

GOVERNMENT ARTS COLLEGE (AUTONOMOUS)
COIMBATORE - 641 018



Learning Outcome-based Curriculum Framework (LOCF)
for
B.Sc. COMPUTER SCIENCE
(Effective from the Academic year 2021-2022)

POSTGRADUATE AND RESEARCH DEPARTMENT OF
COMPUTER SCIENCE

MAY-2021

CHAPTER	CONTENTS	PAGE NUMBER
	Preamble	3
1.	Introduction	6
2.	Learning Outcome – based Approach to Curriculum Planning and Development (LOACP)	9
3.	Graduate Attributes	14
4.	Qualification Descriptors	18
5.	Programme Learning Outcome in course	19
6.	Structure of B.Sc. course objectives, Learning Outcomes, Contents, Pedagogy Strategies and Reference	22
7.	Teaching Learning Methodologies	101
8.	Assessment Methods	103

Preamble

Education is the key to the development of any society. Higher education is crucial for securing the right kind of employment and pursuing further studies in best available world-class institutes within and outside India. Quality education in general and higher education, in particular, deserves high priority to enable the young and future generation of students to acquire skill, training, and knowledge to enhance their thinking, creativity, comprehension, and application abilities and prepare them to compete successfully, and excel globally. Sustained initiatives are required to reform the present higher education system to improve and upgrade the academic resources and learning environments by raising the quality of teaching and standards of achievement in learning outcomes across all undergraduate programs in science, humanities, commerce, and professional streams of higher education including computer science.

Over the past decades, the higher education system of our country has undergone substantial structural and functional changes resulting in both quantitative and qualitative development of the beneficiaries. Such changes have gained momentum with the introduction of the Choice Based Credit System (CBCS), which further expects learning outcome-based curriculum to maximize the benefits of the newly designed curriculum. The learning outcome-based curriculum will help the teachers of the discipline to visualize the curriculum more specifically in terms of the learning outcomes expected from the students at the end of the instructional process. It is pertinent to mention here that the purpose of education is to develop an integrated personality of the individual. The educational system provides all knowledge and skills to the learner for this.

Tamil Nadu State Council for Higher Education (TANSCH) has formed the State Integrated Boards of Studies, which, with great diligence and expertise, has devised the mandatory areas that have to be covered for three-year undergraduation and two-year postgraduation courses to realize the facilitation of the mobility of faculty and students from one university to another and to efficiently solve the problem of equivalence among courses. Great care has been taken to take 75% of the course content, and the individual institutions can decide the remaining 25%. The areas that must be covered by the student that are mandatory for earning the degree to have expected value have been worked out to gain enough knowledge in the subject concerned. The institutions should design 25% percent of the syllabus, and the areas covered under this also must have a weightage of 25%. Hence, it gives the autonomous institution seamless liberty on every Board of Studies (BOS) to innovate and experiment. More importantly, the institution devises appropriate strategies as mentioned below.

- (i) To make creative and critical applications of what has been learned in the mandatory components.
- (ii) To meaningfully connect the learners to the career demands and expectations. The theoretical subject knowledge of the students must be translated into practical hands-on experience.

One of the significant reforms in undergraduate education is introducing the Learning Outcomes-based Curriculum Framework (LOCF), making it student-centric, interactive, and outcome-oriented with well-defined aims, objectives, and goals to achieve. LOCF also aims to ensure uniform education standard and content delivery across the country, which will help the students ensure a similar quality of education irrespective of the institute and location. With University Grants Commission (UGC) initiatives for nation-wide adoption and implementation of the LOCF for bachelor's programmes in colleges, universities, and HEIs in general. A Core Expert Committee (CEC) was constituted to formulate the modalities for developing the LOCF in various subjects taught in undergraduate sciences, humanities, commerce, and professional courses. The CEC also constituted the Subject Expert Committees (SEC) to prepare detailed guidelines for the LOCF in subjects concerned.

The critical components of the planning and development of LOCF are given in terms of clear and unambiguous description of the Graduate Attributes (GA), Qualification Descriptors (QD), Program Learning Outcomes (PLO), and Course Learning Outcomes (CLO) to be achieved at the end of the successful completion of each undergraduate program to be offered by HEIs. In undergraduate education in Information Technology, the program of study leading to B.Sc. in Information Technology is discussed herewith.

