

Assam Don Bosco University

Programme Project Report (PPR)

Bachelor of Computer Application

Introduction

Students who complete the distance-learning Bachelor of Computer Applications (BCA) programme will be prepared with the knowledge and abilities needed to succeed in the fast-paced profession of computer science. The goal of the BCA programme is to give students a thorough understanding of a variety of computer concepts, programming languages, software development, database administration, networking, and other important information technology topics. Students are given the tools they need to develop their problem-solving, critical thinking, and decision-making skills in the context of the quickly changing technological landscape through a combination of academic knowledge and real-world applications. With the help of the distance learning programme, students can obtain a top-notch education without having to travel to conventional brick-and-mortar schools, regardless of their background or location. This degree opens doors to a bright and promising future in the field of computer applications by providing a flexible and rigorous curriculum along with the convenience of distant learning.

The scope of BCA:

The scope of BCA extends beyond traditional industries, as technology is becoming integral to almost every sector, including finance, healthcare, education, entertainment, and more. BCA graduates can also opt for higher studies, such as pursuing a Master's in Computer Applications (MCA) or specialized certifications in areas like cloud computing, artificial intelligence, data science, and machine learning, to further enhance their skills and career prospects. Since technology is becoming more and more ingrained in virtually every industry including finance, healthcare, education, entertainment, the reach of BCA goes beyond conventional industries. To further improve their abilities and employment prospects, BCA graduates might choose to continue their education by obtaining a Master's in Computer Applications (MCA) or specialised certifications in fields including cloud computing, artificial intelligence, data science, and machine learning.

Career Opportunities

The scope of job opportunities varies depending on factors like the candidate's specialization, additional certifications, and work experience. Some of the potential job roles and career paths for BCA graduates are:

- Software Developer
- Web Developer
- Mobile App Developer
- Database Administrator
- System Analyst
- Network Administrator
- IT Support Specialist
- Cybersecurity Analyst
- Data Analyst
- Quality Assurance Analyst
- Digital Marketing Specialist

Teaching Pedagogy

Teaching pedagogy for a distance programme is crucial to ensure effective and engaging learning experiences for students who are not physically present in a traditional classroom setting. Some essential elements of teaching pedagogy for a distance BCA programme are:

- **Online Learning Platform:** An intuitive and trustworthy online learning platform that enables simple access to the course materials, lectures, assignments, and communication features. To meet the needs of various student populations, the platform is usable on a variety of devices.
- **Multimedia Content:** Interactive and captivating multimedia material, such as podcasts, infographics, video lectures, and animations. These resources support difficult concept explanation, accommodate different learning preferences, and keep students' interest.
- **Discussion Forums and Collaboration:** Through discussion boards platforms, it helps the students to communicate with teachers and one another, encouraging peer learning and problem-solving.
- **Virtual Classrooms and Webinars:** Use of live webinars and virtual classrooms to simulate in-person interactions with students. This makes it possible for teachers to give lectures, respond to inquiries, and run group activities all at once.

- **Self-paced Learning:** Allowing students to learn at their own pace, provides flexibility in the learning process. Give students access to lectures that have been recorded and asynchronous learning resources so they can review the topic at their convenience.
- **Regular Assessments:** Regularly provide formative and summative tests to pupils to monitor their learning. Exams, assignments, projects, and quizzes are used in combination to evaluate students' learning.
- **Personalized Support:** Offer students specialised assistance via email, online consultations, or private video conferences.
- **Feedback Mechanism:** Feedback from students on the structure, subject matter, and overall learning experience. Utilise this input to adjust the curriculum and deal with any problems that may occur.
- **Accessibility and Inclusivity:** Provides suitable modifications to make sure that all students, including those with disabilities, can access the course contents and resources.

Academic Counselling

Academic counselling is a critical aspect of a distance education programme as it plays a significant role in guiding and supporting students throughout their learning journey. Some essential elements of academic counselling for a distance BCA programme:

- **Personalized Guidance:** Offer individualised counselling sessions to better understand each student's particular requirements, goals, and difficulties.
- **Course Selection and Planning:** Assist students in making informed course choices and creating academic plans that are in line with their desired academic and professional outcomes. Provide them with information about prerequisites, credit requirements, and optional possibilities.
- **Orientation & Onboarding:** Hold thorough orientation sessions for new students to introduce them to the distance learning platform, course outline, evaluation procedures, and resource availability.
- **Monitoring Academic Progress:** Continually keep tabs on pupils' academic development and promptly give them performance evaluation. Offer methods for improvement and point out areas that need attention.
- **Time Management and Study Skills:** Help students reconcile their academic obligations with their personal and professional commitments by providing advice on time management approaches and efficient study practises.
- **Support for Learning Challenges:** Identify students who are having learning challenges and offer the necessary support, such as extra resources, tutoring programmes, or referrals to learning support centres.

- **Career Counseling:** Provide career counselling services to students so they can consider their career alternatives, pinpoint their skills, and match their educational pursuits with potential job trajectories.

Student Support Services

Some important student support services for a distance BCA programme:

- **Orientation and Onboarding:** Thorough orientation sessions are held to introduce new distance learners to the platform for learning, the structure of the course, the communication tools, and the resources that are available.
- **Academic Guidance:** Provide students with individualised academic guidance to help them plan their course choices, comprehend degree requirements, and develop a clear academic track suited to their needs and aspirations.
- **Technical Support:** Offer specialised technical assistance for any problems with the online learning environment, gaining access to the course materials, and other technology-related concerns.
- **Virtual Help Desk:** Virtual help desk or support center where students can reach out with their queries and concerns. Ensure prompt responses to help resolve issues efficiently.
- **24/7 Access to Resources:** To encourage self-paced learning, it is made sure that students always have access to the course materials, recorded lectures, e-books, and other educational resources.
- **Online library services:** Provide students with access to a sizable online library that includes e-books, journals, research databases, and citation tools to aid them in their academic work.
- **Career Services:** Provides aid with resume writing, job placement, and career counselling to help distance learners enter the workforce or advance in their professions.
- **Gathering student input through surveys or focus groups on a regular basis to determine how well the support services are working and where they might be improved.**
- **Financial Aid & Scholarships:** To help students manage their educational expenses, provide information and guidance on financial aid alternatives, scholarships, or tuition payment plans.

i. Programme's Mission and Objectives

Creating an intense teaching and research environment that moulds individuals into competent professionals who are innovative and committed to meet real world challenges.

Mission

1. To produce competent Computer Science professionals by promoting excellence in education and training.

2. To inculcate the spirit of self-sustainability through research, consultancy, development activities and lifelong learning.
3. To extend technical expertise to meet real world challenges and play a leading role in technical innovation, creativity and application development.
4. To infuse a sense of commitment in individuals for the betterment of the society through technology.

Programme Educational Objectives (PEOs)

1. To prepare the students to have strong foundation in computer science engineering with impetus to higher studies, consultancy, research and development.
2. To prepare the students to be self sustainable and proficient to meet the real world challenges ethically and responsibly, in service to socio-economic development of the society.
3. To inculcate the spirit of life-long learning, understanding, and applying new ideas and technologies to provide novel engineering solutions in the rapidly changing environment.

ii. Relevance of Program with the Assam Don Bosco University Mission & Goals

The programmes mission and objectives are in alignment with the University's mission, vision and goals, as detailed below.

Vision

The vision of Don Bosco University is:

'To mould young persons into intellectually competent, morally upright, socially committed and spiritually inspired citizens at the service of India and the world of today and tomorrow, by imparting holistic and personalized education.'

Guided by this vision and leveraging its century-old expertise in education in India and abroad, Don Bosco University is envisaged to be a centre of excellence in study and research focusing upon the following:

- Providing easier access to higher education for the under-privileged.
- Harmonizing technical excellence with human and religious values.
- Employment-oriented courses in emerging areas of contemporary technology and service.
- 'Teacher Education' as a privileged area of interest to accelerate the pace, reach and quality of education.
- Impetus to research initiatives with practical and social relevance.
- Providing a forum for debate and research on key human issues like religion & culture, peace & justice.
- Contributing to the socio-economic development of North-East India.
- Boosting international linkages and collaboration in university education.

The spirit of the university has been encapsulated in its emblem which has five distinct elements:

- The Cross' signifies its underlying Christian inspiration;
- 'The Profile of St. John Bosco' denotes its distinctive educational philosophy;

- 'The book emphasizes its commitment to academic excellence;
- 'The rays of the Rising Sun' stand for its dynamism and commitment to society; and,
- The motto, 'Carpe Diem' which meaning 'Seize the Day' challenging every Bosconian to grasp the opportunities presented by each new day and attain 'life in its fullness.'

Mission

*Built on a great legacy inherited from our founding fathers, our mission is to create an environment of stimulating intellectual dialogue across disciplines and harvest **knowledge with a cutting-edge through high quality teaching, research, and extension activities** leading to the generation of students who would provide leadership, vision and direction to society.*

Goal

*Our goal is to realize this vision by 2025. **Our strategy is to develop innovative programmes in basic and emerging disciplines** in a phased manner and to update them periodically so as to keep ourselves on track and on time. Our commitment is to involve the faculty and students in **interactive learning environment both within and outside the University through contextual and experiential programmes** so that they would be builders of a **worldwide-network of knowledge-sharing** and excel in their performance with a winning edge in the wider context of globalization.*

iii. Nature of Prospective Target Group of Learners

a) Identify target group:

A distance BCA (Bachelor of Computer Applications) programme may have a wide range of potential students. However, the desired target group's shared traits include the following:

- Working Professionals: People who are already employed but want to improve their expertise in computer applications to develop their careers or change fields.
- School graduates: Students who have graduated from high school and want to work in the field of computer applications but are unable to enrol in standard on-campus courses.
- Non-traditional students: Non-traditional students: Individuals who cannot enroll in a full-time programme on-campus due to obligations to their families or other obligations but yet want to acquire a degree in computer applications.
- Geographically remote students: People who live in rural or remote places with limited access to reputable educational institutions, making distance learning an attractive choice.
- Lifelong learners: Those who embrace lifelong learning and seek to upgrade their expertise in a variety of fields

b) How the curriculum is relevant to their learning needs

The curriculum of a distance BCA programme is created to adhere to the Choice-Based Credit System (CBCS) while meeting the unique learning demands of the target audience.

- Flexibility: The CBCS system allows learners to choose elective courses, giving them the flexibility to tailor their education based on their interests and career goals. Distance BCA programmes can offer a variety of electives.
- Practical Orientation: To prepare students for the demands of industry, the curriculum emphasizes on practical projects, hands-on learning, and real-world applications.

- Internal assessments, initiatives, and recurring exams are all ways that the CBCS system involves continual evaluation. Students can periodically assess their progress using this method and focus on areas that require development.

c) How it caters to diverse groups (low income, rural, women, unskilled, etc.)

The distance BCA programmes follows following inclusion and cater to various groups:

- **Accessibility of Study Materials:** Make study materials available in a variety of media, such as printed books, online resources, and audio and video files, so that students with diverse requirements and preferences can use them.
- **Support for Low-Income Learners:** Provides scholarships, financial aid, or choices for installment payments to help low-income students afford their education.
- **Create virtual learning communities** where students may communicate, work together, and ask for assistance from peers and teachers, lessening the isolation that comes with distance learning.

iv. *Appropriateness of Programme to be conducted in Online Learning mode to acquire specific skills and competence*

a) Learning Outcomes, Skills & Competence to be Acquired by Learners:

- **Programming Proficiency:** Learners in the distance BCA programme gains proficiency in various programming languages like Java, C++, Python, etc., enabling them to develop software applications and solve real-world problems.
- **Web Development Skills:** Learners acquire skills in web technologies.
- **Problem-Solving Abilities:** Developing problem-solving skills by applying computational techniques to analyze, design, and implement solutions to complex programming challenges.
- **Software Development Life Cycle (SDLC):** Understanding the various phases of SDLC, including requirements gathering, design, implementation, testing, deployment, and maintenance.
- **Communication Skills:** Developing effective written and verbal communication skills to document and present software projects clearly and professionally.

b) Outcomes Including Knowledge & Understanding of the Subject, Generic Transferable Skills & Competencies:

- **Knowledge & Understanding:** Successful completion of the distance BCA programme equips learners with a strong foundational knowledge of computer science and its applications. This includes understanding programming concepts, algorithms, data structures, computer architecture, software engineering principles, and more.
- **Technical Skills:** Learners acquire specific technical skills related to programming languages, web development, database management, software testing, and debugging.
- **Problem-Solving:** The programme should foster problem-solving skills, allowing learners to approach challenges with a systematic and analytical mindset.
- **Adaptability:** Online learning make the learners to adapt to digital platforms, which builds their adaptability skills—an essential competency in today's fast-changing tech landscape.
- **Lifelong Learning:** Engaging in an online BCA programme instills the value of lifelong learning, encouraging learners to stay updated with emerging technologies throughout their careers.

v. *Instructional Design*

(a) Curriculum Design

Instructional Design for a distance BCA programme is based on the Choice-Based Credit System (CBCS) involves the development of a well-structured curriculum that caters to the diverse needs of learners. An outline of the curriculum design is:

- **Core Courses:** These are the mandatory courses that provide essential foundational knowledge in computer science and applications. Core courses may include subjects like Programming Fundamentals, Data Structures, Database Management Systems, Computer Networks, Software Engineering, and Mathematics for Computer Science.
- **Elective Courses:** The CBCS system allows learners to choose elective courses based on their interests and career aspirations. Offer a variety of elective courses related to specific areas in computer applications, such as Web Development, Mobile App Development, Artificial Intelligence, Cybersecurity, Game Development, etc. This allows learners to specialize in their preferred domains.
- **Foundation Courses:** Include courses that provide a broader perspective on areas like Communication Skills, English Language, Environmental Studies, and Ethics. These courses help learners develop essential soft skills and a sense of social responsibility.
- **Practical and Project-based Learning:** Integrate practical sessions and projects throughout the curriculum to ensure hands-on learning. Assign real-world projects that challenge learners to apply their knowledge and skills to solve practical problems, thus enhancing their problem-solving abilities.

(b) Detailed Syllabus:

For each course learning objectives, topics covered, and assessment methods are formulated.

The syllabus is divided into units or modules for easier navigation and organization.

Guidelines are provided for assignments, projects, and exams to ensure a consistent evaluation process.

(c) Duration of Program:

The Bachelor of Computer Applications programme is to be normally completed in 3 years, each year having 2 semesters. The total credits of the courses for one semester is 20. Hence the BCA program will have 120 credits.

(d) Faculty & Support Staff Requirements:

As per the directives of UGC(ODL & OL) Regulations, 2020, required administrative, academic and technical staff have been appointed.

For the Centre for Online Education the following are functioning:

Director, Deputy Director, Assistant Director

The Academic Faculty present are: Three full-time faculty members have been appointed for online programs

The Administrative staff present are: Deputy Registrar, Assistant Registrar, Section Officer, two Assistants, two Computer operators, two Multi-Tasking Staff

For **Delivery of Online Programmes (LMS & Data Management)**, the following are present:

One Technical Manager and two Technical Assistants

For handling **Admission, Examination and Result**, there is a Technical Manager and two Technical Assistants

The **Technical Team for Development of e-Content** - Self-Learning e-Modules consists of:

One Technical Manager (Production), one Technical Associate (Audio-Video recording & editing) and one Technical Assistant (Audio-Video recording)

Academic Faculty & Student Support from the Department of Computer Applications at the Campus, the following are involved: One Program Coordinator, three Course Coordinators and one Course Mentor.

(e) Identification of Media and Student Support Services:

Print Media: Study materials and study guides are prepared.

Audio/Video: Video lectures, tutorials, and demonstrations to enhance understanding are developed.

Online: Learning management system (LMS) to host course materials, discussions, and assessments were created.

Student Support: Virtual student support services were setup like counseling, mentoring, and peer-to-peer collaboration platforms.

(f) Credit Hours for Each Course or Module:

Credit hours to each course/module based on the depth and complexity of the content are assigned.

Core courses have more credit hours than electives or general education courses.

