing of n jobs through m machines	15	6
4	**4. PERT & CPM** 4.1 Basic differences between PERT and CPM. 4.2 Network diagram 4.3 Time estimates (Forward Pass Computation, Backward Pass Computation 4.4 Critical Path 4.5 Probability of meeting scheduled date of completion, 4.6 Calculation on CPM network. 4.7 Various floats for activities 4.8 Event Slack 4.9 Calculation on PERT network. 4.10 Application of schedule based on cost analysis and crashing 4.11 Case study-based problems Extra Readings: Optimal Cost estimation by crashing the network, Explore the MS Project tool.	20	10
5	**5. Game Theory** 5.1 Introduction 5.2 n X m zero sum game with dominance 5.3 Solution using Algebraic, Arithmetic and Matrix strategy Extra Readings: Learn the difference between Sequential and Simultaneous game	15	6

6	**6. Decision Analysis** 6.1 Introduction to Decision Analysis 6.2 Types of Decision-making environment 6.3 Decision making under uncertainty and under risk 6.4 Concept of Decision Tree Extra Readings: Decision models in Econometrics and computer science	15	6
	Total:	**100**	**45**

List of Practicals (if any)

Practicals to be conducted on the following topics. It is expected that, Applications to be covered using Python and /or R.

1. Linear Programming
2. Markov Chain and Simulation Techniques
3. Sequential models and related problems
4. CPM and PERT
5. Game Theory
6. Decision Analysis

Course References:

Recommended Books:

Text Books:

1. Operations Research by Pannerselvam
2. Operations Research Theory and Application by J. K. Sharma –Mac-Millan Publication
3. Statistical and Quantative Methods – Mr. Ranjit Chitale

Reference Books:

1. Statistical Methods – S.P.Gupta, Sultan Chand, New Delhi
2. Operation Research by V. k. Kapoor
3. Operations Research by Kanti Swaroop, P. K. Gupta and Man Mohan
4. Introduction to Operations Research by Hiller & Lieberman, Tata Mc Graw Hill
5. Operations Research by H. A. Taha
6. Operation Research by Hira & Gupta
7. What is Game Theory?, David K. Levine, Economics, UCLA
8. Recommended Learning Material:

Research Software:

1. MS Excel Solver
2. TORA
3. Python and / or R programming

Websites:

1. www.orsi.in
2. www.atozoperationalresearch.com

Websites for practical sessions:

1. https://towardsdatascience.com/linear-programming-and-discrete-optimization-with-python-using-pulp-449f3c5f6e99
2. https://github.com/topics/operations-research?l=python
3. https://github.com/Gabeqb/Linear-Programming-With-Python/commit/a61be0d5fc8e66dd38f3d094bb80cef6a9a04152

Journals:

1. International Journal of Operations Research and Management science
2. International Journal of Operations and Quantitative Management
3. Indian Journal of Advance Operations Management.

Recommended Certifications:

1. Data science with Python
2. Data science with R programing
3. Certification in Machine Learning
4. Certification in Tableau

Course Code: IT-23
Course Name: Advanced Internet Technologies

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	25	-	-	50	75

Course Description:

Prerequisite:

Student must have hands-on working knowledge of HTML, CSS, JavaScript and Angular JS

Course Objectives:

1. To impart the design, development and implementation of Dynamic Web Pages.
2. To implement the Latest properties of CSS3
3. To implement the Concept of NodeJS.
4. To develop programs for Web using Angular and SPA.
5. To design and implement dynamic websites with good sense of designing and latest technical aspects.