The Qualification Descriptors (QD), Program Learning Outcomes (PLO), and the Course Learning Outcomes (CLO) were also finalized, keeping the general requirement of the program in view. The LOCF also gives general guidelines for the Teaching-Learning Process (TLP) corresponding to each component of the theory, experiment, tutorials, projects, and industrial / field visits to be followed to achieve the stated outcomes for each component. Finally, some suggestions for using various methods in the assessment and evaluation of learning levels of students are also made. It is a student-centric framework where they are expected to learn fundamentals of Information Technology along with the latest trends and techniques like Artificial Intelligence, Internet of Things, Machine

Intelligence, and advanced skillsets that include Mobile Application Development, Object Oriented Programming, and many other courses.

The Learning Outcomes (LO) specified by the CEC are the guidelines to determine the structure of the undergraduate programs offered by the Higher Educational Institutions (HEI) of our country. The critical components of the planning and development of LOCF are given in terms of clear and unambiguous description of the Graduate Attributes (GA), Qualification Descriptors (QD), Program Learning Outcomes (PLO), and Course Learning Outcomes (CLO) to be achieved at the end of the successful completion of each undergraduate program to be offered by HEIs.

1. Introduction

Computer Science (CS) has been evolving as an essential branch of science and engineering throughout the world in the last couple of decades. It has carved out a space for itself like any other basic science and engineering discipline. Computer science is a discipline that spans theory and practice, and it requires thinking both in abstract terms and in concrete terms. Nowadays, practically everyone is a computer user, and many people are even computer programmers. Computer Science can be seen on a higher level, as a science of problem-solving and problem solving requires precision, creativity, and careful reasoning. The ever-evolving discipline of computer science also has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively through computers. However, finding a solution requires expertise in Computer Science and knowledge in the particular application domain.

Computer science has a wide range of specialties viz., Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Computational Science, and Software Engineering. Mathematics is the origin of Computer Science, which provides the logical reasoning, scientists and engineers practice computer Science. Science provides the methodology for learning and refinement. Engineering provides the techniques for building hardware and software.

Universities and other HEIs introduced programs of studies in computer science as this discipline evolved itself into a multidisciplinary discipline. Information Technology is proliferating. Increasing applications of computers in almost all areas of human endeavor has led to vibrant industries with a rapid concurrent change in technology. Unlike other basic disciplines, developing a core competency in this discipline that can be reasonably stable becomes a challenge. Initially, it was introduced at the Master's (postgraduate) level as MCA and M.Tech in India. Later on, engineering programmes such as B.Tech. and B.E. in Computer Science & Engineering and Information Technology were introduced in various engineering Colleges/Institutions to cater to the growing demand for trained engineering human resources in IT industries. Parallely, BSc and MSc programmes specializing in Computer Science were introduced to train human resources in this highly demanding area. B.Sc. in Computer Science is being planned and introduced in different colleges and institutions.

Computer Science education at the undergraduate level (+3) will result in earning a Bachelor of Arts (BA) or Bachelor of Science (BS) degree in CS. The coursework required to earn a BSc is equally weighted in mathematics and science. B.Sc. in CS is aimed at undergraduate level training facilitating

multiple career paths. Graduated students can take up postgraduate programmes in CS leading to research & development, be employable at IT industries, or pursue a teachers' training programme such as B.Ed. in Computer Education, or can adopt a business management career. BSc with CS aims at laying a solid foundation of CS at an early stage of the career along with two other subjects such as Physics, Maths, Electronics, Statistics. There are several employment opportunities; after completing an undergraduate in CS, graduating students can fetch employment directly in companies as Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The Learning Outcome-based Curriculum Framework in Computer Science aims to allow flexibility and innovation in design and development of course content, method of imparting training, teaching learning process, and assessment procedures of the learning outcomes. It helps students learn to solve problems, accomplish IT tasks, and express creativity, both individually and collaboratively. The proposed framework would help students learn programming skills and the syntax of one or more programming languages.

Many of the learning outcomes of Computer Science can be achieved only by programming a computer for several different meaningful purposes. Therefore, all students must have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students learn to adapt to changes in programming languages and learn new languages as they are developed.

The present Learning Outcome-based Curriculum Framework for bachelor's degrees in CS is intended to facilitate the students to achieve the following.

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with a good foundation on theory, systems, and applications such as algorithms, data structures, data handling, data communication, and computation.
- To develop the ability to use this knowledge to analyze new situations
- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit the purpose mentioned above.