Typically, one credit hour corresponds to one hour of classroom (or virtual classroom) instruction per week over a semester.

vi. Procedure for admissions, curriculum transaction and evaluation

a. Admission Policy

Admissions to be conducted twice a year (January and July), and as per common regulation for online programs already in place and in effect at the time, in conjunction with UGC Regulations.

b. Minimum Eligibility

- Passed Higher Secondary School Leaving Examination (or equivalent) in any stream with 50% in the aggregate. Those who have pursued Mathematics at Class XII will have an advantage.
- Candidates must appear for a personal interview

c. Fee Structure

As per fee structure in place for online programs and as decided by fee committee from time to time

d. Programme Delivery Methodology

Courses to be delivered under the 'online' mode of learning, with students being provided Self learning material in eBook format, along with access to online Self-Assessment tools. Contact Programs Sessions to be conducted online once per semester.

e. Web Based Tools

A Web based Portal which shall allow the student access to the following

- Admission & Enrolment Details
- Fee Details and Online Fee Payment Gateway
- Prospectus, Regulations & Syllabus
- Notifications (Admissions, fees, examinations etc)
- Course List, with completion status and scores / results
- eBook's of SLM's.
- Self Assessment Tests (unscored)
- Internal Assessments - IA1 & IA2
- Online PCP Lectures (Recorded or via Virtual Classroom session) as conducted each semester.
- Online Learners forum, for student to student interaction.
- Online Copy of the Grade sheet.
- Access to online support in the form of web chat, ticketed email support etc.

f. Evaluation Methodology – Tools & Methods

Internal Assessment marks to comprise the results of Internal Assessment Tests (IA1 & IA2) over the duration of the Semester/Term towards 30% of the final marks, whereas the End Term Assessment , conducted at Examination centres (as per UGC Regulations) to comprise of the remaining 70%.

End Term Exam Regulations to be as per regulations governing on campus programs.

Being an online program, there shall be no provision or requirement to repeat or drop a year within the program.

vii. Requirement of the laboratory support and Library Resources

For a distance education program like BCA, providing laboratory support and library resources is crucial to ensure students receive a comprehensive learning experience. Below are guidelines for each:

(a) Guidelines about Laboratory Support to Perform Practical:

- Virtual Labs: Utilize virtual lab platforms that simulate real-world laboratory environments. These platforms allow students to perform experiments and gain hands-on experience through online simulations.
- Remote Access: Collaboration with local educational institutions or learning centers where students can access physical labs under proper supervision.
- Video Demonstrations: Video demonstrations of practical experiments were developed, explaining the procedure, expected results, and observations.
- Online Support: Dedicated tutors or teaching assistants are assigned who can provide support and clarification through online platforms or email.
- Assessment Criteria: Clear outline of the assessment criteria for practicals and projects, including how they will be evaluated and graded.

(b) Provision of Practical Book:

Detailed Instructions: A comprehensive practical book that includes step-by-step instructions for each experiment or activity is created.

viii. *Cost Estimate of the Programme and the provisions*

A comprehensive cost estimate for a distance BCA programme is considered involving various factors, including infrastructure, faculty, technology, study materials, administrative expenses, and student support services. Here's a general breakdown of the cost estimate and provisions:

- Infrastructure:
Development and maintenance of online learning platforms.
Virtual labs and software licenses.
Data storage and server maintenance.
- Faculty and Staff:
Salaries and benefits for faculty members, instructional designers, and support staff.
Training and professional development for faculty to enhance their online teaching skills.
- Study Materials:
Creation and distribution of digital study materials, e-books, and resources.
Licensing fees for academic content, textbooks, and reference materials.
- Technology:
Computers, laptops, and other hardware for faculty and administrative staff.
Technical support and maintenance of online learning platforms and virtual labs.
Software licenses for various tools and applications used in the curriculum.
- Administrative Expenses:
Office supplies, printing, and stationery.
Marketing and outreach efforts to attract potential learners.
- Student Support Services:
Counseling services for academic and personal support.
Online student help desk and support staff.
Virtual tutoring and peer mentoring programs.

ix. *Quality Assurance mechanism and expected programme outcomes*

(a) Review Mechanism for Enhancing Curriculum & Instructional Design:

- Curriculum is reviewed on a regular basis to make sure it's current, relevant, and in line with business trends and technology improvements.
- Stakeholder Input: To discover areas for development and to comprehend the program's influence on students and the job market, feedback from students, professors, alumni, and business experts are gathered.
- Assessment of Learning Outcomes: Analysis of learning outcomes achieved over time to spot strengths and flaws in curriculum and instructional design.
- Faculty Development: Faculty Members are given continual professional development opportunities so they can improve their methods of instruction, adopt new technologies, and stay up to date on their fields of study.
- External Review: Consultation with outside authorities and accrediting organizations to evaluate the program's efficiency and quality.

(b) Benchmarking Claims:

A clear definition of the learning objectives for each course and the entire BCA programme is made, and these objectives are in line with best practices and industry standards.

- Comparisons to Standards: To ensure compliance and competitiveness, the program's learning objectives and curriculum to relevant national or international standards for BCA programmes are compared.
- It is ensured that the programme is in line with the demands and needs of the IT sector today to increase the employability of graduates.

(c) Programme effectiveness monitoring system for online mode:

- Data analytics is utilized to assess student development, involvement, and performance to enable targeted interventions for at-risk students.
- Feedback from students are taken regularly in order to assess how well they learned, how well online tools worked for them, and how well they were supported.
- Online assessment thorough online tests to measure students' comprehension and application of subjects.
- Faculty Performance is evaluated to assess the efficiency with which professors deliver online courses and assist students.

Bachelor of Computer Application

| Term | Course | Credits |
|------|---|---------|
| 1 | Communication Skills in English | 4 |
| 1 | Data Structure | 4 |
| 1 | Computer Fundamentals and Windows Based Application | 4 |
| 1 | Programming in C | 4 |
| 1 | Programming in C Lab | 2 |
| 1 | Data Structure Lab | 2 |
| 2 | OOPS with C++ | 4 |
| 2 | Mathematics | 4 |
| 2 | DBMS | 4 |
| 2 | Computer Organization and Architecture | 4 |
| 2 | OOPS with C++ Lab | 2 |
| 2 | DBMS Lab | 2 |
| 3 | Operating Systems | 4 |
| 3 | Data Communication | 4 |
| 3 | Computer Graphics | 4 |
| 3 | Client Server Architecture | 4 |
| 3 | Indian Constitution | 4 |
| 4 | Unix and Shell Programming | 4 |
| 4 | Software Engineering | 4 |
| 4 | Java Programming | 4 |
| 4 | Environmental Studies | 4 |
| 4 | Distributed Database Management Systems | 4 |
| 5 | System Programming | 4 |
| 5 | Internet Programming | 4 |
| 5 | Application Programming | 4 |
| 5 | Algorithm and Analysis | 4 |
| 5 | System Programming Lab | 2 |
| 5 | Algorithm and Analysis Lab | 2 |
| 6 | MIS and Enterprise Resource Planning (ERP) | 4 |
| 6 | Computer Network and Security | 4 |
| 6 | Ecommerce | 4 |
| 6 | Project Work | 8 |

Bachelor of Computer Applications

Semester I

Course Name- Communication Skills in English

Course Outline-

| Topics | Unit Reference (from Text) |
|--|-----------------------------------|
| Introduction to business communication; process of communication; components of communication; factors of communication. | Unit 1- Basics of Communication |
| Introduction to non-verbal communication; personal appearance; facial expressions; movements; posture; gestures; eye contact; vocal communication techniques. | Unit 2- Non-verbal Communication |
| Introduction to barriers to communication; physical barriers; psychological barriers; semantic barriers; organisational barriers; interpersonal barriers. | Unit 3- Barriers to Communication |
| Purpose of listening; cognitive process of listening; barriers to listening; overcoming listening barriers; guidelines for improving listening skills. | Unit 4- Listening Skills |
| Introduction to note-making; writing an effective note. | Unit 5- Note-making |
| Types of visuals; use of audio-visuals; principles to use audio-visuals; use and applications of audio- visual equipment. | Unit 6- Audio-visual Aids |
| Preparing for the presentation; structure of the presentation; plan the presentation; mastering the techniques of delivery; impromptu speaking; rehearsing the presentation; guidelines for final speech; handling question-and-answers session. | Unit 7- Oral Communication |
| Various spelling rules; silent consonant letters in some words; variant spellings. | Unit 8- Spelling Rules |
| Compound words with hyphens; use of hyphens with numbers; use of hyphen with prefixes; compound words without hyphens; prefixes without hyphens. | Unit 9- Hyphenation |
| Numbers spelled out; numbers expressed in figures; | Unit 10- Transcribing Numbers |

| | |
|--|-------------------|
| large numbers. | |
| Introduction to effective writing; diction; effective sentences; effective paragraphs. | Effective Writing |

Course Text- Professional Communication by Aruna Koneru.

Course Name- Computer Fundamentals and Windows Based Application
(4 credits - 60 hours)

Objective

This course aims to introduce students to the fundamental concepts of computers such as the evolution of computers, computer organization and architecture, Boolean algebra, digital circuits, operating systems, and computer software. By exploring these topics, students will develop a solid knowledge base and be equipped to analyse and apply these fundamental concepts in various aspects of information technology.

COURSE / LEARNING OUTCOMES

At the end of this course, students will be able to:

1. Recall the key concepts of computer evolution, including the generations of computers and their classification. Also list a computer system's fundamental characteristics and components, storage devices, and input/output devices. *(Remembering)*
2. Understand the concept of computer organization and architecture, including internal communications, machine cycles, and the role of the bus. Also, outline the different types of memory and storage classes. *(Understanding)*
3. Apply proficiency in using different computer software, including system management programs, standard application programs, and Microsoft software (MS-DOS, MS Word, MS Excel, MS PowerPoint, MS Access, and MS Publisher), to solve problems, organize logic, and accomplish specific objectives. *(Applying)*
4. Analyze and compare different computer codes, including the decimal, binary, hexadecimal, and octal systems, to evaluate their characteristics, advantages, and limitations. Additionally, analyze and evaluate the application of Boolean algebra principles, logic gates, and laws in the design and optimization of digital circuits, while simplifying Boolean expressions. *(Analysing)*
5. Evaluate the functions and features of different operating systems, analyze their roles in process management, memory management, file management, and security management, and compare popular operating systems. *(Evaluating)*

Course Outline-

| Topics | Module Reference <i>(from Course Text)</i> |
|--|---|
| Module I: Introduction to Computers (4 Hours) Introduction, Evolution of Computers, Generation of Computers, Classification of Computers, Computing Concepts, The Computer System, Applications of Computers. | Unit 1: Understanding the Computer |
| Module II: Computer Architecture (14 Hours) Introduction, Central Processing Unit, Internal Communications, Machine Cycle, The Bus, Instruction Set. Memory Representation, Random Access Memory, Read Only Memory, Storage Systems, Classification of Storage | Unit 2: Computer Organization and Architecture Unit 3: Memory and Storage Systems Unit 4: Input Devices Unit 5: Output Devices |

| | |
|---|---|
| Systems, Solid-state Storage Devices, Storage Evolution Criteria, Input Devices: Keyboard, Pointing Devices, Scanning Devices, Optical Recognition Devices, Digital Camera, Voice Recognition System, Data Acquisition Sensors, Media Input Devices, Output Devices: Display Monitors, Printers, Classification of Printers, Plotters, Voice Output Systems, Projectors, Terminals. | |
| <p>Module III: Number Systems, Arithmetic, and Logic Gates (24 Hours)</p> <p>Introduction, Decimal System, Binary System, Hexadecimal System, Octal System, 4-bit Binary Coded Decimal (BCD) Systems, 8-bit BCD Systems, 16-bit Unicode, Conversion of Numbers, Different Types of Binary Arithmetic Operations, Signed/Unsigned Numbers, Complements of Binary Numbers, Binary Subtraction Using Complements, Representing Numbers, Integer Arithmetic, Floating-Point Arithmetic, Errors in Arithmetic, Laws of Arithmetic, Elements of Boolean Algebra, Basic Postulates of Boolean Algebra, Boolean Operations, Principle of Duality, Basic Laws of Boolean Algebra, Demorgan's Theorem, Boolean Expressions, Venn Diagram, Basic Logic Gates, Derived Logic Gates, Conversation of Boolean Functions, Adder Circuits, Flip-Flop Circuits, Application of Flip-Flops</p> | <p>Unit 6: Computer Codes</p> <p>Unit 7: Computer Arithmetic</p> <p>Unit 8: Boolean Algebra of Switching Circuits</p> <p>Unit 9: Logic Gates and Digital Circuits</p> |
| <p>Module IV: Operating System and Computer Software (12 Hours)</p> <p>Introduction, Types of Computer Software, System Management Programs, System Development Programs, Standard Applications Programs, Unique Application Programs, Problem Solving, Structuring the Logic, Using the Computer, History of Operating Systems, Functions of Operating Systems, Process Management, Memory Management, File Management, Device Management, Security Management, Types of Operating Systems, Providing User Interface, Popular Operating Systems</p> | <p>Unit10: Computer Software</p> <p>Unit 11: Operating Systems</p> |
| <p>Module V: Microsoft Software (6 Hours)</p> <p>Introduction, MS-DOS, MS Word System, MS Excel System, MS PowerPoint System, MS Access System, MS Publisher</p> | <p>Unit 12: Microsoft Software</p> |

Suggested Readings

Course Text: Self-Learning Material (*SLM) provided by University.

Books:

1. Rajaraman, V. Fundamentals of Computers, PHI Publications
2. Sinha P.K. Fundamental of Computers
3. Manuals of Office Software

Mapping of COs to syllabus:

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V |
|-----------------|----------|-----------|------------|-----------|----------|
| CO1 | H | H | | | |
| CO2 | | H | | | |
| CO3 | | | | H | H |
| CO4 | | | H | | |
| CO5 | | | | H | |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | | M | | | | | | M |
| CO2 | H | | | | | | | M |
| CO3 | | M | M | | | | | |
| CO4 | H | | | | | | M | H |
| CO5 | H | | | | | | M | |

Course Name- Data Structure

(4 credits - 60 hours)

Objective

The objective of the course is to learn how to create data structures to represent a collection of similar data and solve problems using C language. After completion of this course, a student will be able to

- Understand and use the process of abstraction using a C programming language
- Implement various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs
- Understand various searching and sorting techniques.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Recall the basic C constructs and familiarize with basic C syntax, also define and outline the relationship between data and operations on these data using different data structures like arrays, linked list, stacks and queues, graphs and trees. (Remembering)
- CO2: Define C constructs for explaining and generalizing these data structures and choosing appropriate algorithms for efficient program design using C syntax. (Understanding)
- CO3: Compute and demonstrate these data structures and algorithms in different real world problem domain (Applying)
- CO4: Compare and analyse the performance of algorithms based on the problem domain. (Analysing)
- CO5: Review the choice of data structure and algorithms based on problem domain, also would be able to judge and assess the algorithm efficiency based on space and time complexity which forms the fundamental step in the design of an efficient program. (Evaluating).

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|--|---|
| Module I: Basic Concept of Data, Data Structures, Linear Data Structures, Non-Linear Data Structures Operations on Data , Structures Problem Analysis, Algorithm Complexity, Big O Notation | Chapter 2- Data Structures and Algorithm: An Introduction |
| Module II : Use of Various Data Structures for Searching and Sorting, Searching, Linear Search, Binary Search, Sorting, Comparison of Different Search Algorithms, Internal Sorting, Comparison of Various Sorting Algorithms | Chapter 3- Searching and Sorting |
| Module III: Memory Representation of Stacks, Applications of Stacks, Recursion, Reversing Strings, Polish Notation, Memory Representation of Queues, Circular Queue, Priority Queue, Multiple Queue Implementation | Chapter 4- Stacks and Queues |
| Module IV: Singly-Linked Lists, Memory Representation, Operations, Circular Linked Lists Traversing, Insertion Deletion, Doubly-Linked | |

| | |
|--|-------------------------|
| Lists Insertion, Deletion, Dynamic Storage Management: Application of a Doubly-Linked List, Generalized Lists, Garbage Collection, Stack using Linked List Queue using Linked List | Chapter 6- Linked Lists |
| Module V: Trees: Definition and Basic Concepts, Binary Trees, Forms of Binary Trees, Binary Tree Representations, Binary Tree Traversals, Binary Search Tree Searching a Node Inserting a Node Deleting a Node, Binary Tree Traversals, Threaded Binary Tree, Threaded Storage Representation, Applications of Trees, Set Representation, B-Trees, B+ Trees | Chapter 7- Trees |

Course Text- Data Structures Using C by A.K. Sharma, Second Edition, Pearson.