Course Outcomes:

Student will be able to

CO1: Outline the basic concepts of Advance Internet Technologies (Understand)
CO2: Design appropriate user interfaces and implements webpage based on given problem Statement (Apply)
CO3: Implement concepts and methods of NodeJS (Apply)
CO4: Implement concepts and methods of Angular (Apply)
CO5: Build Dynamic web pages using server-side PHP programming with Database Connectivity (Apply)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Introduction to HTML5** 1.1. Basics of HTML5 – Introduction, features, form new elements, attributes and semantics in HTML5 1.2. <canvas>, <video>, <audio>. 1.3. Introduction to Scalable Vector Graphics (SVG) 1.4. Introduction to Version compatibility 1.5. Installation of Apache Tomcat (Xampp/Lampp/MySQL)	10	5

	Extra Reading: Geo location, Drag, Drop, Web Storage		
2	**2. Introduction to CSS3** 2.1. Architecture of CSS 2.2. CSS Modules 2.3. CSS Framework 2.4. Selectors and Pseudo Classes 2.5. Fonts and Text Effects 2.6. Colors, Background Images, and Masks Extra Reading: Transitions, Transforms and Animations Embedding Media, Gradients, Bootstrap	10	5
3	**3. Node JS** **3.1.** introduction and how it works 3.2. installation of node js 3.3. REPL 3.4. NPM 3.5. How modules work 3.6. Webserver Creation 3.7. Events Extra Reading: Node.js with MySQL	25	12
4	**4. Angular (Latest Stable Version)** 4.1. Introduction (Features and Advantage) 4.2. Type Script 4.3. Modules 4.4. Components 4.5. Directives, Expression, Filters 4.6. Dependency Injection 4.7. Services 4.8. Routing 4.9. SPA (Single Page Application) Extra Reading: Data binding, property binding, Event Binding, Two-way data binding, String Interpolation.	25	12
5	**5. PHP** 5.1. Installing and Configuring PHP 5.2. Introduction 5.2.1. PHP and the Web Server Architecture, PHP Capabilities	30	11

	5.2.2. PHP and HTTP Environment Variables 5.2.3. Variables 5.2.4. Constants 5.2.5. Data Types 5.2.6. Operators 5.2.7. Working with Arrays 5.3. Decision Making, Flow Control and Loops 5.4. Introduction to Laravel 5.5. Creating a Dynamic HTML Form with PHP 5.6. Database Connectivity with MySQL 5.6.1. Performing basic database operations (CRUD) 5.7. Using GET, POST, REQUEST, SESSION, and COOKIE Variables Extra Reading: Sending Emails, PHP with AJAX and XML, Payment Gateway Integration		
Total:		**100**	**45**

Course References:

Recommended Books:

Text Books:

1. Complete reference HTML, TMH
2. HTML5 & CSS3, Castro Elizabeth 7th Edition
3. Beginning Node.js by Basarat Ali Syed
4. Angular: Up and Running- Learning Angular, Step by Step by Shyam Seshadri
5. Beginning PHP, Apache, MySQL web development

Reference Books

1. Introducing HTML5 - Bruce Lawson, Remy Sharp
2. Node.js in Action, 2ed by Alex Young, Bradley Meck
3. Mastering Node.js by Pasquali Sandro
4. Angular Essentials by Kumar Dhananjay
5. Complete Ref. PHP

Recommended Learning Material:

- MOOC Courses

1) **Introduction to HTML5 – University of Michigan**
https://www.coursera.org/learn/html

2) **Introduction to Web Development – University of California**
   https://www.coursera.org/learn/web-development
3) **HTML, CSS and JavaScript for Web Developers – Johns Hopkins University**
   https://www.coursera.org/learn/html-css-javascript-for-web-developers
4) **Web Design for Everybody: Basics of Web Development & Coding Specialization – University of Michigan**
   https://www.coursera.org/specializations/web-design
5) **Introduction to CSS3 – University of Michigan**
   https://www.coursera.org/learn/introcss
6) **Server-side Development with NodeJS, Express and MongoDB – The Hong Kong University of Science and Technology**
   https://www.coursera.org/learn/server-side-nodejs
7) **Front-End Web UI Frameworks and Tools: Bootstrap 4 – The Hong Kong University of Science and Technology**
   https://www.coursera.org/learn/bootstrap-4
8) **Front-End JavaScript Frameworks: Angular – The Hong Kong University of Science and Technology**
   https://www.coursera.org/learn/angular
9) **Single Page Web Applications with AngularJS – John Hopkins University**
   https://www.coursera.org/learn/single-page-web-apps-with-angularjs
10) **Building Web Applications in PHP – University of Michigan**
    https://www.coursera.org/learn/web-applications-php
11) **Building Database Applications in PHP – University of Michigan**
    https://www.coursera.org/learn/database-applications-php
12) **Web Applications for Everybody Specialization**
    https://www.coursera.org/specializations/web-applications