- The ability to synthesize the acquired knowledge, understanding, and experience for a better and improved comprehension of the real-life problems
- To learn skills and tools like mathematics, statistics, physics, and electronics to find the solution, interpret the results, and make predictions for future developments.

1.1 Course Structure – Types of Courses.

The following types of courses are offered under CBCS-LOCF:

- a) **Core Courses (CC).** A core course is a compulsory discipline-specific course. A student of Computer Science needs to take 25 courses, including 15 theory and 10 practical, over six semesters.
- b) **Elective Courses (EC).** An elective course is a course that is to be chosen from a specified set of courses. These courses are of two types. Which may be very specific or specialized or advanced or supportive to the discipline/subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.
 - i. **Discipline-Specific Electives (DSE).** These are elective courses that provide advanced undergraduate training in specialized areas of Computer Science. A set of six semester-specific courses of this kind are offered in the first through the fourth semester of the Computer Science Undergraduate programme.
 - ii. **Project.** An elective course designed to acquire special/advanced knowledge, such as supplement study/support study for project work. A candidate studies such a course independently with advisory support by a teacher/faculty member is called dissertation/project. Such a course is compulsory in the sixth semester.
 - iii. **Generic Electives (GE).** These courses, in disciplines other than Computer Science, are intended to broaden the training of a student in the Computer Science Undergraduate programme. A student of Computer Science will take one such course, offered by another department, in each of Semester V and VI.

2. Curriculum Planning- Learning Outcomes-based Approach

2.1 Nature and Extent of the B.Sc. (CS) Programme

The undergraduate programs in Computer Science build on science-based education at the +2 level. The +2 senior secondary school education aims and achieves a sound grounding in understanding the basic scientific temper with an introduction to the computation process by introducing some programming languages. This prepares a young mind to launch a rigorous investigation of the exciting world of computer science.

Framing and implementation of curricula and syllabi is envisaged to provide an understanding of the basic connection between theory and experiment and its importance in understanding the foundation of computing. This is very critical in developing scientific temperament and venture a career with a broad spectrum of applications and theoretical investigations. The undergraduate curriculum provides students with theoretical foundations and practical experience in computers hardware and software aspects. The curriculum in computer science is integrated with courses in the sciences and the humanities to offer a broad education yet of enough depth and relevance to enhance student employment opportunities upon graduation. As a Bachelor's degree program, the curriculum is based on the criterion that graduates are expected to function successfully in a professional employment environment immediately upon graduation.

The undergraduate program in Computer Science is presently being offered through the courses designed for granting the following degrees by various colleges and universities in India. All the courses are of 3-year duration spread over six semesters. B.Sc. with Computer Science or Bachelor of Science with Computer Science is a general multidiscipline bachelor programme. The programme has a balanced emphasis on three science subjects, one of which is computer science. A student studying B.Sc. with Computer Science is required to choose two other subjects from a pool of subjects: Physics, Mathematics, Statistics, Electronics, and Chemistry. Different institutions offer a different choices of combinations of subjects. The most popular combinations are Physics and Mathematics, Physics and Electronics, Mathematics and Electronics. However, there are also combinations like Statistics and Economics or Commerce and Economics.

B.Sc. in Computer Science

B.Sc. (CS) in India is generally a three-year degree program that develops advanced theoretical and research skills in a subject in which Honours is opted. It is a specialized programme specializing in

one science subject and another auxiliary science subject. This programme helps in building an advanced professional or academic career. It is an appropriate course for students who wish to pursue a Master of Science (M.Sc.) or Doctor of Philosophy (Ph. D.) and research or academic career. This program facilitates students who wish to pursue an independent research project in an area of interest under the supervision of an academic. B.Sc. (CS) differs from BSc in the number of courses in the subject in which Honours is opted. Thus B.Sc. (CS) has more CS courses than that of BSc programme.

B.Sc. in CS follows CBCS structure as mandated by UGC. In accordance with CBCS guidelines, the courses are categorized into compulsory courses, elective courses, ability enhancement courses. These categories of courses are discussed below, keeping the present context of undergraduate education in CS in mind.