Suggested Readings

1. Lipschutz, S., Theory and Problems of Data Structures (International Edition), Schaum's Outline Series, New Delhi: Tata McGraw-Hill.
2. Kanetkar, Y. P., Data Structures Through C Language, New Delhi: BPB Publications.
3. Chattopadhyay, S.; D. G. Dastidar; M. Chattopdhyay, Data Structures Through C Language, New Delhi: BPB Publications.
4. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V |
|-----------------|----------|-----------|------------|-----------|----------|
| CO1 | H | M | | M | |
| CO2 | H | H | | H | |
| CO3 | M | | H | | H |
| CO4 | L | | M | | M |
| CO5 | | | H | H | M |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | M | | | M | M | | |
| CO2 | H | M | | | | H | | |
| CO3 | M | | L | | | | H | |
| CO4 | L | | L | | | | M | H |
| CO5 | | | | | | | | H |

(4 credits - 60 hours)

Objective: The objective of this course is to make the student to understand the basics of C Programming. The course will build a strong logical capability in the students. The course will start with an understanding of the basic concept of approaching towards the programming paradigm with Pseudocode, flowchart etc. All the basic common programming constructs will be covered in this course.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Define and describe various terms and concepts of C programming language. (Remembering)
- CO2: Interpret information based on their understanding of the concepts of C language's syntax, data types, control statements, functions, pointers, arrays, structures and pointers in C. (Understanding)
- CO3: Solve problems using standard algorithms and translate pseudo-codes into C programs and implement them. (Applying)
- CO4: Apply their analytical skills for choosing the right data structure, function, data types and develop logic to solve various instances of problems. (Analyzing)
- CO5: Evaluate various algorithms used for searching, sorting etc. in terms of correctness and computation cost. (Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|---|--|
| Module 1: (16 Hours) Introduction to Algorithms and Programming Languages Introduction to structured programming and problem solving methods: Algorithms, key features of algorithms, flowcharts, pseudocode, generation of programming languages, structured programming languages. Overview of C: Introduction to C, basic structure of a C program, compiling and executing C programs, comments, characteristics of a good program, character set, identifiers, keywords, data types, constants and variables, I/O statements, operators and expressions, precedence and associativity of operators, type conversion and typecasting. | Chapter 8: Designing Efficient Programs Chapter 9: Introduction to C Chapter 10: Decision Control and Looping Statements |

| | |
|---|---|
| <p>Module 2: (16 hours)</p> <p>Decision Control Statements, Loops and Functions</p> <p>Decision Control Statements and Loops: Introduction to decision control statements, conditional branching statements, go to statements, while loop, do-while loop, for loop, nested loops, break and continue statements.</p> <p>C Functions: Need for functions, function declaration and definition, user defined and library functions, passing parameters to function, return statement, scope of variables, storage classes, recursive functions.</p> | <p>Chapter 11: Functions</p> |
| <p>Module III: (15 hours)</p> <p>Arrays, Strings and Pointers</p> <p>Arrays and Strings: One-dimensional arrays, passing array to function, multidimensional arrays and their applications, character arrays and string operations.</p> <p>Pointers: Introduction to pointers, pointer expressions, null pointers, generic pointers, pointers and arrays, dynamic memory allocation.</p> | <p>Chapter 12: Arrays Chapter 13: Strings Chapter 14: Pointers</p> |
| <p>Module IV: (13 hours)</p> <p>Structures, Files and Preprocessor Directives</p> <p>Structures and Unions: Declaration of structures and simple implementation of structures, unions, enumerated data types.</p> <p>Files: Introduction to files, file management – open, close, input/output operations, Command line arguments.</p> <p>Preprocessor Directives: Introduction to preprocessor directives, macros and file inclusion.</p> | <p>Chapter 15: Structure, Union, and Enumerated Data Type Chapter 16: Files Chapter 17: Preprocessor Directives</p> |

Course Text – Reema Thareja, Computer Fundamentals and Programming in C, Oxford University Press, N Delhi

Suggested Readings

1. Thareja, R., Computer Fundamentals and Programming in C, Oxford University Press, New Delhi.

Course Name- Data Structure Lab

(2 credits - 30 hours)

Objective

The objective of the course is to learn how to create data structures to represent a collection of similar data and solve problems using C programming language. After completion of this course, a student will be able to

- Understand and use the process of abstraction using a C programming language
- Implement various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs
- Understand various searching and sorting techniques.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Recall the basic C constructs and familiarize with basic C syntax, also define and outline the relationship between data and operations on these data using different data structures like arrays, linked list, stacks and queues, and trees. (Remembering)
- CO2: Explain C constructs for generalizing these data structures and choosing appropriate algorithms for efficient program design using C syntax. (Understanding)
- CO3: Compute and demonstrate the data structures and algorithms in different real world problem domain (Applying)
- CO4: Compare and analyse the performance of algorithms based on the problem domain. (Analysing)

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|---|---|
| Module I: a) Implement the solutions on given problems by using the following data structures- <ul style="list-style-type: none">● Arrays● Stacks and● Queues | Chapter 2- Data Structures and Algorithm: An Introduction Chapter 3- Searching and Sorting Chapter 4- Stacks and Queues |
| Module II: a) Implement the solutions on given problems by making use of the following data structures- <ul style="list-style-type: none">● Linked Lists and● Trees | Chapter 6- Linked Lists Chapter 7- Trees |

Course Text- Data Structures Using C by A.K. Sharma

Suggested Readings

1. Lipschutz, S., Theory and Problems of Data Structures (International Edition), Schaum's Outline Series, New Delhi: Tata McGraw-Hill.
2. Kanetkar, Y. P., Data Structures Through C Language, New Delhi: BPB Publications.
3. Chattopadhyay, S.; D. G. Dastidar; M. Chattopadhyay, Data Structures Through C Language, New Delhi: BPB Publications.
4. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II |
|-----------------|----------|-----------|
| CO1 | H | M |
| CO2 | H | H |
| CO3 | M | |
| CO4 | L | |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | M | | | M | M | | |
| CO2 | H | H | | | H | H | | |
| CO3 | M | | H | | | | | |
| CO4 | | | M | M | | | M | M |

Course Name- Programming in C Lab

(2 credits - 30 hours)

Objective

The objective of the course is to understand the basics of Programming in C Language. The main objective of this course is to

- Understand the basic structure of C programs.
- Clarify the different concepts of the different constructs used in C Programming.
- Understand the basic concepts all the useful concepts of C Programming

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

CO1: Apply their analytical skills for choosing the right data structure, function, data types and develop logic to write programs in C. (Analyzing)

CO2: Evaluate various algorithms used for searching, sorting etc. through implementation in terms of correctness and computation cost. (Evaluating)

CO3: Combine the various concepts and ideas learnt in C to plan, propose and develop a product. (Creating)

CO4: Evaluate real-life problems using C Programming. (Evaluating)

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|---|--|
| Module I: Introduction to OS: Linux/Unix, Vi editor, file handling, directory structures, creating and editing simple C programs. C programming using variables, assignment and simple arithmetic expressions. if else Switch-case statements Break, continue. <ul style="list-style-type: none">● Loops, Single and multidimensional arrays. | Chapter 8: Designing Efficient Programs Chapter 9: Introduction to C Chapter 10: Decision Control and Looping Statements |
| Module II: Functions and recursion. Pointers, address operator, declaring pointers and operations on pointers File handling in C. | Chapter 11: Functions Chapter 12: Arrays Chapter 13: Strings Chapter 14: Pointers |

Course Text – Reema Thareja, Computer Fundamentals and Programming in C, Oxford University Press, N Delhi

Suggested Readings

6. Thareja, R., Computer Fundamentals and Programming in C, Oxford University Press, New Delhi.
7. Balagurusamy, E., Computer Fundamentals and C Programming, Tata McGraw Hill Publishing Company Limited, New Delhi.
8. Gottfried, Byron S., Programming with C (Schaum's Outlines Series), Tata McGraw Hill Publishing Company Limited, New Delhi.
9. Kanetkar, Y., Let us C, BPB Publication, New Delhi.
10. Kernighan, B.W., and Ritchie, Dennis M., The C Programming Language, Prentice Hall Pvt. Ltd, New Jersey.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II |
|-----------------|----------|-----------|
| CO1 | H | L |
| CO2 | M | H |
| CO3 | L | H |
| CO4 | L | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | H | | | H | H | | |
| CO2 | M | M | | | M | L | | |
| CO3 | H | | M | | | | | |
| CO4 | | | H | H | | | M | H |

Bachelor of Computer Applications

Semester II

Course Name- Mathematics

Course Outline-

| Topics | Reference from Text Book |
|---|---|
| Define Statistics, Variable and Attribute, Primary Data and Secondary Data, Population and Sample, Complete Enumeration and Sample Survey, Statistical Enquiry, Classification, Tabulation, Mechanical Tabulation | Unit 1- Introduction: Scope, Data Collection and Classification |
| Permutation, Fundamentals Rule of Counting, Results on Permutation | Unit 2- Permutation |
| Combination, Result of Combination | Unit 3- Combination |
| Set, Methods of Set Representation and Notation, Types of Sets, Venn Diagram, Set Operations, Union (Set Addition), Intersection (Set Multiplication), Complement, Difference, Set Operations, Laws of Algebra of Sets, Duality, Verification of Laws (Using Venn Diagram), Proof of the Laws of Set Algebra, Number of Elements in a Set | Unit 4- Set Theory |
| Define Logarithm, Laws of Logarithm, Common Logarithm and Natural Logarithm, Antilogarithm | Unit 5- Logarithm |
| Binomial Theorem, | Unit 6- Binomial Theorem |

| | |
|---|--|
| General Theorem of $(a+x)^n$, Middle Term (s) of $(a+x)^n$, | |
|---|--|

| | |
|--|---|
| Equidistant Terms and Coefficients, Greatest Binomial Coefficient (s), Properties of Binomial Coefficient (s) | |
| Simple Interest, Compound Interest, Interest Compounded Continuously, Rate of Interest, Nominal and Effective Rate of Interest, Growth and Depreciation | Unit 7- Compound Interest |
| Immediate Annuity or Ordinary Annuity, Annuity Due, Deferred Annuity, Perpetual Annuity or Perpetuity, Amortisation, Sinking Fund | Unit 8- Annuities |
| Rounding of Numbers, Absolute, Relative and Percentage Errors, Significant Figures, Short Processes of Calculation, Roots and Reciprocals Expressed as Power, A.P. Series and G.P. Series, Sum and Sum of the Squares of Numbers, Inequalities, Concept of 'Function', Polynomial, Sigma Notation, Simple Interpolation | Unit 9- Other Useful Mathematics Devices |

Course Text- Business Mathematics by NG Das and JK Das.

Course Name- Computer Organization and Architecture

(4 credits - 60 hours)

Objective: This course aims to provide the student with the concepts and basic knowledge necessary to understand the organization and architecture of computing systems.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO6: Recognize, and define the basic components and design of a computer, including CPU, memories, and input/output units. (Remembering)
- CO7: Identify and classify the major components of a CPU and discuss, review the operations intrinsic to it. (Understanding)
- CO8: Discover the parameters of performance improvement and use them in predicting the issues in instruction cycle. (Applying)
- CO9: Analyze the mapping techniques of different memory units; instruction sets interpret the different performance trade-offs between them. (Analyzing)
- CO10: Assess critique and evaluate the performance of CPU, Memory and I/O operations. (Evaluate)

Course Outline-

| Topics | Module Reference (from Course Text) |
|--|--|
| Module I: Arithmetic Logic Unit (10 Hours) Addition and Subtraction (Addition and Subtraction with Signed-Magnitude Data, Hardware Implementation, Addition and Subtraction with Signed-2's Complement Data); Booth's Multiplication Algorithm; Division Algorithm; Floating-Point Arithmetic Operations (Addition, Subtraction, Multiplication, Division).. | Chapter 10. Number Systems Chapter 11. Computer Arithmetic Chapter 12. Digital Logic |

| | |
|--|--|
| <p>Module II: Control Unit (12 Hours)</p> <p>a) Major Components of a CPU; General Register Organization; Stack Organization (Register Stack, Memory Stack, Reverse Polish Notation); Subroutine Call and Return; Fetch Routine; Types of Interrupts; Characteristics of Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer(RISC)</p> <p>b) Micro operations, Control Function, Role of Three-State Bus Buffers in Memory Transfers; Arithmetic Microoperations, Logic Microoperations, Shift Microoperations; Microprogrammed Control and Hardwired Control; Control Memory, Control Word, Microinstruction, Microprogram, Mapping of Instructions; Instruction Formats(Three- Address Instructions, Two-Address Instructions and Zero-Address Instructions); Addressing modes.</p> | <p>Chapter 2. Performance Concepts Chapter 3. A Top-Level View of Computer Function and Interconnection Chapter 13. Instruction Sets: Characteristics and Functions</p> <p>Chapter 14. Instruction Sets: Addressing Modes and Formats</p> <p>Chapter 15. Assembly Language and Related Topics</p> <p>Chapter 12. Digital Logic</p> |
| <p>Module III: Parallel Processing and Multiprocessors (14 Hours)</p> <p>a) Parallel Processing: Flynn’s Classification of computers; Pipelining, Data Dependency, Handling of Branch Instructions, Delayed Load, Delayed Branch; Vector Processing, Supercomputers; Array Processors.</p> <p>b) Multiprocessors: Tightly Coupled, Loosely Coupled; Interconnection Structures (Time- Shared Common Bus, Multiport Memory, Crossbar Switch, Multistage Switching Network, Hypercube Interconnection); Interprocessor Arbitration (Serial Arbitration Procedure, Parallel Arbitration Logic, Rotating Daisy-Chain); Interprocessor Communication and Synchronization, Mutual Exclusion with a Semaphore.</p> | <p>Chapter 16. Processor Structure and Function Chapter 17. Reduced Instruction Set Computers Chapter 18. Instruction-Level Parallelism and Superscalar Processors Chapter 19. Control Unit Operation and Microprogrammed Control VI. Parallel Organization Chapter 20. Parallel Processing</p> |
| <p>Module IV: Memory Organization (14 Hours)</p> <p>Hardware Organization for Associative Memory; Mapping methods for Cache Memory (Associative Mapping, Direct Mapping, Set-Associative Mapping), Write Through, Write Back, Cache Initialization, Cache Coherence; Virtual Memory, Memory management hardware.</p> | <p>Chapter 3. A Top-Level View of Computer Function and Interconnection Chapter 4. The Memory Hierarchy: Locality and Performance Chapter 5. Cache Memory Chapter 6. Internal Memory Chapter 7. External Memory</p> |
| <p>Module V: Input-Output Organization (10</p> | <p>Chapter 8. Input/Output</p> |

| | |
|---|---|
| Hours) Input Output Interface, I/O Bus, Memory Bus, Isolated I/O, Memory-Mapped I/O; Asynchronous Data Transfer, Strobe Control, Handshaking; Modes of Transfer- viz. Direct Memory Access, Programmed I/O, and Interrupt-Initiated I/O; Priority Interrupt (Daisy-Chain Priority, Parallel Priority Interrupt, Priority Encoder); Input-Output Processor; Serial Communication(Character-Oriented Protocol and Bit-Oriented Protocol). | Chapter 16. Processor Structure and Function Chapter 17. Reduced Instruction Set Computers Chapter 18. Instruction-Level Parallelism and Superscalar Processors Chapter 19. Control Unit Operation and Microprogrammed Control |
|---|---|

Course Text – William Stallings, Computer Organization and Architecture, Sixth Edition, Prentice Hall of India Pvt. Ltd., New Delhi, 2002