- Other Learning Material

  - **HTML 5, CSS3, JavaScript**
    - https://www.htmldog.com/
    - https://www.w3schools.com/
    - https://qhmit.com/
    - http://www.landofcode.com/
    - https://www.codecademy.com/
    - http://www.echoecho.com/html.htm
    - https://www.awwwards.com/
  - **Bootstrap**
    - https://www.w3schools.com/bootstrap4/
    - https://getbootstrap.com/
    - https://www.freecodecamp.org/news/learn-bootstrap-4-in-30-minute-by-building-a-landing-page-website-guide-for-beginners-f64e03833f33/
    - https://www.freecodecamp.org/news/want-to-learn-bootstrap-4-heres-our-free-10-part-course-happy-easter-35c004dc45a4/
  - **NodeJS**
    - https://nodejs.org/en/docs/guides/
    - https://www.w3schools.com/nodejs/
    - https://www.nodebeginner.org/
    - http://visionmedia.github.io/masteringnode/

  - **Angular**

- https://www.c-sharpcorner.com/topics/angular-8
- https://www.javatpoint.com/angular-8

❖ **PHP**

- https://www.php.net/manual/en/index.php
- https://phptherightway.com/
- https://www.tutorialspoint.com/php/php_useful_resources.htm
- https://www.w3schools.com/php/

Recommended Certifications:

1. Microsoft HTML5 and CSS3 (https://www.microsoft.com/en-us/learning/exam-70-480.aspx)
2. Certification available on Coursera and Udemy.

Course Code: IT-24
Course Name: Advanced DBMS

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
3 Hrs./Week	-	3	15	10	-	50	75

Course Description:

Prerequisite:

Basics of Database Concepts

Course Objectives:

1. To understand core concepts of database management system and its types
2. To provide database design approaches using E-R model and normalization
3. To discuss transaction management and concurrency control
4. To gain an awareness of the structure of object-oriented database and its applications
5. To gain familiarization of Database crash, recovery concepts and security issues
6. To Demonstrate SQL, XML schema and NO SQL database

Course Outcomes:

Student will be able to

CO1: Describe the core concepts of DBMS and various databases used in real applications (Understand)

CO2: Design relational database using E-R model and normalization (Apply)

CO3: Demonstrate XML database and nonprocedural structural query languages for data access (Apply)

CO4: Explain concepts of Parallel, Distributed and Object-Oriented Databases and their applications (Understand)

CO5: Apply transaction management, recovery management, backup and security – privacy concepts for database applications (Apply)

Course Structure:

Sr. No.	Topics Details	Weightage in %	No of Sessions
1	**1. Introduction DBMS – Concepts & Architectures** 1.1 Database and Need for DBMS, Characteristics of DBMS 1.2 Database 3-tier schema (ANSI/SPARC) and system architecture of DBMS 1.3 Views of data- Schemas and instances, Data Independence	10	4