2.2 Types of Courses

2.2.1 Core Course (CC)

A core course is a mandatory course required for the degree. A core course of the study refers to a series or selection of courses that all students must complete before they can move on to the next level in their education or earn a diploma. The general educational purpose of a core course of study is to ensure that all students take and complete academically and culturally essential courses. These courses teach students the foundational knowledge and skills to secure a specific degree or diploma. The core courses are designed with an aim to cover the basics that is expected of a student to imbibe in that particular discipline. Thus, a course that a candidate should compulsorily study as a core requirement is termed as a Core course. The present document specifies the core courses for B.Sc. The courses (as referred popularly) under this category will be taught uniformly across all universities with 30% deviation proposed in the draft. The purpose of fixing core papers is to ensure that all the institutions follow a minimum common curriculum so that each institution/ university adheres to the common minimum standard.

2.2.2 Electives

Generally, a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/ subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course. Different types of elective courses mandated in the present framework are the following.

- Domain Specific Elective (DSE)
- Generic Elective (GE)
- Ability Enhancement Elective (AEEC)

2.2.3 Discipline Specific Elective (DSE)

Elective courses offered under the main discipline/subject of study is referred to as Discipline Specific Elective. The list provided under this category are suggestive in nature and HEI has freedom to suggest its own papers under this category based on their expertise, specialization, requirements, scope and need. The University/Institute may also offer discipline-related elective courses of interdisciplinary nature (to be offered by the main discipline/subject of study).

2.2.4 Generic Elective (GE)

An elective course chosen from another discipline/subject with an intention to seek exposure beyond the discipline/s of choice is called a Generic Elective. The purpose of this category of papers is to offer the students the option to explore disciplines of interest beyond the choices they make in Core and Discipline Specific Elective papers. The list provided under this category are suggestive in nature and HEI can design its own papers under this category based on available expertise, specialization, and contextual requirements, scope and need.

2.2.5 Dissertation/Project

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

2.2.6 Ability Enhancement Courses (AEC)

The Ability Enhancement Courses may be of two kinds:

A. Ability Enhancement Compulsory Courses (AECC): AECC are based on the content that leads to knowledge enhancement. These are mandatory for all disciplines. Ability Enhancement Compulsory Courses (AECC) are the following.

- AECC-I English
- AECC-II English/Hindi/ MIL Communications
- AECC-III Environment Science

B. Skill Enhancement Courses (SEC): SEC courses are value-based and/or skill-based and are aimed at providing hands-on-training, competencies, skills, etc. SEC are at least 2 courses for Honours courses and 4 courses for General bachelor programmes. These courses may be chosen from a pool of courses designed to provide value-based and/or skill-based knowledge and should contain both theory and lab/hands-on/training/field work. The main purpose of these courses is to provide students life-skills in hands-on mode to increase their employability. The list provided under this category are suggestive in nature and each university has freedom to suggest their own papers under this category based on their expertise, specialization, requirements, scope and need.

2.2.7 Practical/Tutorial

For each core course and DSE course there will be one practical. The list of practical's provided is suggestive in nature and each university has the freedom to add/subtract/edit practical from the list depending on their faculty and infrastructure available. Addition will however be of similar nature.

2.3 Aims of Bachelor of Science Programmes in Computer Science

The Bachelor of Science degree in Computer Science emphasizes problem solving in the context of algorithm development and software implementation and prepares students for effectively using modern computer systems in various applications. The curriculum provides required computer science courses such as programming languages, data structures, computer architecture and organization, algorithms, database systems, operating systems, and software engineering; as well as elective courses in artificial intelligence, computer-based communication networks, distributed computing, information security, graphics, human-computer interaction, multimedia, scientific computing, web technology, and other current topics in computer science. The main aim of this Bachelor's degree is to deliver a modern curriculum that will equip graduates with strong theoretical and practical backgrounds to enable them to excel in the workplace and to be lifelong learners. The purpose of the BS programs in computer science are twofold: (1) to prepare the student for a position involving the design, development and implementation of computer software/hardware, and (2) to prepare the student for entry into a program of postgraduate study in computer science/engineering and related fields.

The Bachelor of Science program with Computer Science as one subject (BSc with CS) and the Bachelor of Science Honours programme in Computer Science (BSc (CS)) focus on the concepts and techniques used in the design and development of software systems. Students in this program explore

the conceptual underpinnings of Computer Science -- its fundamental algorithms, programming languages, operating systems, and software engineering techniques. In addition, students choose from a rich set of electives that includes data science, computer graphics, artificial intelligence, database systems, computer architecture, and computer networks, among other topics. A generous allotment of free electives allows students to combine study in computer science with study in auxiliary fields to formulate a program that combines experiences across disciplines.