Suggested Readings

11. M. Morris Mano, Computer System Architecture, Third Edition, Prentice Hall of India Pvt. Ltd., New Delhi, 2002.
12. V. Carl Hamacher, Zvonko G. Vranesic and Safwat G. Zaky, Computer Organization, Fourth Edition, McGraw Hill, 1996
13. William Stallings, Computer Organization and Architecture, Sixth Edition, Prentice Hall of India Pvt. Ltd., New Delhi, 2002

Mapping of COs with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 |
|-----------------|----------|----------|----------|----------|
| CO1 | H | | | |
| CO2 | L | | | |
| CO3 | M | H | | |
| CO4 | | | H | L |
| CO5 | | | M | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | | | | | | | |
| CO2 | H | | | | | | | |
| CO3 | L | H | | | | | | |
| CO4 | | | H | | L | L | | H |
| CO5 | | | | | | | H | M |

Course Name- Database Management Systems

(4 credits - 60 hours)

Objective

The objective of the course is to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve information from a DBMS effectively and efficiently. The course offers the fundamental elements of database management systems, relational data model, normalization, and SQL.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO11: Recall the fundamental elements of database management systems. (Remembering)
- CO12: Design relational data model, entity-relationship model, relational algebra, and schema diagram to represent simple database application scenarios. (Creating)
- CO13: Use the various types of SQL queries to create and handle database. (Applying)
- CO14: Examine and apply normalization to improve the database design and query optimization for query processing. (Analyzing/Applying)
- CO15: Compare and contrast the various types of indexing and advanced databases used in recent days and also summarize the underlying storage hierarchy for such databases. (Understanding)

Course Outline-

| Topics | Module Reference (from Course Text) |
|---|--|
| Module 1: (18 Hours) <ul style="list-style-type: none">a. Database System – Concepts and architecture, Data Abstraction, Data Independence, Database usersb. Data models - Types of data models – Hierarchical, Network, Relational, Object Oriented, and Object-Relational Data Models; Entity Relationship (ER) diagram, Mapping ER Model to Relational Model | Chapter 1 – Database and Database Users Chapter 2 - Database System Concepts and Architecture Chapter 3 – Data Modeling using the ER Model |
| Module 2: (20 Hours) <ul style="list-style-type: none">a. Relational Model - Relational Algebra (RA) , Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC)b. SQL - SQL queries for data manipulation, data definition, data retrieval, transaction control, and data control | Chapter 6 – The Relational algebra and Relational calculus Chapter 9 – Introduction to SQL Programming Techniques |
| Module 3: (12 Hours) <ul style="list-style-type: none">a. Normalization – Basics of Normalizations, Functional dependencies, Normal Forms (1NF to 5NF), Denormalizationb. Query Processing and Optimization – Basic steps in query processing, Evaluation of Relational Algebra Expressions, Join strategies, Heuristic Query Optimization | Chapter 10 – Functional dependencies and Normalization for Relational databases Chapter 15 – Algorithms for Query Processing and Optimization |
| Module 4: (10 Hours) | Chapter 17 – Introduction to Transaction |

| | |
|--|--|
| <p>a. Transactions and Concurrency Control: Transaction processing, States of a Transaction, Concurrency control, Locking, Timestamp-based ordering</p> <p>b. Advanced topics: Distributed Databases, Client-Server architecture, Data Warehouse and Data Mining</p> | <p>processing</p> <p>Chapter 18 – Concurrency control techniques</p> <p>Chapter 25 – Distributed Databases and Client-server architecture</p> <p>Chapter 28 – Data mining concepts</p> <p>Chapter 29 – Overview of Data Warehousing and OLAP</p> |
|--|--|

Course Text - Ramez Elmasri and Shamkant B Navathe, Fundamentals of Database Systems, 5th Edition, Pearson Education

Suggested Readings

14. Ramez Elmasri and Shamkant B Navathe, Fundamentals of Database Systems, 5th Edition, Pearson Education
15. Abraham Silberschatz, Henry F Korth and S Sudarshan, Database System Concepts, 5th Edition, Mc-Graw Hill
16. C.J. Date, Introduction to Database Systems, 8th ed., Pearson Education
17. Bipin Desai, An introduction to Database System, Galgotia Publication
18. Course Text: Self Learning Material (*SLM) provided by the University

Mapping of COs with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 |
|-----------------|----------|----------|----------|----------|
| CO1 | H | M | | L |
| CO2 | H | L | | |
| CO3 | | H | | |
| CO4 | | H | | |
| CO5 | | | H | |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | L | | | | | | | |
| CO2 | H | | | | | | | |
| CO3 | | | L | | | | L | M |
| CO4 | M | | | | | | | |
| CO5 | | | | | | | H | L |

Course Name- OOPS with C++
(4 credits - 60 hours)

Objective

The Main aim of this paper is to give the students a broad understanding of the object oriented approach to problem solving through C++. It provides a practical, productive way to develop software for most applications. It also includes an introduction to object- oriented design, which can promote a

better understanding of the requirements, cleaner designs, and more maintainable systems.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

1. Describe the various features of Object Oriented programming by utilizing the C++ language construct. (Remembering)
2. Explain the standard library, scope and lifetime of a variable and various control statements used in C++ program. (Understanding)
3. Interpret the concept of classes and object in C++ and apply exception handling to solve various exceptions (Analysing, Applying)
4. Evaluate the different type of inheritance and polymorphism and Analyse it in resolving various problems (Analysing, Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|--|--|
| <p>Module I Introduction To Object-Orientation Concepts And OOP (16 Hours)</p> <p>a. Introduction to Object-Oriented Programming: Basic concepts of OOP (Abstraction, Encapsulation, Inheritance, Polymorphism), comparison of procedural programming and OOP; code reusability, creating new data types. C++ Language basics, cin and cout, << and >> operators, setw and endl, Control statements, differences between C and C++.</p> <p>b. Classes and Objects: C++ extension to structures, member access operators static members, arrays of objects, returning objects from functions, Friend functions, Pointers to members, Friend classes, stack class.</p> <p>c. Constructors: Default constructors, overloaded constructors, constructors with default arguments default constructor, copy constructor, dynamic constructor, destructors.</p> | <p>Chapter 1 Chapter 2 Chapter 3 Chapter 4</p> |
| <p>Module II Templates and Exception Handling (12 Hours)</p> <p>a. Templates: string template, instantiation, template parameters, type-checking, function templates, template argument deduction, specifying template arguments, function template overloading, default template arguments, specialisation, conversions.</p> <p>b. Exception handling: Error handling, grouping of exceptions, catching exceptions, catch all, re-throw, resource management, auto ptr, exceptions and new, resource exhaustion,</p> | <p>Chapter 5 Chapter 6</p> |

| | |
|---|---------------------------------|
| <p>exceptions in constructors, exceptions in destructors, uncaught exception, standard exceptions.</p> | |
| <p>Module III Inheritance, Virtual Functions and Polymorphism (16 Hours)</p> <ol style="list-style-type: none"> a. Overloading: Defining operator overloading, operator function as member function and friend function, overloading unary and binary operators, type conversions, function overloading. b. Inheritance: Types of inheritance, Defining derived class, Access specifiers: public, private and protected; public and private inheritance, accessing base class members, ambiguity in multiple inheritance, virtual base classes, abstract classes, Derived class constructor with arguments, Initialization lists in constructors, classes within classes. c. Virtual functions and polymorphism: Virtual functions, pure virtual functions, abstract classes, implementation of virtual functions (virtual pointers and virtual tables in classes with virtual functions), this pointer, static and dynamic binding, virtual functions in derived classes, object slicing, virtual functions and constructors, calling virtual functions from constructors, destructors and virtual destructors, calling virtual functions from destructors, virtual base classes, Rules for virtual functions. d. File handling and streams. e. Basics of file handling in C++, classes for stream operations, operations on files, file opening modes, file pointer, error handling during file operations | <p>Chapter 7 Chapter 8</p> |
| <p>Module IV Object Oriented Design (16 Hours)</p> <p>Overview of object oriented designing (concepts), steps involved in object oriented designing, advantages of OOD, what is modeling, why modeling is required, UML, different views captured by UML diagrams, Use Case diagram(actors, generalization, association, include dependency, extend dependency etc.),organization of use cases, Use Case Packaging, constraints in use case models, how to find out actors, use cases and use case relationships, Class diagrams, representations, association and links, aggregation, composition, dependency,</p> | <p>Chapter 9 Chapter 10</p> |

| | |
|--|--|
| constraints, interaction diagrams(sequence diagrams and collaboration diagrams), representation, boundary objects, controller objects, entity objects, Booch's object identification method, CRC cards, equivalence of sequence diagram and collaboration diagrams, activity diagrams, representation(action states, action flow, object flow, initial state , final state etc..), swim lanes, branching, fork, join etc, OOD goodness criteria. | |
|--|--|

Course Text- E. Balagurusamy, Object-Oriented Programming with C++, Tata McGraw-Hill Publishing Company Limited, New Delhi.

Suggested Readings

1. Nabajyoti Barkakati, Object-Oriented programming in C++, Prentice hall of India private Limited, New Delhi, 2005.
2. James Rumbaugh, Micheal Blaha, William Premerlani, Frederick Eddy, William Loorenson, Object-Oriented Modeling and Design, Prentice hall of India private Limited, New Delhi, 2005.
3. Bjarne Stroustrup, The C++ Programming Language, Special edition, Pearson Education Publication.
4. David Parsons, Object-Oriented Programming with C++, BPB Publications, B-14 Cannought Place, New Delhi.
5. Grady Booch, Object-Oriented Analysis and Design with Applications, Second Edition, Addison-Wesley Publishing Company.
6. Steve Qualline, Practical C++ Programming, Second Edition, Shroff publishers and Distributors Private limited

Mapping of COs to Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V |
|-----------------|----------|-----------|------------|-----------|----------|
| CO 1 | H | H | | | M |
| CO 2 | H | | | | |
| CO 3 | | H | | | |
| CO 4 | | | H | | |

Mapping of COs with POs

| Course Outcomes | PO | PO | PO | PO | PO | PO | PO | PO |
|-----------------|----|----|----|----|----|----|----|----|
| CO1 | | | | | | | | |
| CO2 | M | | | | | | | H |
| CO3 | | | | | | L | | M |
| CO4 | H | | | | L | | | |

Course Name- Database Management Systems Lab

(2 credits - 30 hours)

Objective: The main objective of this lab course is to understand the practical applicability of database management system concepts. Given a schema diagram, students should be able to use a DBMS package to practically create the database, and handle any relational database queries. The lab course also provides practical knowledge of PL/SQL programming for database handling using ORACLE.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO16: Build a database using any DBMS package for a given relational schema. (Creating)
- CO17: Implement SQL queries for various database handling such as DDL, DML, TCL, and Data Retrieval related transactions. (Apply)
- CO18: Predict and associate each multi-table SQL command with a database query to solve any complex database-oriented tasks. (Understand)
- CO19: Write PL/SQL programs for efficient relational database handling using the ORACLE software. (Creating)

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|--|--|
| <p>Module 1: (18 Hours)</p> <ul style="list-style-type: none"> c. Basic SQL Commands: Use SQL Queries to apply the Data Definition Language (DDL) Commands – Create, Alter, Drop, and Rename; Data Manipulation Language (DML) Commands – Insert, Update, and Delete; and Transaction Control Language (TCL) Commands - Commit and Rollback; SQL Retrieval Command SELECT Using From, Where, Group By, Having, and Order By Clauses d. Advanced SQL Commands: Multi-Table Queries Using Joins – Inner Join, Left Join, Right Join, and Full Join; Use of Subqueries; Use of EXISTS and NOT EXISTS Operators | <ul style="list-style-type: none"> a. Interactive SQL Part - 1 b. Interactive SQL Part - 2 c. Interactive SQL Part - 3 d. Interactive SQL Part – 4 |
| <p>Module 2: (12 Hours)</p> <ul style="list-style-type: none"> c. Basics of PL/SQL: PL/SQL block structure, character set, identifiers, literals, delimiters, comments, data types in PL/SQL d. PL/SQL Program Structure: Conditional constructs, iterative constructs, exception handling e. SQL in PL/SQL – Using DDL, DML and TCL Commands in PL/SQL, Data Retrieval, Cursors (Explicit and Implicit) | <ul style="list-style-type: none"> a. Introduction To PL/SQL b. PL/SQL Transactions |

Oracle, 4th Edition, BPB Publications

Suggested Readings

19. Ivan Bayross, SQL, PL/SQL – The Programming Language of Oracle, 4th Edition, BPB Publications
20. Ramez Elmasri and Shamkant B Navathe, Fundamentals of Database Systems, 5th Edition, Pearson Education
21. Abraham Silberschatz, Henry F Korth and S Sudarshan, Database System Concepts, 5th Edition, Mc-Graw Hill
22. Course Text: Self Learning Material (*SLM) provided by the University

Mapping of COs with Syllabus:

| Course Outcomes | Module 1 | Module 2 |
|-----------------|----------|----------|
| CO1 | H | |
| CO2 | H | |
| CO3 | H | L |
| CO4 | L | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | L | | | M | L | | M | H |
| CO2 | M | | | | | | L | |
| CO3 | M | | | | | | | |
| CO4 | H | | | L | | | L | |

Course Name- OOPS with C++ Lab

(2 credits - 30 hours)

COURSE / LEARNING OUTCOMES

At the end of the Lab experiments students will be able to:

1. Identify classes, objects, members of a class and relationships among them needed for a specific problem.
(Remembering/Evaluating)
2. Write C++ programs using OOP principles and proper Program structuring. (Applying/ Understanding)
3. Demonstrate the concepts of polymorphism and inheritance. (Applying)
4. Write C++ programs to implement error handling techniques using exception handling.(Applying)
5. Analyse the real world problems and solve using C++ programming.(Analysing/ Applying)

Course Outline-

| Topics/Experiments | Module Reference (from Course Text) |
|--------------------|-------------------------------------|
|--------------------|-------------------------------------|

| | |
|---|--|
| <p>M1 and M2 Programs on</p> <ol style="list-style-type: none"> 1. Concept of classes and objects, constructors and destructors 2. Use of memory management. 3. Inheritance 4. Virtual functions 5. Using polymorphism – i) operator overloading ii) dynamic binding 6. Use of operator overloading. 7. Exception handling and use of templates. | <p>Referrer at the end of each chapter programming problem section</p> |
|---|--|

Course Text- E. Balagurusamy, Object-Oriented Programming with C++, Tata McGraw-Hill Publishing Company Limited, New Delhi.

Suggested Readings

1. Nabajyoti Barkakati, Object-Oriented programming in C++, Prentice hall of India private Limited, New Delhi, 2005.
2. James Rumbaugh, Micheal Blaha, William Premerlani, Frederick Eddy, William Loorenson, Object-Oriented Modeling and Design, Prentice hall of India private Limited, New Delhi, 2005.
3. Bjarne Stroustrup, The C++ Programming Language, Special edition, Pearson Education Publication.
4. David Parsons, Object-Oriented Programming with C++, BPB Publications, B-14 Cannaght Place, New Delhi.
5. Grady Booch, Object-Oriented Analysis and Design with Applications, Second Edition, Addison-Wesley Publishing Company.
6. Steve Qualline, Practical C++ Programming, Second Edition, Shroff publishers and Distributors Private limited

E-resource for learning

- OOPD, www.spoken-tutorial.org

Mapping of COs with Syllabus:

| Course Outcomes | 1 | 2 |
|-----------------|---|---|
| CO1 | H | H |
| CO2 | M | M |
| CO3 | L | |
| CO4 | | L |
| CO5 | | |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | L | | | M | L | | M | H |
| CO2 | M | | | | | | L | |

| | | | | | | | | |
|-----|---|---|--|---|--|---|---|--|
| CO3 | M | | | | | | | |
| CO4 | H | | | L | | | L | |
| CO5 | | M | | | | L | | |

Bachelor of Computer Application

Syllabus- Semester III

Course Name- Client Server Architecture
(4 credits - 60 hours)

Objective

The main objective of this course is to make the student understand the client server architecture. After the completion of the course students will be able to understand the key terminologies, devices, methodologies and the technologies that are used while deploying services using client server communication.