	1.4 Centralized, Client-Server system, Transaction servers, Data servers, Cloud based servers Extra Reading: Indexing and Hashing - Basic concepts of indexing, ordered index, B+ tree index, B+ tree extensions, Multiple key access, Hashing concepts, types of hashing, Bitmap indices		
2	**2. Data Modelling and Relational Database Design** 2.1 Data Modelling using ER Diagram: Representation of Entities, Attributes, Relationships and their Type, Cardinality, Generalization, Specialization, Aggregation. 2.2 Relational data model: Structure of Relational Database Model, Types of keys, Referential Integrity Constraints 2.3 Codd's rules 2.4 Database Design using E-R, E-R to Relational 2.5 Normalization – Normal forms based on primary (1 NF, 2 NF, 3NF, BCNF) Note: Case studies based on E-R diagram & Normalization Extra Reading: Database languages - Relational Algebra, Relational database languages, Data definition in SQL, Views and Queries in SQL, Joins, specifying constraints and Indexes in SQL, Specifying constraints management systems Postgres/ SQL/MySQL	16	8
3	**3. Transaction and Concurrency control** 3.1. Concept of transaction, ACID properties, States of transaction 3.2. Concurrency control, Problems in concurrency controls 3.3. Scheduling of transactions, Serializability and testing of serilaizibity 3.4. Lock-based Protocol and Time stamp-based ordering protocols 3.5. Deadlock Handling Extra Readings: Semantic data controls & Multi-version concurrency control	13	6
4	**4. Parallel Databases** 4.1. Introduction to Parallel Databases 4.2. Parallel Database Architectures 4.3. I/O parallelism 4.4. Inter-query and Intra-query parallelism 4.5. Inter-operational and Intra-operational parallelism	13	6

	4.6. Key elements of parallel database processing: Speed-up, Scale-up Synchronization and Locking Extra Readings: Parallel handling and Load balancing		
5	**5. Distributed Databases** 5.1. Introduction to Distributed Database System 5.2. Homogeneous and Heterogeneous Databases 5.3. Distributed data storage (Fragmentation and Replication) 5.4. Distributed transactions 5.5. Concurrency control schemes in DDBMS 5.6. Commit protocols 2 phase and 3 Phase Commit Protocol Extra Readings: Reliability issues in DDBMS and Web based interface of DDBMS	13	6
6	**6. Object Oriented Databases & Applications** 6.1. Overview of Object- Oriented Database concepts & characteristics 6.2. Database design for OODBMS – Objects, OIDs and reference type 6.3. Spatial data and Spatial indexing (Any two techniques) 6.4. Mobile Database: Need, Structure, Features, Limitations and Applications 6.5. Temporal databases, temporal aspects valid time, transaction time or decision time 6.6. Multimedia Database: Architecture, Type and Characteristics	10	4
7	**7. Crash Recovery and Backup** 7.1. Failure classifications 7.2. Recovery & Atomicity 7.3. Log based recovery 7.4. Checkpoint and Shadow Paging in Data recovery 7.5. Database backup and types of backups Extra Readings: Role and Functions of Database administrator	10	5
8	**8. Security and Privacy** 8.1. Database security issues 8.2. Discretionary access control based on grant & revoking privilege 8.3. Mandatory access control and role-based access control for multilevel security 8.4. Encryption & public key infrastructures	10	4
9	**9. NO-SQL Database** Introduction, Types of NOSQL, Need of NoSQL	5	2

	databases, Use Cases		
Total:		**100**	**45**

List of Practicals (if any)

1. To install and configure database software (ORACLE/MYSQL)
2. To design a database (logical & physical database)
3. To Perform all SQL operations and queries on designed physical database
4. To install and configure NO-SQL database and practice for core operations
5. To perform experiments on database crash and recovery
6. To perform experiments on database Backup – restoring operations on database server
7. To perform some operations on Object oriented databases

Course References:

Recommended Books:

Text Books:

1.
2. Introduction to database systems C.J. Date, Pearson.
3. Fundamentals of Database Systems by Elmasri Navathe
4. Principles of Database Management James Martin, PHI
5. Database System Concepts by Abraham Silberschatz, H. Korth, Sudarshan

Reference Books:

1. Database Management System by Raghu Ramakrishnan / Johannes Gherke
2. Database Management System (DBMS)A Practical Approach. By Rajiv Chopra
3. Database system practical approach to design, implementation & management by Connoly & Begg,
4. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence Martin Fowler

Recommended Certifications:

1. Oracle certified associate (OCA)
2. Oracle certified professional (OCP)
3. Database administrator (DBA)
4. Database related certification courses available at NPTEL/Coursera/Udemy