3. Graduate Attributes

Graduate Attributes (GA) are the qualities, skills, and understandings students should develop during their time with the HEI. These are qualities that also prepare graduates as agents of social good in the future. Graduate Attributes can be viewed as qualities in the following subcategories.

- Knowledge of the discipline
- Creativity
- Intellectual Rigour
- Problem Solving and Design
- Ethical Practices
- Lifelong Learning
- Communication and Social Skills

Among these attributes, categories attributes under Knowledge of the Discipline are specific to a programme of study.

3.1.a. Knowledge of Discipline of CS

Knowledge of a discipline is defined as "command of a discipline to enable a smooth transition and contribution to professional and community settings. This Graduate Attribute describes the capability of demonstrating comprehensive and considered knowledge of a discipline. It enables students to evaluate and utilize information and apply their disciplinary knowledge and professional skills in the workplace.

3.1.b. Creativity

Creativity is a skill that underpins most activities, although this may be less obvious in some disciplines. Students are required to apply imaginative and reflective thinking to their studies. Students are encouraged to look at the design or issue through differing and novel perspectives. Creativity allows the possibility of a powerful shift in outlook and enables students to be open to thinking about different concepts and ideas.

3.1.c. Intellectual Rigour

Intellectual Rigour is the commitment to excellence in all scholarly and intellectual activities, including critical judgment. The students are expected to have clarity in thinking. This capability

involves engaging constructively and methodically when exploring ideas, theories and philosophies. It also relates to the ability to analyse and construct knowledge with depth, insight and intellectual maturity.

3.1.d. Problem Solving and Design

Problem-solving skills empower students not only within the context of their programmes, but also in their personal and professional lives. Many employers cite good problem-solving skills as a desired attribute that they would like graduates to bring to the workplace. With an ability to seek out and identify problems, effective problem solvers are able to actively engage with a situation, think creatively, to consider different perspectives to address identified challenge, to try out possible solutions and subsequently evaluate results as a way to make decisions. Through this process they can consolidate new and emergent knowledge and develop a deeper understanding of their subject discipline.

3.1.e. Ethical Practices

Ethical practice is a key component of professionalism and needs to be instilled in curricula across courses. When operating ethically, graduates are aware that we live in a diverse society with many competing points of view. Ethical behaviour involves tolerance and responsibility. It includes being open-minded about cultural diversity, linguistic difference, and the complex nature of our world. It also means behaving appropriately towards colleagues and the community and being sensitive to local and global social justice issues.

3.1.f. Life-Long Learning

The skill of being a lifelong learner means a graduate is open, curious, willing to investigate and consider new knowledge and ways of thinking. This flexibility of mind means they are always amenable to new ideas and actively seek new ways of learning or understanding the world.

3.1.g. Communication and Social Skills

The ability to communicate clearly and work well in a team setting is critical to sustained and successful employment. Good communication and social skills involve the ability to listen to and clearly express information back to others in a variety of ways - oral, written, and visual - using a range of technologies.

3.1.h. Self-Management

Graduates must-have capabilities for self-organization, self-review, personal development, and life-long learning.

3.2 LIST OF GRADUATE ATTRIBUTES for B.Sc. (CS)

Afore-mentioned GAs can be summarized in the following manner.

- GA 1.** A commitment to excellence in all scholarly and intellectual activities, including critical judgement
- GA 2.** Ability to think carefully, deeply and with rigour when faced with new knowledge and arguments.
- GA 3.** Ability to engage constructively and methodically when exploring ideas, theories and philosophies.
- GA 4.** Ability to consider other points of view and make a thoughtful argument.
- GA 5.** Ability to develop creative and effective responses to intellectual, professional and social challenges.
- GA 6.** Ability to apply imaginative and reflective thinking to their studies.
- GA 7.** Commitment to sustainability and high ethical standards in social and professional practices.
- GA 8.** To be open-minded about cultural diversity, linguistic difference, and the complex nature of our world.
- GA 9.** Ability to be responsive to change, to be inquiring and reflective in practice, through information literacy and autonomous, self-managed learning.
- GA 10.** Ability to communicate and collaborate with individuals, and within teams, in professional and community settings.
- GA 11.** Ability to communicate effectively, comprehending and writing effective reports and design documentation, summarizing information, making effective oral presentations and giving and receiving clear oral instructions.
- GA 12.** Ability to demonstrate competence in the practical art of computing by showing in design an understanding of the practical methods, and using modern design tools competently for complex real-life IT problems.
- GA 13.** Ability to use a range of programming languages and tools to develop computer programs and systems that are effective solutions to problems.