COURSE/LEARNINGOUTCOMES

At the end of this course students will be able to:

- CO1. Recall the conceptual elements in Client Server architecture. (Remembering)
- CO2. Illustrate and differentiate the various services provided by the client server system. (Understanding)
- CO3. Demonstrate with various methodologies used in client server architecture. (Applying)
- CO4. Analyse and design for hosting applications in client server system. (Analysing)
- CO5. Summarize and validate a practical solution towards the deployment of applications. (Evaluating)

Course Outline-

| | |
|--|---|
| <p>Module I (15 hours)</p> <p>Client–Server Computing, Peer-to-Peer Network, Client–Server Network, Client–Server Model, Butler Pyramid Model of Client–Server Computing, Client–Server Layers, Types of Client–Server Architecture, Two-Tier Architecture, Three-Tier Architecture, Multi or N-Tier Architecture, Client–Server-based Services, Heterogeneous Computing, Cross- Platform Computing, Cross-Platform Software, Distributed Computing, Distributed Systems Management, Complexity Management</p> <p>Scope of Distributed Systems Management, Systems Management Mechanisms, Distributed Systems Management— Requirement and Framework, Software Distribution Requirements, Golden Rules of Client–Server Implementation, Costs of Client–Server Computing, Strategies for Cost Reduction</p> | <p>Chapter 1- client–server computing</p> |
| <p>Module II (15 hours)</p> <p>Fundamentals of Client–Server Design, Client–Server System Interaction, Load Balancing Between Client and Server, Interaction, Types of Communication, Synchronous Communication, Asynchronous Communication, Communication Techniques, Remote Procedure Call (RPC), Native Communication Protocol, Messaging, Object Orientation, Communication Protocols, Client–Server Interaction Protocols, Techniques for implementing Client–Server Applications, Optimizing Applications for Client–Server Communication, Client–Server Application System Implementation, Request Acceptance and Dispatching, Execution of Request, Client–Server Interaction using Messages, Client–Server Interaction Protocol, Request and Response Communication Channel, Client–Server Interaction using DHCP Servers</p> | <p>Chapter 2- Design</p> |
| <p>Module III (10 hours)</p> <p>Multiprogramming, Multitasking, Single Tasking and Multitasking, Cooperative Multitasking/Time-Sharing, Preemptive Multitasking/Time-Sharing, Memory Protection, Memory Swapping, Programming in a Multitasking Environment, Process Management in Multitasking Operating Systems, Multiprocessing, Processor Symmetry, Processor Coupling, Instruction and Data Streams, Advantages and Disadvantages</p> | <p>Chapter 3- Multitasking</p> |

| | |
|--|---|
| <p>of Multiple Processors, Child and Parent Processes, Developing Server Applications using Processes, Threads, Multithreading, Multithreading Models, Thread Cancellation, Thread Pools, Benefits of Multithreading, Server Communication Model</p> | |
| <p>Module IV (10 hours)</p> <p>What is a Process?, Process States, Process Control Block, Cooperating Processes and Independent Processes, Implementation of Scheduling, Schedulers, Processing Queues, Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Avoidance by using Banker's Algorithm, Detection and Recovery, Context Switching, Buffering, CPU Scheduler, Dispatcher, Scheduling Criteria, Scheduling Algorithms, Pre-emptive Scheduling, Non-Pre-emptive Scheduling, First Come First Served, Round Robin, Priority-based Scheduling, Pre-emptive or Non-Pre-emptive Priority Scheduling Algorithm, Shortest Job First (SJF), Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Multiple Processor Scheduling, Real-Time Scheduling, Critical Section, Mutual exclusion, Semaphores, Drawbacks of Semaphores</p> | <p>Chapter 4 - Synchronization</p> |
| <p>Module V (10 hours)</p> <p>Network Communication, the OSI Model, Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer, Cryptography, Interfaces, Inter-Process Communication, Child Process, Parent Process, Pipes, Sockets, Types of Sockets, Socket States and Client-Server Model, Creating a Socket, Binding a Socket, Listening to the Socket, Accepting Connections, Connecting to the Daemon, Messaging, Semaphores, Initializing a Semaphore Set, Controlling Semaphores, Semaphore Operations, Building Portable Client-Server Applications, Procedural Designs Using C, Object-Oriented Designs Using C++, Object Encapsulation, Objects and their Classes, Messages and Methods, Overriding, Architecting Platform-Independent Source Code, Client-Server Application Architecture Using Frameworks</p> | <p>Chapter 5- Network Communication</p> |

Course Text- Client Server Computing by Dawna Dewire

Suggested Readings

1. An Introduction to client server computing, S.C.Yadav, S.K. Singh, New Age International

Publishers

2. Enterprise Cloud Computing by Gautam Shroff, Cambridge
3. Information and Communication Technology for Development (ICT4D), Richard Heeks, 2017, Taylor and Francis.
4. Information and Communications Technologies for Development: A Comparative Analysis of Impacts and Costs, Balaji, P. and Keniston, K

Mapping of Cos with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 |
|------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| CO1 | H | H | M | M | M |
| CO2 | H | M | M | | |
| CO3 | M | M | H | H | H |
| CO4 | H | H | | | |
| CO5 | H | M | M | | |

Mapping of Cos with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|------------------------|------------|------------|------------|------------|------------|------------|------------|------------|
| CO1 | M | | | | M | H | M | M |
| CO2 | L | | M | | | M | H | H |
| CO3 | H | M | M | | | M | M | M |
| CO4 | H | H | H | M | M | M | M | M |
| CO5 | H | | | | H | H | | H |

Course Name- Computer Graphics

Course Outline-

| Topics | Reference from Text Book |
|--|--|
| Display Devices, Pixel, Resolution, Image Resolution vs Dot Pitch, Aspect Ratio, Cathode Ray Tube (CRT), Raster Scan Display, Random Scan Display, Direct View Storage Tube (D VST), Flat Panel, LCD, Raster Scan Systems, Random Scan Systems- Input Devices, Keyboard, Mouse, Trackball, Joystick, Digitizer and Graphics Tablet, Touch Panel, Light Pen, Data Glove, Voice Recognition System, Hard Copy Devices, Printer, Impact vs Non-impact Technologies, Dot Matrix Printer, Line Printer, Ink et Printer, Laser Printer, Electrostatic Printer , Plotter, Graphics Software | Unit 2- Overview of Graphics Systems |
| Line Drawing Algorithms, Simple Digital Differential Analyzer (DDA), DDA Algorithm, Bresenham's Circle Generation, Bresenham Algorithm | Unit 3- Output Primitives Unit 4- Attributes of Output Primitives |

| | |
|--|--|
| <p>(Pseudocode) Midpoint Circle Algorithm, Character Generation, Line Attributes, Area Fill Attributes, Character Attributes, Bundled Attributes, Inquiry Functions</p> | |
| <p>Translation, Rotation, Rotation about Origin, Rotation about an Arbitrary Pivot Point, Scaling, Scaling with Respect to the Origin, Scaling with Respect to any Arbitrary Point, Reflection, Reflection about X axis, Reflection about Y axis, Reflection about the Straight Line $y = x$, Reflection about the Straight Line $y = -x$, Reflection Relative to the Origin, Shear- Matrix Representation, Homogeneous Coordinates, Composite Transformations, Inverse Transformation, General Pivot Point Rotation, General Fixed Point Scaling, Reflection through an Arbitrary Line, Raster Methods for Transformations, 2D Viewing, Viewing Pipeline, Viewing Transformations, Concepts of Window and Viewport, Window to Viewport Transformation, Clipping Operations, Point Clipping, Line Clipping, Cohen-Sutherland Line Clipping, Sutherland-Hodgman Algorithm for Polygon Clipping, Curve Clipping, Text Clipping</p> | <p>Unit 5- Two-Dimensional Geometric Transformations Unit 6- Two-Dimensional Viewing</p> |
| <p>Editing Structure, Graphical User Interface, Elements of GUI, Guidelines for Designing an</p> | <p>Unit 7- Structures and Hierarchical Modeling Unit 8- Graphical User Interfaces and</p> |

| | |
|--|---|
| <p>Effective GUI, Interactive Input Methods, Interactions with a Mouse, Mouse Programming (In C), The User Dialogue, Input of Graphical Data, Input Functions, Interactive Picture Construction Techniques, Interactives Devices, Techniques of Picture Construction</p> | <p>Interactive Input Methods</p> |
| <p>Three-Dimensional Graphics Packages', Polygon Surfaces, Polygon Tables, Polygon Meshes, Sweep, Representation, Constructive Solid Geometry (CSG) Methods, Octrees, Quad Trees, Types of Quad Trees, Common Applications of Quad Trees</p> | <p>Unit 9- Three-Dimensional Concepts Unit 10- Three-Dimensional Object Representations</p> |

Course Text- Computer Graphics by Donald D. Hearn and M. Pauline Baker.

Course Name- Data Communication

(4 credits - 60 hours)

Objective

The main objective of this course is to make the students understand the characteristics of signals propagated through different transmission media, including concepts of attenuation and noise, error-detection, and error-correction techniques and interfacing and synchronization issues.

COURSE/LEARNINGOUTCOMES

At the end of this course students will be able to:

- CO1 : Define the fundamentals of data communication and various techniques of communications. They will also be able to recall the layered structure of computer networks.(Remembering)
- CO2: Explain about different network topology and the type of protocol required for different communication techniques. (Understanding)
- CO3: Understand the requirements of various networking devices and make use of the network accordingly.(Applying)
- CO4: Compare different networking devices. They will also be able to analyse different network behavior depending on performance parameters. (Analysing)
- CO5: Establish and determine a computer network either Wired or Wireless (Applying, Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|---------------|--|
|---------------|--|

| | |
|--|--|
| <p>Module I (14hours)</p> <ul style="list-style-type: none"> a. Network Classification, LAN, MAN, WAN, Wireless Networks, Internet, Narrow Band and Broadband ISDN, b. Network Models, OSI Model, Layers and their Functions in OSI Model, TCP/IP Protocol Suite, c. Signal and Data, Channel, Baud Rate or Data Rate, Bandwidth, Bit Rate, Simplex and Duplex Communication, Protocols and Standards, Transmission Impairments, d. Attenuation –Distortion, Channel Noise, Throughput Asynchronous Transfer Modes, Communication Concepts | <p>Chapter 1- Data Communications Chapter 2- Network Models Chapter 3- Data and Signals Chapter 4- Digital Transmission</p> |
| <p>Module II (12hours)</p> <p>M</p> <ul style="list-style-type: none"> a. Physical Layer and Media, FDM, TDM and Spread Spectrum, b. Guided Media, Unguided Media c. Circuit Switched Networks, Virtual Circuit Networks, Datagram Networks d. Data Transfer Networks, Dial-Up Modems, Digital Subscriber Line | <p>Chapter 6- Multiplexing Chapter 7- Transmission Media Chapter 8- Switching Chapter 9- Using Telephone and Cable Networks for Data Transmission</p> |
| <p>Module III (15 hours)</p> <ul style="list-style-type: none"> a. Error Detection and Correction, Block Codes, Cyclic Codes, Point-to-Point protocol (PPP), Frame Relay, Connecting Devices: Hubs, Switches, Repeaters, Bridges, Routers, Gateway, Satellite Networks, Cellular Telephony b. Framing, Flow and Error Control, Sliding Window Protocol, c. ALOHA, CSMA, CSMA/CD, Channelization Methods, d. LAN Topologies and Protocols, CSMA/CD and IEEE 802.2, Wired LAN, Media Access Control, LAN Hardware, LAN Operating Systems, Transmission Media - Bounded | <p>Chapter 10- Error Detection and Correction Chapter 11- Data Link Control Chapter 12- Multiple Access Chapter 13- Wired LANs: Ethernet Unit 14- Wireless LAN's</p> |

| | |
|--|--|
| <p>Media, Ethernet Standards, IEEE Standards, IEEE 802 Standards for LANs and MANs, IEEE 802.2 LLC-Logical Link Control Layer, IEEE 802.3 Ethernet Technologies, IEEE 802.4 Token Bus, IEEE 802.5 Token Ring, IEEE 802.6 Distributed Queue Dual Bus, Wireless LAN</p> <p>e. Connecting Devices: Hubs, Switches, Repeaters, Bridges, Routers, Gateway</p> <p>f. Satellite Networks, Cellular Telephony, Frame Relay</p> | <p>Chapter 15- Connecting LANs, Backbone Networks, and Virtual LANs</p> <p>Chapter 16- Wireless LAN's: Cellular Telephone and Satellite Networks Unit 18- Virtual-Circuit Networks: Frame Relay and ATM</p> |
| <p>Module IV (13 hours)</p> <p>Network Layer, Logical Addressing, Multicasting and Broadcasting, Internet Protocol: IPv4, IPv6, IPv4 Addressing, Internet Protocol Version 6 (IPv6), Address Mapping, Uni casting, ICMP, IGMP</p> | <p>Chapter 19- Network Layer: Logical Addressing</p> <p>Chapter 20- Network Layer: Internet Protocol</p> <p>Chapter 21- Network Layer: Address Mapping, Error Reporting and Multicasting</p> <p>Chapter 22- Network Layer: Delivery, Forwarding and Routing</p> |
| <p>Module V (6 hours)</p> <p>Transport Layer, UDP, TCP, Congestion Control, Open and Closed Loop, Quality of Service, Application Layer, Domain Name System (DNS) Name Space, Domain Name Systems and Resolution, Remote Logging, Electronic Mail, File Transfer,</p> <p>WWW, HTTP, Multimedia Communication, Voice Over IP</p> | <p>Chapter 23- Process-to-Process Delivery: UDP, TCP, and SCTP</p> <p>Chapter 24- Congestion Control and Quality of Service</p> <p>Chapter 25- Domain Name System</p> <p>Chapter 26- Remote Logging, Electronic Mail and File Transfer</p> <p>Chapter 27- WWW and HTTP Unit Chapter 29- Multimedia</p> |

Course Text- B.A. Forouzan, Data Communications and Networking, McGraw-Hill

Suggested Readings

1. William Stallings, Data and Computer Communications, Sixth Edition, Pearson Education Asia.
2. Prakash C. Gupta, Data Communications and Computer Networks, PHI
3. William L. Scweber, Data Communication, McGraw Hill.
4. Tenenbaum, A. S., Computer Networks (Fourth Edition), New Delhi: Prentice-Hall India
5. Larry L. Peterson and Bruce S. Davie, Computer Networks: A systems approach, 3rd Edition, Morgan Kaufmann Publishers.
6. Mary E.S. Loomis, Data Communications, PHI.

Mapping of Cos with Syllabus:

| CourseOutcomes | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 |
|-----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| CO1 | H | M | M | L | M |
| CO2 | | M | H | H | H |
| CO3 | H | H | M | H | L |
| CO4 | M | H | H | H | L |
| CO5 | | M | | H | L |

Mapping of Cos with POs

| CourseOutcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------------|------------|------------|------------|------------|------------|------------|------------|------------|
| CO1 | M | H | M | L | | H | | H |
| CO2 | M | M | | | H | M | H | H |
| CO3 | H | H | | | H | M | H | H |
| CO4 | | | | | | H | | H |
| CO5 | | | | | M | H | M | M |

Course Name- OPERATING SYSTEMS

(4 credits - 60 hours)

Objective

The main objective of this course is to introduce the students to a layer of software called Operating Systems, whose job is to manage all the devices of a computer system and provide user programs with a simple interface to the hardware. This course will familiarize the students with the concepts of processes, memory management, file management, Input/Output management and the potential problem of deadlocks.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Define the basic concepts of operating systems and terminology related to operating systems such as processes, threads, files, semaphores, IPC abstractions, shared memory regions. (Remembering)
- CO2: Analyze important algorithms such as Process scheduling and memory management algorithms. (Analysing)
- CO3: Categorize the operating system's resource management techniques, dead lock management techniques, memory management techniques. (Analysing)
- CO4: Discuss and examine the importance of File and I/O system management in operating systems. (Evaluating)
- CO5: Description of protection and security in operating system. (Understanding).