Course Code: IT-21L
Course Name: Practicals

Credit Scheme			Evaluation Scheme				
Lecture	**Practical**	**Credit**	**Internal**			External	Total
			Written	**Practical**	**Tutorial**		
-	10 Hrs./Week	5	-	75	-	50	125

Course Description:

This Practical course contains 2 sections. –

1. List of Practicals – Python Programming
2. List of Practicals – Advanced Internet Technologies

Course Outcomes:

Student will be able to

CO1: implement python programming concepts for solving real life problems. (Apply)
CO2: Implement Advanced Internet Technologies (Apply)

Course Structure:

List of Practicals – Python Programming

Note:

- Recommended IDE for python – IDLE
- Exception handling concepts should be used with file handling programs.

1. Python installation and configuration with windows and Linux
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. (Real life examples must be covered for the implementation of object-oriented concepts)
6. Programs for parsing of data, validations like Password, email, URL, etc.
7. Programs for Pattern finding should be covered.
8. Programs covering all the aspects of Exception handling, user defined exception, Multithreading should be covered.
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.

List of Practicals – Advanced Internet Technologies

1. Program to implement Audio and Video features for your web page.
2. Program to design form using HTML5 elements, attributes and Semantics.
3. Programs using Canvas and SVG.
4. Programs to demonstrate external and internal styles in the web page using font, text, background, borders, opacity and other CSS 3 properties.
5. Implement Transformation using Translation, Rotation and Scaling in your web page.
6. Program to show current date and time using user defined module
7. Program using built-in modules to split the query string into readable parts.
8. Program using NPM which will convert entered string into either case
9. Write a program to create a calculator using Node JS. (Install and configure Node JS and Server)
10. Write Program for Form validation in Angular.
11. Program to demonstrate the ngif, ngfor, ngswitch statements.
12. Create angular project which will demonstrate the usage of component directive, structural directive and attribute directives
13. Create angular project which has HTML template and handle the click event on click of the button (Installation of Angular and Bootstrap 4 CSS Framework)
14. Program for basic operations, array and user interface handling.
15. Program to demonstrate session management using various techniques.
16. Program to perform the CRUD Operations using PHP Script.

Course Code: ITC21
Course Name: Mini Project

Credit Scheme			Evaluation Scheme				
Lecture	Practical	Credit	Internal			External	Total
			Written	Practical	Tutorial		
-	10 Hrs./Week	5	-	75	-	50	125

Course Description:

A mini project is an assignment that the student needs to complete at the end of every semester to strengthen the understanding of fundamentals through effective application of the subjects learnt.

Course Outcomes:

Student will be able to

CO1: Create working project using tools and techniques learnt in this semester (Create)

Course Structure:

Guidelines for Mini Project

1. Students are expected to undertake one mini project starting from first semester till third semester.
2. The student may take up the mini project in first semester based on the courses learnt in that semester and for every next semester the mini project may be based on the courses learnt in the current semester along with all the subjects learnt in earlier semesters.
3. The student may take up the project individually or in group. However, if project is done in group, each student must be given a responsibility for distinct modules.
4. Selected project/module must have relevant scope as per the marks assigned and must be carried out in the Institute.
5. Internal guide should monitor and evaluate the progress of the project on individual basis through handwritten workbook (Project Diary) maintained by students containing various project milestones with learnings and remarks from internal guide for concurrent evaluation.
6. The Project Synopsis should contain an Introduction to Project clearly stating the project scope in detail justifying enough scope for 125 marks. The project work will carry 75 marks for internal assessment and 50 marks for external assessment.
7. Students are expected to show working demo of the project during final evaluation.
8. **Students are expected to upload mini-project on GITHUB as project repository of the institution.**
9. Students are expected to submit the soft copy of mini project report as a part of final submission.
10. The project will be assessed internally as well as externally by the examiners appointed by University. University may appoint Industry Experts as an external examiner
11.