- GA 14.** Ability to understand, design, and analyse precise specifications of algorithms, procedures, and interaction behaviour.
- GA 15.** Ability to apply mathematics, logic, and statistics to the design, development, and analysis of software systems.
- GA 16.** Ability to be equipped with a range of fundamental principles of Computer Science that will provide the basis for future learning and enable them to adapt to the constant rapid development of the field.
- GA 17.** Ability of working in teams to build software systems.
- GA 18.** Ability to identify and to apply relevant problem-solving methodologies.
- GA 19.** Ability to design components, systems and/or processes to meet required specifications.
- GA 20.** Ability to synthesise alternative/innovative solutions, concepts and procedures.
- GA 21.** Ability to apply decision making methodologies to evaluate solutions for efficiency, effectiveness and sustainability.
- GA 22.** A capacity for self-reflection and a willingness to engage in self-appraisal.
- GA 23.** Open to objective and constructive feedback from supervisors and peers.
- GA 24.** Able to negotiate difficult social situations, defuse conflict and engage positively in purposeful debate.

4. Qualification Descriptors

Qualification descriptors are generic statements of the outcomes of study. Qualification descriptors are in two parts. The first part is a statement of outcomes, achievement of which a student should be able to demonstrate for the award of the qualification. This part will be of interest to those designing, approving and reviewing academic programmes. They will need to be satisfied that, for any programme, the curriculum and assessments provide all students with the opportunity to achieve, and to demonstrate achievement of, the outcomes. The second part is a statement of the wider abilities that the typical student could be expected to have developed. It will be of assistance to employers and others with an interest in the general capabilities of holders of the qualification. The framework has the flexibility to accommodate diversity and innovation, and to accommodate new qualifications as the need for them arises. It should be regarded as a framework, not as a straitjacket.

4.1. Qualification Descriptor for B.Sc. with CS

On completion of B.Sc. with Computer Science, the expected learning outcomes that a student should be able to demonstrate are the following.

- QD 1.** Fundamental understanding of the principles of Computer Science and its connections with other disciplines.
- QD 2.** Procedural knowledge that creates different types of professionals related to Computer Science, including research and development, teaching and industry, government and public service.
- QD 3.** Skills and tools in areas related to computer science and current developments in the academic field of study.
- QD 4.** Use knowledge, understanding and skills required for identifying problems and issues, collection of relevant quantitative and/or qualitative data drawing on a wide range of sources, and their application, analysis and evaluation using methodologies as appropriate to Computer Science for formulating solutions
- QD 5.** Communicate the results of studies undertaken in Computer Science accurately in a range of different contexts using the main concepts, constructs and techniques
- QD 6.** Meet one's own learning needs, drawing on a range of current research and development work and professional materials.
- QD 7.** Apply Computer Science knowledge and transferable skills to new/unfamiliar contexts.
- QD 8.** Demonstrate subject-related and transferable skills that are relevant to industry and employment opportunities.

5. Programme Learning Outcomes

These outcomes describe what students are expected to know and be able to do by the time of graduation. They relate to the skills, knowledge, and behaviours that students acquire in their graduation through the program

5.1. Programme Learning Outcomes for BSc with CS

The Bachelor of Science with Computer Science (BSc with CS) program enables students to attain, by the time of graduation:

- PLO-A.** Demonstrate the aptitude of Computer Programming and Computer based problem solving skills.
- PLO-B.** Display the knowledge of appropriate theory, practices and tools for the specification, design, implementation.
- PLO-C.** Ability to learn and acquire knowledge through online courses available at different MOOC Providers.
- PLO-D.** Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study.
- PLO-E.** Display ethical code of conduct in usage of Internet and Cyber systems.
- PLO-F.** Ability to pursue higher studies of specialization and to take up technical employment.
- PLO-G.** Ability to formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate.
- PLO-H.** Ability to operate, manage, deploy, configure computer network, hardware, software operation of an organization.
- PLO-I.** Ability to present result using different presentation tools.
- PLO-J.** Ability to appreciate emerging technologies and tools.

Programme Learning Outcomes of B. Sc. Computer Science

These outcomes describe what students are expected to know and be able to do by the time of graduation. They relate to the skills, knowledge, and behaviors that students acquire in their graduation through the program.