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|---|---|
| Module I: Need and Services of Operating system, Classification of operating system, Single user, Multi user, Simple batch processing Multiprogramming, Multitasking, Parallel system, Distributed system, Real time system | Unit 3- Overview of Operating Systems |
| Module I: Process concept, Process scheduling, Threads, CPU scheduling, Basic concept, Scheduling criteria, Scheduling algorithms | Unit 5- Process Management Unit 7- Scheduling |
| Module II: Deadlock characteristics, Deadlock Prevention, Deadlock Avoidance, Detection and Recovery, Critical sections, Synchronization hardware, Semaphores, Combined approach to deadlock handling | Unit 8- Deadlocks |
| Module III: Memory Concept, Logical versus physical Address space, Swapping, Partition, Segmentation | Unit 11- Memory Management |
| Demand paging, Page replacement | Unit 12- Virtual Memory |

Course Name- Indian Constitution

Course Outline-

| Topics | Unit Reference (from Text) |
|--------------------------------------|---|
| Framing of Indian Constitution | Meaning of Term Constitution, Importance, Development of Indian Constitution (<i>Pages 1-38</i>) |
| Philosophy of the Constitution | Salient Features of Indian Constitution (<i>Pages 39-54</i>) |
| Preamble | Amendment of the Preamble, Importance of the Preamble (<i>Pages 55-63</i>) |
| Constitution and Amendments | Balance between Rigidity and Flexibility, Parliamentary Democracy with an Elected Principle, Procedure of Amending the Constitution (<i>Pages 116-144</i>) |
| Human Rights | Meaning and Importance, Universal Declaration of Human Rights, Development of Human Rights and Fundamental Rights, International Law and Position of India, Social and Gender Discrimination, Torture and Genocide, two Human Rights Covenants (<i>Pages 145-160</i>) |
| Specific Fundamental Rights | Fundamental Rights and Duties, Special Privileges for SC/STs, Backward Classes, Woman, Children and religious and Linguistic Minorities (<i>Pages 78-101</i>) |
| Directive Principles of State Policy | Distinction between Fundamental Rights and Directive Principles, Classification of the Directive Principles, Values and Limitations, Reorganization of States and National Integration (<i>Pages 180-193</i>) |
| Union Executive | President, Prime Minister, Council of Minister (<i>Pages 195-250</i>) |
| | Composition of Parliament, Chairman of the House, Lok Sabha, |

| | |
|-------------------|--|
| Union Legislature | Rajya Sabha, Indian Federalism and Political Parties <i>(Pages 251-277)</i> |
| State Government | Legislatures, Governors, Chief Ministers and Council of Minister <i>(Pages 278-298)</i> |

| | |
|--------------------------|--|
| Central- State Relations | Legislative, Administrative and Financial, Adult Franchise and Election Commission <i>(Pages 299-323)</i> |
| Judiciary | Supreme Court and High Court <i>(Pages 346-368)</i> |

Course Text- Indian Constitution by *Manoj Sharma*

Bachelor of Computer Applications Semester IV

Course Name- Software Engineering

(4 credits - 60 hours)

Objective: To provide ability Analyse a scenario and produce a problem statement. The learners will be able to produce a conceptual solution which includes sample prototypes, domain models, and user stories. The learners will be able to describe the attributes and activities involved in software development process models and testing.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Relate and examine the life cycle models of software. (Remembering)
- CO2: Interpret and differentiate various software life cycle models(Understanding) Discover the parameters of performance improvement and use them in predicting the issues in instruction cycle. (Applying)
- CO3: Experiment with different software architectures and identify the best feasible one (Applying)
- CO4: Analyse and design the software requirement specification(Analysing)
- CO5: Evaluate the software project by using maintenance plan. (Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|--|--|
| Module I: Introduction (10 Hours) Problems and solutions: Why software is developed. Problem and vision statements. Goals and objectives. Definitions and paradigms, A generic view of software engineering. Software development life cycle, Role of quality, metrics and measurement. | Chapter 1: Introduction Chapter 2: Software Life cycle model |
| Module II: Requirements Analysis (15 Hours) The feasibility study, Software Requirement Analysis and Specifications, work breakdown structure (WBS), Problem Analysis, Creating software requirement specification document (SRS). | Chapter 3: Software Project Management Chapter 4: Requirement Analysis and Specification |
| Module III: Designing Software applications (15 Hours) Process Models: How software is built. The fundamental design concept for data, architectural and procedural designs. Conceptual solutions. Agile concept and User stories. Domain modeling with UML diagrams-Class diagram, Use cases etc, Object oriented design paradigm; Creation of technical design document. | Chapter 5: Software Design Chapter 6: Function Oriented Software Design Chapter 7: Object Modeling using UML Chapter 8: Object Oriented software Design |

| | |
|---|---|
| Module IV: Software Implementation (10 Hours) The relationship between design and implementation, Implementation, Coding the procedural design, good coding style and review of correctness and readability. | Chapter 9: User Interface Design |
| Module V: Software Testing and Maintenance (10 Hours) | Chapter 10: Software coding and Testing |

Course Text – Rajib Mall, Fundamentals of Software Engineering, PHI.

Suggested Readings

1. Sommerville, “Software Engineering”, Addison Wesley.
2. Roger S.Pressman, “Software Engineering–A Practitioner’s Approach”, McGraw Hill Companies.
3. Rajib Mall, Fundamentals of Software Engineering, PHI.

Mapping of COs with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 |
|-----------------|----------|----------|----------|----------|
| CO1 | H | | | |
| CO2 | H | | | |
| CO3 | H | M | | |
| CO4 | | | M | M |
| CO5 | | | L | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | | | | | | | |
| CO2 | M | | | | | | | |
| CO3 | L | H | | | | | | |
| CO4 | | | H | | M | L | | H |
| CO5 | | | | | | | H | M |

(4 credits - 60 hours)

Objective

The course is designed to impart the knowledge and skill required to solve real world problems using an object oriented approach utilizing Java language constructs. This course covers the two main parts of Java i.e. Java Language and Java Library (JDK 5). After completion of the course, a student is expected to be able to

- Do Object Oriented Programming using Java
- Implement Exception handling and Multithreading in Java.
- Create Java Applets.
- Set up a GUI using Swing components
- Do Network Programming in Java.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

1. Recall the knowledge of the structure and model of the Java programming language, (Remembering)
2. Explain the use of Java programming language for various programming technologies (Understanding)
3. Develop software in the Java programming language. (Applying)
4. Analyse user requirements for software functionality required to decide whether the Java programming language can meet user requirements (Analysing)
5. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (Evaluating)
6. Propose the use of certain technologies by implementing them in the Java programming language to solve the given problem (Creating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|--|--|
| Module I: Java Fundamentals (16 hours) Genesis, Java Philosophy, Java & Internet, Object-Oriented Programming features, Java Applet and Application, Java Environment and Java Development Kit (JDK) and Java Standard Library (JSL), Tokens, Expressions, Using Data Types, Declarations, Control Flow | Chapter 1 Chapter 2 Chapter 3 Chapter 4 |
| Module II: Java Classes, Packages and Interfaces, Java Streams (14 hours) Introduction, Classes, Working with Objects, Packages, Inheritance, Interfaces Data Flow with Java Streams, Input Streams, Output Streams | Chapter 5 Chapter 6 |
| Module III: Exception Handling in Java and Java threads (10 hours) Introduction, Exception Methods, java.lang Exceptions Introduction, Creating Threads, the Life Cycle of Thread, Thread Methods, Using Threads, Synchronization of Threads | Chapter 7 Chapter 8 Chapter 9 |

| | |
|---|--|
| Module IV: Java Applets (10 hours) Introduction, Applet Examples, The java.applet.Applet Class, The Five Stages of an Applet's Life Cycle, Methods for Adding UI Components, Methods for Drawing and Event Handling | Chapter 10 Chapter 11 |
| Module V: Java AWT (10 hours) Introduction, Swing Component and Container classes, Layout managers (Flow Layout, Grid Layout, Border Layout), Handling events, Adapter classes, Anonymous inner classes, Swing GUI components :JLabel, JTextField, JTextArea, JButton, JCheckBox, JRadioButton, JList, JComboBox, JScrollBar, JScrollPane, JToolTip, JPanel, JFrame, introduction to database connectivity with JDBC. | Chapter 12 Chapter 13 Chapter 14 |

Course Text- Deitel, H. M.; P. J. Deitel, Java: How to Program, New Delhi: Prentice Hall India.

Suggested Readings

- Schildt, H., The Complete Reference Java 2, New Delhi: Tata McGrawHill
- Moss, K., Java Servlets, New Delhi Tata McGraw-Hill
- Russell, Java Programming for the absolute Beginner, New Delhi: Prentice-Hall India
- Hanagan D., Java Examples in a Nutshell, New Delhi: O' Reilly

Mapping of COs to Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V |
|-----------------|----------|-----------|------------|-----------|----------|
| CO 1 | H | M | | | |
| CO 2 | H | M | | | |
| CO 3 | | M | M | M | M |
| CO 4 | | M | M | M | M |
| CO 5 | | M | M | M | M |
| CO 6 | | M | M | M | M |

Mapping of COs with POs

| Course Outcomes | PO | PO | PO | PO | PO | PO | PO | PO |
|-----------------|----|----|----|----|----|----|----|----|
| CO1 | | | | | | | | |
| CO2 | M | | | | | | | H |
| CO3 | | | | | | L | | M |
| CO4 | H | | | | L | | | |
| CO5 | M | | L | | | | M | |
| CO6 | | | L | | | M | | H |

Course Name- Unix and Shell Programming
(4 credits - 60 hours)

Objective

The course aims to provide students with a comprehensive understanding of Unix, proficiency in shell scripting, and the ability to automate tasks effectively using Unix utilities and commands.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO1: Introduction to Unix: Explain the basic concepts, features, and history of the Unix operating system. (Understanding)
- CO2: Unix File System: Gain knowledge about the Unix file system structure, file permissions, and file manipulation commands. (Remembering)
- CO3: Unix Shell: Explain the role of processes and the basic commands to control processes. (Understanding)
- CO4: Shell Scripting Basics: Learn and apply the fundamentals of shell scripting, including script execution, variables, data types, and basic control structures (loops, conditions) and also explore advanced shell scripting techniques, such as command substitution, input/output redirection, and process control. (Applying)
- CO5: Advanced Shell Scripting: Explore advanced scripting concepts, including file handling, regular expressions, and pattern matching. (Applying)

Course Outline-

| Topics | Reference from Text Book |
|--|--|
| Module I: History, Define UNIX, UNIX Components, UNIX Commands, Basic Commands, Getting Help, Command Substitution, Multiple Commands, Aliases- Alternate Names to Commands | Chapter 1- Introduction to UNIX |
| Module II: UNIX Files, Categories of Files, Hidden Files- Dot Files, File System- Organizing the Unix Files, Path Names- Reaching a Required File, Home Directory, Directory Commands, Dot (.) and Double Dot (. .) File Names, File Related Commands, Wild Cards-Filename Generation, Displaying the Contents of File, Printing of Files, Comparing Files | Chapter 2- Files and File Organization |

| | |
|--|--|
| <p>Files Ownership, File Attributes, The <i>ls</i> Command, The <i>file</i> Command, The <i>chmod</i> Command, The <i>chown</i> Command, The <i>chgrp</i> Command, Times Associated with File, The unmask Command</p> | <p>Chapter 3- File Attributes and Permissions</p> |
| <p>Module III: Standard I/O, Redirection, Pipe and Pipeline, Mixing Inputs from Standard Input and a File, Filter, The tee Command, Terminal (/dev/tty) and Trash (/dev/null) Files,</p> | <p>Chapter 4- Standard I/O, Redirection Pipes and Filters</p> |
| <p>Sample Database File, Handling Columns and Fields, The <i>sort</i> Command, The <i>uniq</i> command, The <i>tr</i> command</p> | |
| <p>The vi Editor, Editing with vi, Moving the Cursor, Editing, Copying and Moving Text, Pattern Searching, Repeating Last Editor Command, Undoing Commands, Joining Lines, Writing Selected Lines on to Separate File, Using the Shell from vi, Configuring the vi Environment</p> | <p>Chapter 5- The vi Editor</p> |
| <p>Module IV: Regular Expressions, The <i>grep</i> Family, The <i>egrep</i> Command, The <i>fgrep</i> Command, The Stream Editor- sed</p> | <p>Chapter 6- Regular Expressions- grep Family of Commands and the sed</p> |
| <p>Module V: Meaning of Process, Parent and Child Processes, Types of Processes, Foreground and Background Processes, Internal and External Commands,</p> | <p>Chapter 7- The Process</p> |

| | |
|--|--|
| <p>The <i>ps</i> Command, Process Creation, The <i>nohup</i> Command, The <i>nice</i> Command, The <i>time</i> Command, Signals, The <i>trap</i> Command, The <i>stty</i> Command, The <i>kill</i> Command, The <i>wait</i> Command, Job Control, Command History, Scheduling Jobs Execution</p> | |
| <p>Module VI: Shell Variables, The <i>export</i> Command, The <i>.profile</i> File, The First Shell Script, The <i>read</i> Command, Positional Parameters, The \$? Variable, The <i>set</i> Command, The <i>exit</i> Command, Branching Control Structures, Loop-Control Structures, The <i>continue</i> and <i>break</i> Statements, The <i>expr</i> Command, Real Arithmetic in Shell Programs, The <i>here</i> Document (<<), The <i>sleep</i> Command, Debugging Scripts, The <i>script</i> Command, The <i>eval</i> Command, The <i>exec</i> Command</p> | <p>Chapter 8- Shell Programming</p> |
| <p>Communication Process, The <i>news</i> Command, Message of the Day- <i>The motd</i>, The <i>mesg</i> Command, The <i>write</i> Command, The <i>wall</i> Command, The <i>talk</i> Command, The <i>finger</i> Command, Electronic Mail</p> | <p>Chapter 10- Basic Communication Tools</p> |

Course Text- Introduction to UNIX and SHELL Programming by M.G. Venkateshmurthy.

Suggested Readings

1. The Linux Command Line by William E. Shotts Jr.
2. Unix and Linux System Administration Handbook by Evi Nemeth, Garth Snyder, Trent R. Hein, and

Ben Whaley.

3. Classic Shell Scripting by Arnold Robbins and Nelson H.F. Beebe.
4. Unix and Shell Programming" by Richard F. Gilberg and Behrouz A. Forouzan.
5. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V | Module VI |
|-----------------|----------|-----------|------------|-----------|----------|-----------|
| CO1 | H | M | L | | | |
| CO2 | | H | M | | | |
| CO3 | | | | | H | |
| CO4 | | | | | | H |
| CO5 | | | | M | | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | L | | | | | | | |
| CO2 | | | | | | | | |
| CO3 | | | | | | L | | |
| CO4 | | | | | | | | |
| CO5 | | | | | | | | L |

Course Name- Distributed Database Management Systems

(4 credits - 60 hours)

Objective

The objective of the course is to present an introduction to distributed database management systems, with an emphasis on distributed DBMS framework and architecture. The course offers an in-depth study of the classical distributed database management issues such as distribution design, distributed

query processing and optimization, and distributed transaction management.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO20: Recall the fundamental concepts related to distributed database management systems. (Remembering)
- CO21: Design a distributed database framework by considering all the necessary and preliminary set-up requirements. (Creating)
- CO22: Apply methods and techniques for distributed query processing and optimization (Applying)
- CO23: Analyze the various constraints so as to be able to provide security to data in a distributed database environment (Analyzing)
- CO24: Understand the importance of reliability in handling deadlocks and concurrency problem. (Understanding)

Course Outline-

| Topics | Module Reference (from Course Text) |
|---|--|
| <p>Module I: Features of Distributed versus Centralized Databases, Distributed Database Management Systems (DDBMSs), Review of Databases and Computer Networks</p> <p>Reference Architecture for Distributed Databases, Types of Data Fragmentation, Distribution Transparency for Read-Only Applications, Distribution Transparency for Update Applications, Distributed Database Access Primitives, Integrity Constraints in Distributed Databases</p> | <p>Chapter 1- Distributed Databases- An Overview Chapter 2- Review of Databases and Computer Networks Chapter 3- Levels of Distribution Transparency</p> |
| <p>Module II: A Framework for Distributed Database Design, Design of Database Fragmentation, Allocation of Fragments</p> <p>Equivalence Transformation for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries</p> | <p>Chapter 4- Distributed Database Design Chapter 5- Translation of Global Queries to Fragment Queries</p> |
| <p>Module III: Framework for Query Optimization, Join Queries, General Queries</p> <p>Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions</p> | <p>Chapter 6- Optimization of Access Strategies Chapter 7- Transaction Management</p> |

| | |
|--|--|
| <p>Module IV:</p> <p>Foundations of Distributed Concurrency Control, Distributed Deadlocks</p> <p>Reliability – Basic Concepts, Nonblocking commitment protocols, Reliability and Concurrency control, Checkpoints and Cold Restart</p> | <p>Chapter 8- Distributed Concurrency Control</p> <p>Chapter 9 - Reliability</p> |
|--|--|

Course Text- Ceri, Stefano, and Giuseppe Pelagatti. Distributed databases principles and systems. McGraw-Hill, Inc., 1984.

Suggested Readings

23. Stefano, and Giuseppe Pelagatti. Distributed databases principles and systems. McGraw-Hill, Inc., 1984.
24. Rahimi, Saeed K., and Frank S. Haug. Distributed database management systems: A Practical Approach. John Wiley & Sons, 2010.
25. Burleson, Donald K. Managing distributed databases: building bridges between database islands. Wiley-QED Publishing, 1995.
26. Course Text: Self Learning Material (*SLM) provided by the University

Mapping of COs with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 |
|-----------------|----------|----------|----------|----------|
| CO1 | H | | | |
| CO2 | L | H | | |
| CO3 | | M | H | |
| CO4 | H | | | |
| CO5 | | | | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | | | | | | | | |
| CO2 | M | | | | | | | H |
| CO3 | | | | | | L | | M |
| CO4 | H | | | | L | | | |
| CO5 | M | | L | | | | M | |

Course Name: Environment Studies

Course Outline:

| Topics | Reference from Text Book |
|---|---|
| Definition, Scope, Importance, Institutions in Environment, People In Environment | The Multidisciplinary Nature of Environmental Studies |

| | |
|--|--|
| <p>Introduction and Natural Resources and Associated Problems</p> <p>a) Forest Resources: Use And Over-Exploitation, Deforestation, Case Studies. Timber Extraction, Mining, Dams and their Effects on Forests and Tribal People.</p> <p>b) Water Resources: Use And Over-Utilization Of Surface and Ground Water, Floods, Drought, Conflicts over Water, Dams – Benefits and Problems</p> <p>c) Mineral Resources: Use and Exploitation, Environmental Effects of Extracting and using Mineral Resources, Case Studies</p> | <p>Natural Resources</p> |
| <p>a) Food Resources: World Food Problems, Changes caused by Agriculture and Overgrazing, Effects of Modern Agriculture, Fertilizer/ Pesticide Problems, Water Logging, Salinity, Case Studies</p> <p>b) Energy Resources: Growing Energy Needs, Renewable/ Non-renewable Energy Sources, Use Of Alternate Energy Sources, Case Studies</p> <p>c) Land Resources: Land as a Resource, Land Degradation, Man-Induced Land-Slides, Soil Erosion and Desertification.</p> <p>(i) Role of an Individual in Conservation of Natural Resources</p> <p>(ii) Equitable Use of Resources for Sustainable Lifestyles</p> | <p>Food, Energy and Land Resources</p> |
| <p>Concept of an Ecosystem</p> <p>(i) Structure and Functions of an Ecosystem,</p> <p>(ii) Producers, Consumers and Decomposers</p> <p>(iii) Energy Flow in the Ecosystem</p> <p>(iv) Ecological Succession</p> <p>(v) Food Chains, Food Webs And Ecological Pyramids</p> | <p>Ecosystems</p> |

| | |
|---|---|
| <p>Introduction, Types, Characteristic Features, Structure and Function of the Following Ecosystem:</p> <ul style="list-style-type: none"> (i) Forest Ecosystem (ii) Grassland Ecosystem (iii) Desert Ecosystem (iv) Aquatic Ecosystems (Ponds, Lakes, Streams, Rivers, Estuaries, Oceans) | <p>Types of Ecosystem</p> |
| <ul style="list-style-type: none"> (i) Introduction – Definition: Genetic, Species, Ecosystem Diversity (ii) Biogeographic Classification of India (iii) Value Of Biodiversity: Consumptive, Productive Use, Social, Ethical, Aesthetic and Option Values (iv) Biodiversity at Global, National and Local Levels (v) India as a Mega Diversity Nation (vi) Hotspots of Biodiversity. Threats to Biodiversity: Habitat Loss, Poaching of Wildlife, Man-Wildlife Conflicts (vii) Endangered and Endemic Species of India (viii) Conservation of Biodiversity: In-Situ and Ex-Situ | <p>Biodiversity And Its Conservation</p> |
| <p>Definition, Causes, Effects and Control Measures of:</p> <ul style="list-style-type: none"> (i) Air Pollution (ii) Water Pollution (iii) Soil Pollution (iv) Marine Pollution (v) Noise Pollution (vi) Thermal Pollution (vii) Nuclear Hazards | <p>Environmental Pollution</p> |
| <ul style="list-style-type: none"> (i) Solid Waste Management: Types, Collection, Causes, Effects and Control Measures of Municipal Urban and Industrial Wastes. (ii) Role of an Individual in Prevention of Pollution (iii) Pollution Case Studies (iv) Disaster Management: Floods, Earthquakes, Cyclones, Landslides | <p>Solid Waste Management and Disaster Management</p> |

| | |
|---|---|
| <p>(i) From Unsustainable to Sustainable Development (ii) Urban Problems Related to Energy (iii) Water Conservation, Rain Water Harvesting, Watershed Management (iv) Resettlement and Rehabilitation of People; its Problems and Concerns. Case Studies</p> | <p>Social Issues and the Environment</p> |
| <p>(i) Environmental Ethics: Issues and Possible Solutions (ii) Climate Change, Global Warming, Acid Rain, Ozone Layer Depletion, Nuclear Accidents and Nuclear Holocaust. Case Studies (iii) Wasteland Reclamation (iv) Consumerism and Waste Products</p> | <p>Environmental Issues</p> |
| <p>(i) Environment Protection Act (ii) Air (Prevention and Control of Pollution) Act (iii) Water (Prevention and Control of Pollution) Act (iv) Wildlife Protection Act (v) Forest Conservation Act (vi) Issues Involved in Enforcement of Environmental Legislation</p> | <p>Environmental Legislation</p> |
| <p>(i) Population Growth, Variation Among Nations (ii) Population Explosion – Family Welfare Program</p> | <p>Human Population and the Environment</p> |
| <p>(i) Human Rights (ii) Value Education (iii) HIV/AIDS (iv) Women and Child Welfare</p> | <p>Environmental and Human Health</p> |
| <p>Introduction, Role of Information Technology in Environment Education, Environment Awareness Programs, Role of Public In Environment Education</p> | <p>Environment Education</p> |
| <p>Environment Economics, Environment Impact Assessment, Environment Management System And Life Cycle Assessment</p> | <p>Environmental Management</p> |

Course Text: Environmental Studies by Erach Bharucha

Bachelor of Computer Applications
Semester V

Course Name- Algorithm and Analysis

(4 credits - 60 hours)

Objective

The objective of the course is to learn how to design and analyse the algorithms and solve problems using C language. After the completion of this course, a student will be able to

- Understand the basics of computational algorithms.
- Understand the necessary mathematical abstraction to solve problems.
- Implement important algorithmic design paradigms and methods of analysis.

Course / Learning Outcomes

At the end of this course students will be able to:

CO1: Define algorithms, importance of analysis of an algorithm and their asymptotic bounds and relate the different types of problem and their solutions. (Remembering)

CO2: Explain different design strategies such as brute force, divide and conquer, decrease and conquer, dynamic programming, greedy and backtracking used for the design of algorithms. (Understanding)

CO3: Design algorithms for given problems. (Applying)

CO4: Compare and analyze the time and space complexity based on different design strategies. (Analyzing)

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|--|--|
| Module I: Define Algorithm, Fundamentals of Algorithm Problem Solving, Important Problem Types, Fundamental Data Structures. Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms, Example: Fibonacci Numbers | Chapter 1- Introduction Chapter 2- Fundamentals of the analysis of Algorithm Efficiency |
| Module II: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search. Insertion Sort, Depth-First Search and Breadth-First Search, Topological Sorting, Algorithms for Generating Combinatorial Objects. | Chapter 3- Brute Force and Exhaustive Search Chapter 4- Decrease-and-Conquer |
| Module III: Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals and Related Properties, Multiplication of Large Integers and Strassen's Matrix Multiplication Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction | Chapter 5- Divide-and-Conquer Chapter 6- Transform-and-Conquer |
| Module IV: Sorting by Counting, Input Enhancement in String Matching, Hashing. Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions | Chapter 7- Space and Time Tradeoffs Chapter 8- Dynamic Programming |
| Module V: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees. Lower Bound Arguments, Decision Trees, P, NP, and NP-Complete Problems | Chapter 9- Greedy Technique Chapter 11- Limitations of Algorithm Power |
| Module VI: Backtracking, Branch-and-Bound, Approximation Algorithms for NP-hard problems | Chapter 12- Coping with the Limitations of Algorithm Power |

Course Text- Introduction to the Design and Analysis of Algorithms by Anany Levitin, 3rd Edition, Pearson

Suggested Readings

1. S. Sridhar, Design and Analysis of Algorithms, Oxford University Press, 2014.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, The MIT Press, 2009.
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Universities Press, 2008.
4. Narsimha Karumanchi, Data Structures and Algorithms Made Easy, CareerMonk Publications; Fifth Edition, 2016
5. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V | Module VI |
|-----------------|----------|-----------|------------|-----------|----------|-----------|
| CO1 | H | M | | M | | |
| CO2 | H | H | H | H | M | |
| CO3 | M | H | H | H | H | |
| CO4 | M | | M | H | M | H |

Mapping of COs with Pos

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | M | | | M | M | | |
| CO2 | H | M | | | | | M | |
| CO3 | M | | M | | | | M | H |
| CO4 | M | | M | | | M | M | M |

(4 credits - 60 hours)

Objective

To develop a strong foundation in web technologies and understand the principles of three-tier architecture, PHP programming, database management using MySQL, validation and error handling, session management, and web application security.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

CO1: Analyze the concepts and components of three-tier architecture and its application in web development. (Analyzing)

CO2: Apply programming constructs in PHP, including conditions, branches, loops, functions, and working with different data types. (Applying)

CO3: Explain database operations using MySQL, including managing databases and tables, performing data manipulation with SQL queries, and utilizing PHP for database interactions. (understanding)

CO4: To have a solid understanding of how to validate user input, manage sessions, authenticate users, handle errors, and ensure data security in web applications. (Remembering)

Course Outline-

| Topics | Reference from Text Book |
|---|--|
| Module I: The Web, Three-Tier Architecture | Unit 1- Database Applications and the Web |
| Overview of PHP, Conditions and Branches, Loops, Functions, Working with Types, User- Defined Functions | Unit 2- The PHP Scripting Language |
| Arrays, Strings, Regular Expressions, Date and Times, Integers and Floats | Unit 3- Arrays, Strings and Advanced Data Manipulation in PHP |
| Classes and Objects, Inheritance, Throwing and Catching Exceptions | Unit 4- Introduction to Object-Oriented Programming with PHP 5 |
| Module II: Database Basics, MySQL Command Interpreter, Managing Databases and Tables, Inserting, Updating and Deleting Data, Querying with SQL SELECT, Join Queries, Case Study | Unit 5- SQL and MySQL |
| Querying a MySQL Database Using PHP, Processing User Input, MySQL Function Reference | Unit 6- Querying Databases |
| Overview on PEAR, Core Components, Packages | Unit 7- PEAR |
| Database Inserts, Updates and Deletes, Issues in Writing Data to Databases | Unit 8- Writing to Web Databases |
| Module III: Validation and Error Reporting Principles, Server-Side Validation with PHP, JavaScript and Client-Side Validation | Unit 9- Validation with PHP and JavaScript |
| Introducing Session Management, PHP Session Management, Case Study, Use of Session, PHP Session API and Configuration | Unit 10- Sessions |
| Module IV: HTTP Authentication, HTTP Authentication with PHP, Form-Based Authentication, Protecting Data on the Web | Unit 11- Authentication and Security |
| Errors, Common Programming Errors, Custom Error Handlers | Unit 12- Errors, Debugging and Deployment |

Course Text- PHP and MySQL by Huge E. Williams and David Lane.

Suggested Readings

6. PHP and MySQL Web Development" by Luke Welling and Laura Thomson.
7. Learning PHP, MySQL & JavaScript" by Robin Nixon.
8. PHP, MySQL, JavaScript & HTML5 All-in-One for Dummies" by Steve Suehring, Janet Valade, and Tricia Ballad.
9. Beginning PHP and MySQL: From Novice to Professional" by W. Jason Gilmore.
10. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV |
|-----------------|----------|-----------|------------|-----------|
| CO1 | H | M | | |
| CO2 | H | M | | |
| CO3 | | H | | |
| CO4 | | | H | H |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | L | | | | | | | |
| CO2 | | | | | | | H | M |
| CO3 | | | | | | | M | M |
| CO4 | L | | | | | | H | M |

Course Name- Internet Programming

(4 credits - 60 hours)

Objective

The objective of the course is to present an introduction to the fundamentals and basic requirements of Internet technologies. After completion of this course, students should be able to design and implement a website on their own by including client-side and server-side technologies.

COURSE / LEARNING OUTCOMES

At the end of this course students will be able to:

- CO25: Examine the growth and services of the Internet and recalling the history behind it. (Remembering)
- CO26: Illustrate and differentiate the various services provided by the internet. (Understanding)
- CO27: Experiment with various mark-up languages and style sheets. (Applying)
- CO28: Analyse the client requirements and accordingly design a dynamic website containing scripting languages. (Analysing/Creating)
- CO29: Summarize and validate a practical solution towards a web application development and also deploy a website of their own. (Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|--|--|
| Module 1: (12 Hours) Introduction to Internet – History of Internet, WWW, Web Browsers, Web Servers, URLs, MIME, HTTP, Internet Connectivity, WWW, Browser, Search Engine XHTML - Origins of HTML and XHTML, Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, Frames, Syntactic Differences between HTML and XHTML | Chapter 1 - Introduction to Computers and the Internet Chapter 2 - Introduction to HTML5: Part I Chapter 3 - Introduction to HTML5: Part II |
| Module 2: (18 Hours) Cascading Style Sheets - Introduction, Level of Style Sheets, Style Specification Formats, Selector Forms, Property Value Forms, Font Properties, Text Properties, Background properties, Absolute positioning, List Properties, Alignment of Text, The Box Model XML - Introduction, Syntax, Document Structure, Document Type Definitions, Namespaces, XML Schemas, Displaying Raw XML Documents, Displaying XML Documents with CSS, XSLT Style Sheets, XML Processors | Chapter 4 - Introduction to Cascading Style Sheets: Part I Chapter 15 – XML |
| Module 3: (18 Hours) JavaScript - Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions, Constructor, Pattern Matching Using Regular Expressions, Errors in Scripts, JavaScript Form Validation, Events in JavaScript Advanced JavaScript - Introduction to Dynamic Documents, Positioning Elements, Moving Elements, Element Visibility, | Chapter 6 – JavaScript: Introduction to Scripting Chapter 7 - Objects in JavaScript Chapter 8 - JavaScript: Control Statements I Chapter 9 - JavaScript: Control Statements II Chapter 10 - JavaScript: Functions Chapter 10 - JavaScript: Arrays Chapter 13 – JavaScript Event Handling |

Course Name-System Programming
(4 credits - 60 hours)

Objective:

This course aims to enable students to analyze and justify the usage, requirements, and functionality of system programming tools, and use data structures and algorithms to design efficient system programs.

COURSE / LEARNING OUTCOMES:

At the end of this course, students will be able to:

1. Define systems programs, assemblers, macro preprocessors, loaders, linkers, and software tools. *(Remembering)*
2. Understand the usage and functionality of software tools in system programming tasks. *(Understanding)*
3. Make use of different data structures and algorithms to design system programs. *(Applying)*
4. Analyze the function of the data structures used in designing assemblers, macro preprocessors, linkers, and loaders. *(Analyzing)*
5. Justify the requirements of multiple passes in designing assemblers, macro preprocessors, loaders, and linkers. *(Evaluating)*

Course Outline-

| Topics | Module Reference (from Course Text) |
|---|---|
| <p>Module 1: Introduction to System Programming (6 Hours) Overview of system programming and its importance, Comparison with application programming, Key concepts and tools used in system programming.</p> | Chapter 1: Introduction |
| <p>Module 2: Assembler Design (16 Hours)</p> <ol style="list-style-type: none"> a. Introduction: Sample assembly language, Overview of assembly language syntax, Types of assembly statements: imperative, assembler directives, and declaration statements, Handling labels, symbols, operands and literals in assembly language. Phases of Language Processing in Assemblers- Analysis Phase and Synthesis Phase. b. Two Pass Assembler: Algorithm for Pass I, Data structures, Intermediate code forms, Algorithm for Pass II, Generating the final object code, Reporting and handling errors during assembly. c. Single-Pass Assembler: Design and operation of a single-pass assembler, Handling forward references in a single-pass assembler | Chapter 3: Assemblers |
| <p>Module 3: Loader and Linkers (14 Hours)</p> <ol style="list-style-type: none"> a. Loaders: Overview of loader functions, Absolute loader, Bootstrap loader. b. Linkers: Linkers, Concepts of linking and relocation, Relocation algorithm, Self-relocating programs, Linking for overlays, Direct Linking Loader, and Design of a linker. | Chapter 5: Linkers and loaders |
| <p>Module 4: Macro Preprocessor Design (12 Hours) Macro definition and call, Macro expansion, Nested macro calls, Conditional expansion, Understanding the design requirements of a macro preprocessor, Design of macro preprocessor.</p> | Chapter 4: macro and Macro Preprocessor |
| <p>Module 5: Software tools (12 Hours) Introduction to text editors, Overview of interpreters and their usage in system programming, Introduction to program generators for system programming,</p> | Chapter 9: Software Tools |

| | |
|---|--|
| Overview of debug monitors and their role in system programming | |
|---|--|

Suggested Readings

Course Text: D.M.Dhamdhare, System Programming and Operating Systems, Second Edition, TMH.

Books:

1. John J. Donovan, Systems Programming, First Edition, TMH.
2. Leland L. Beck, System Software – An Introduction to System Programming, Third Edition, Pearson.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II | Module III | Module IV | Module V |
|-----------------|----------|-----------|------------|-----------|----------|
| CO1 | H | | | | |
| CO2 | | | | | H |
| CO3 | | H | H | H | |
| CO4 | | H | H | H | |
| CO5 | | H | H | H | |

Mapping of COs with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | | M | | | | | | |
| CO2 | | L | | | | | | |
| CO3 | | | | | | | H | M |
| CO4 | H | | | | | | | |
| CO5 | H | | | | | | | |

Course Name- Algorithm and Analysis Lab

(2 credits - 30 hours)

Objective

The objective of the course is to learn how to design and analyse the algorithms and solve problems using C programming language. After the completion of this course, a student will be able to

- Understand the basics of computational algorithms.
- Understand the necessary mathematical abstraction to solve problems.
- Implement important algorithmic design paradigms and methods of analysis.

Course / Learning Outcomes

At the end of this course students will be able to:

CO1: Recall existing algorithms and how to analyse them using graph notation. (Remembering)

CO2: Demonstrate implementation of the existing algorithms. (Understanding)

CO3: Apply existing algorithms in developing different applications. (Applying)

CO4: Evaluate an algorithm in terms of time and space efficiency. (Evaluating)

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|---|---|
| Module I: a) Design and analysis of on-Recursive and Recursive Algorithms, Example: Fibonacci Numbers. b) Implementation of basic sorting and searching algorithms, Example Selection Sort and Bubble Sort, Sequential Search, Insertion Sort, Depth-First Search and Breadth-First Search | Chapter 1- Introduction Chapter 2- Fundamentals of the analysis of Algorithm Efficiency Chapter 3- Brute Force and Exhaustive Search Chapter 4- Decrease-and-Conquer |
| Module II: a) Implement the Dynamic programming technique and analyse the algorithm showing the graph notation. b) Implement the Transform-and-Conquer technique and analyse the algorithm showing the graph notation. c) Implement the Greedy programming technique analyse the algorithm showing the graph notation. d) Implement the Divide and Conquer technique and analyse the algorithm showing the graph notation. | Chapter 5- Divide-and-Conquer Chapter 6- Transform-and-Conquer Chapter 8- Dynamic Programming Chapter 9- Greedy Technique |

Course Text- Introduction to the Design and Analysis of Algorithms by Anany Levitin.

Suggested Readings

1. S. Sridhar, Design and Analysis of Algorithms, Oxford University Press, 2014.

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, The MIT Press, 2009.
3. Narsimha Karumanchi, Data Structures and Algorithms Made Easy, CareerMonk Publications; Fifth Edition, 2016
4. Course Text: Self Learning Material (*SLM) provided by University.

Mapping of COs with Syllabus

| Course Outcomes | Module I | Module II |
|-----------------|----------|-----------|
| CO1 | H | M |
| CO2 | H | H |
| CO3 | M | H |
| CO4 | H | H |

Mapping of COs with Pos

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | M | | | M | M | | |
| CO2 | H | M | | | | | M | |
| CO3 | M | | M | | | | M | H |
| CO4 | H | | M | | | M | M | M |

(2 credits - 30 Hours)

COURSE / LEARNING OUTCOMES:

At the end of this course, students will be able to:

1. Understand the debugging techniques and will be able to utilise a debugger for system programming tasks effectively. (*Understanding*)
2. Apply assembler design principles; identify labels, literals, opcodes, and operands, and generate intermediate code (IC). (*Applying*)
3. Analyze and evaluate a macro preprocessor's design, including identifying macro definitions, creating data structures (MNT, MDT, and ALA), and analysing macro calls for expansion. (*Analysing and evaluating*)
4. Construct and comprehend essential data structures, such as ESD (External Symbol Directory) cards, GEST (Global External Symbol Table), and LESA (Local External Symbol Array), for the effective implementation of linkers and loaders. (*Creating*)

Course Outline-

| | |
|--|---|
| Module 1: - Designing an Assembler (10 Hours) File tokenisation, identifications of labels, literals, opcodes and operands, LC (location counter) processing, creating of SYMTAB (Symbol Table), OPTAB (Opcode Table), LITAB (Literal Table) and POOLTAB (Pool Table), generation of intermediate code (IC). | Chapter 3: Assemblers |
| Module 2: Designing a Macro Preprocessor(7 Hours) a. Pass-I: Identification of macro definitions and creation of MNT (Macro Name Table), MDT (Macro Definition Table), and ALA (Argument List Array) for Pass-I. b. Pass-II: Identification of macro calls and macro expansion, creating ALA for Pass-II. | Chapter 4: Macro and Macro pre-processor |
| Module 3: Basics of Linker and Loader (7 Hours) Construction of ESD (External Symbol Directory) card, Construction of GEST (Global External Symbol Table) and LESA (Local External Symbol Array) from an ESD card | Chapter 5: Linkers and Loaders |
| Module 4: Software tools (6 Hours) Introduction to debugging techniques and usage of a debugger | Chapter 9: Software Tools |

Suggested Readings

Course Text: D.M.Dhamdhare, System Programming and Operating Systems, Second Edition, TMH.

Books:

1. John J. Donovan, Systems Programming, First Edition, TMH.
2. Leland L. Beck, System Software – An Introduction to System Programming, Third Edition, Pearson.

Bachelor of Computer Applications

Semester VI

Course Name- MIS and Enterprise Resource Planning
(4 credits - 60 hours)

Objective

The main objective of this course is to make the student understand the conceptual elements of Management Information System and ERP and its theory and implementation. This is especially poignant in view of large number of organizations implementing MIS and ERP applications in recent years. The student will appreciate the impact that MIS and ERP brings into the daily operations of firms with respect to their productivity, integration, communication, etc

COURSE/LEARNINGOUTCOMES

At the end of this course students will be able to:

CO1: Recall the conceptual elements of MIS and ERP. (Remembering)

- CO2: Demonstrate the Influence of MIS and ERP in Large Organizations. (Understanding)
 CO3: Identify the impact into the daily operations of firms with respect to their productivity, integration, communication etc. (Applying)
 CO4: Analyse the practical side of MIS and ERP implementation with different vendors. (Analysing)
 CO5: Discuss and evaluate the best practices with various case studies and real time examples. (Creating, Evaluating)

Course Outline-

| Topics | Module Reference (from Course Text) |
|---|--|
| <p>Module I (15 hours)</p> <p>e. Evolution and structure of ERP, ERP concepts, growth of the ERP market, conceptual model of ERP, 2-tier and 3-tier architecture, elements in ERP architecture, advantages/benefits of ERP, overview of an enterprise, integrated management information, business modeling, integrated data model</p> <p>f. ERP and related technologies: Business Process Reengineering (BPR), Management Information Systems (MIS), Decision Support Systems (DSS), Data Warehousing, Data Mining, Online Analytical Processing (OLAP), Supply Chain Management.</p> | <p>Chapter 1- Introduction to ERP</p> <p>Chapter 2-Enterprise – An Overview</p> <p>Chapter 3-ERP and Related Terminologies</p> |
| <p>Module II (15 hours)</p> <p>M</p> <p>Item types in ERP, Manufacturing, distribution and Financial requirements, item control module in ERP, Finance module, Manufacturing and Production Planning module, Sales and Distribution module, Plant Maintenance module, Quality Management module, Materials Management module, Capital Requirement Planning module, Purchase Control module, Human Resources modules; concept of Bill of materials, concept of formula management.</p> | <p>Chapter 4-A manufacturing perspective</p> <p>Chapter 5-ERP Modules</p> |
| <p>Module III (10 hours)</p> <p>a. SAP AG: R/3 –, overview of R/3 system, R/3 modules, R/3 and the internet</p> <p>b. BAAN: Baan ERP modules, Baan ERP Tools</p> | <p>Chapter 7-ERP Market</p> |

| | |
|--|---|
| <p>c. Oracle: Oracle modules – Financials, Human Resources, Projects, Manufacturing, Supply chain.</p> <p>d. PeopleSoft: Accounting and control, Treasury Management, Performance Management, Sales and Logistics, Procurement</p> | |
| <p>Module IV (10 hours)</p> <p>Elements of implementation methodology, Pre-evaluation Screening, Package evaluation, project planning phase, Gap Analysis, Business Process Re-engineering, configuration, Implementation team training, testing, product migration and support, Problems in ERP implementation, cost of ERP.</p> | <p>Chapter 8-Implementation Lifecycle</p> |
| <p>Module V (10 hours)</p> <p>a. Concept of Best Practices, concept of Customer Order Decoupling Point(CODP), Demand Management – Sales and Operations Planning, ERP scenario in India, future directions in ERP.</p> <p>b. Case studies should also be introduced to highlight situations where ERP projects are implemented, and the success stories/benefits/difficulties of these implementations.</p> | <p>Chapter 11-ERP Case studies</p> |

Course Text- Alexis Leon, ERP Demystified, Tata McGraw Hill Pub. Co. Ltd, 2000

Suggested Readings

1. O’Leary, Daniel E, Enterprise Resource Planning Systems: systems, life cycle, electronic commerce and risk, Cambridge University Press.
2. Alexis Leon, Enterprise Resource Planning, 14th reprint, Tata McGraw Hill, New Delhi 2005
3. Rahul V Altekhar, Enterprise Resource Planning (Theory and Practice), Prentice Hall India, New Delhi 2004
4. Alexis Leon, ERP Demystified, Tata McGraw Hill Pub. Co. Ltd, 2000
5. Kent Sandoe, Enterprise Integration, John Wiley and Sons
6. Garg and Venkitakrishnan, Enterprise Resource Planning : Concepts and Practice, 2nd edition, Prentice Hall India
7. Garg and Venkitakrishnan, ERPWARE: ERP Implementation Framework, Prentice Hall India
8. ERP – Concepts and Cases, ICFAI University Press, 2004

Mapping of Cos with Syllabus:

| Course Outcomes | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 |
|-----------------|----------|----------|----------|----------|----------|
|-----------------|----------|----------|----------|----------|----------|

| | | | | | |
|-----|---|---|---|---|---|
| CO1 | H | | | | |
| CO2 | | M | H | | |
| CO3 | | H | M | | |
| CO4 | | L | | H | |
| CO5 | | | | | H |

Mapping of Cos with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | M | | L | | M | M | H | H |
| CO2 | M | L | | | L | M | M | H |
| CO3 | H | | M | M | M | | H | H |
| CO4 | H | M | H | | H | M | M | |
| CO5 | H | M | | | H | M | M | H |

Course Name: PROJECT WORK (8 credits – 120 hours)

Objective : This course aims to provide the student with various practical knowledge on different problem specific system deployment techniques and analysis.

COURSE / LEARNING OUTCOMES

At the end of the project work students will be able to:

1. Recall different process involved in software development life cycle (Remembering).
2. Understanding the problem statement and related literature study (Understanding).
3. Analyse end user requirements for identifying system functionality metrics and decide whether developed system can meet end user requirements. (Analysing)
4. Choose an engineering approach to solving problems, starting from the acquired knowledge to visual interpretation. (Evaluating)
5. Propose the use of certain technologies by implementing them in different programming languages to solve the problem statement (Creating).

Course Outline-

| Topics | Module Reference (<i>from Course Text</i>) |
|---|---|
| Module I: (10 Hours) a. Feasibility study of the problem statement and solution, evaluation of existing system b. Finding the limitation of existing system. | Chapter 2: A Generic view of process |
| Module II: (20 Hours) a. Literature study b. Identification of software and hardware requirement c. Schedule feasibility, economic and operational | Chapter 3: Process Models Chapter 5: Software Engineering Practice Chapter 7: Requirement |

| | |
|--|---|
| feasibility d. Identification software development metrics. | Engineering |
| Module III: (20 Hours) a. Evaluate Design diagrams for the solving the problem statement. b. Identify deployment strategy. | Chapter 8: Building the analysis model Chapter 9: Design Engineering |
| Module IV: (40 Hours) a. Develop algorithm and implement using programming languages. b. Resolve and Implement the problem statement through proposed software model and system architecture. | Chapter 12: Performing User Interface design Chapter 13: Testing Strategies Chapter 14: Testing Tactics |
| Module 5: (30 Hours) Prepare a report to highlight the problem statement, design diagram, technologies deployed and references used to resolve end users requirement. | Chapter 23: Estimation Chapter 24: Project Scheduling |

Course Text- Roger S. Pressman, “Software Engineering–A Practitioner’s Approach”, McGraw Hill Companies.

Suggested Readings:

1. Richard E Fairley, Software Engineering Concepts, Tata McGraw Hill Publishing Company Limited
2. Rajib Mall, Fundamentals of Software Engineering, PHI.
3. Ian Sommerville, Software Engineering, Sixth Edition, Addison Wesley, Pearson Education.

Mapping of COs to Modules

| | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 |
|-------------|----------|----------|----------|----------|----------|
| CO 1 | H | H | L | L | |
| CO 2 | H | H | | | |
| CO 3 | L | L | H | H | M |
| CO4 | | L | H | H | L |
| CO5 | | | M | H | H |

Mapping of Cos with POs

| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| CO1 | H | | M | | H | H | M | L |
| CO2 | H | | | | H | M | H | H |
| CO3 | M | M | H | M | M | M | M | |
| CO4 | H | H | H | | M | M | M | M |
| CO5 | H | M | M | M | M | M | M | H |