Programme Learning Outcomes for B.Sc. Computer Science

The Bachelor of Science with Computer Science (B.Sc. with CS) program enables students to attain, by the time of graduation:

- PLO-1.** Demonstrate the aptitude of Computer Programming and Computer based problem solving skills. **(Disciplinary Knowledge)**
- PLO-2.** Display the knowledge of appropriate theory, practices and tools for the specification, design, and implementation. **(Communication Skills)**
- PLO-3.** Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study. **(Critical Thinking)**
- PLO-4.** Display ethical code of conduct in usage of Internet and Cyber systems. **(Moral and Ethical Reasoning)**
- PLO-5.** Ability to pursue higher studies of specialization and to take up technical employment. **(Reflective Thinking)**
- PLO-6.** Ability to formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate. **(Problem Solving)**
- PLO-7.** Ability to operate, manage, deploy, configure computer network, hardware, software operation of an organization. **(Scientific Reasoning)**
- PLO-8.** Ability to present result using different presentation tools. **(Information Literacy)**
- PLO-9.** Ability to appreciate emerging technologies and tools. **(Analytical Reasoning)**
- PLO-10.** Ability to work independently, identify appropriate resources required for a project and manage a project through to completion. **(Self-Directed Learning)**
- PLO-11.** Ability to work effectively and respectfully with diverse teams; facilitate cooperative or coordinate effort on the part of a group, and act together as a group and work efficiently as a member of a team. **(Team Work)**

PLO-12. Meet one's own learning needs, drawing on a range of current research and development work and professional materials. **(Research Related Skills)**

PLO-13. Ability to acquire knowledge and skills, including “learning how to learn”, that are necessary for participating in learning activities throughout life. **(Life Long learning)**

6. Course Structure of B.Sc. Computer Science

UG - SCHEME OF EXAMINATIONS: CBCS PATTERN

(For the students admitted during the academic year 2021-2022 and onwards)

Part	Sub Code	Title of the Paper	Hrs. (wk)	Internal (CA) Marks	External Marks	Total Marks	Ext- Min.	Total Pass Mark	Credits
Semester – 1									
I	21TAM11L	Part–I: Language: Tamil I	6	50	50	100	20	40	3
II	21ENG12L	Part–II: English I	6	50	50	100	20	40	3
III	21BCS13C	Core 1 : Programming Methodology	4	50	50	100	20	40	2
III	21BCS14C	Core 2 : Digital Computer Fundamentals	3	50	50	100	20	40	3
III	21BCS15P	Core Practical 1 : Programming Methodology Lab	3	50	50	100	20	40	2
III	21BCS16A	Allied – 1: Statistics and Numerical Methods	6	50	50	100	20	40	3
IV	21ENV1GE	Environmental Studies	2	50	50	100	20	40	2
			30			700			18
Semester – II									
I	21TAM21L	Part–I: Tamil II	6	50	50	100	20	40	3
II	21ENG22L	Part–II: English II	6	50	50	100	20	40	3
III	21BCS23C	Core 3: C++ Programming	4	50	50	100	20	40	2
III	21BCS24C	Core 4: Computer System Architecture	4	50	50	100	20	40	3
III	21BCS25P	Core Practical 2: C++ Programming Lab	2	50	50	100	20	40	2
III	21BCS26A	Allied – 2: Discrete Mathematics for Computer Science	6	50	50	100	20	40	3
IV	21VAL2GE	Value Education– Gandhian Thoughts	2	50	50	100	20	40	2
			30			700			18

Part	Sub Code	Title of the Paper	Hrs (wk)	Internal (CA) Marks	External Marks	Total Marks	Ext – Min.	Total Pass Mark	Credits
Semester – III									
III	21BCS31C	Core 5 : Software Engineering	5	50	50	100	20	40	3
III	21BCS32C	Core 6 : Data Structures	5	50	50	100	20	40	3
III	21BCS33C	Core 7 : Programming in JAVA	5	50	50	100	20	40	3
III	21BCS34P	Core Practical 3:JAVA Programming Lab	3	50	50	100	20	40	2
III	21BCS35P	Core Practical 4 : Assembly Language Programming Lab	2	50	50	100	20	40	2
III	21BCS36A	Allied – 3 : Operations Research for Computer Science	6	50	50	100	20	40	3
IV	21BCS37S	Skill Based Subject – I: Micro Processor & Assembly Language Programming	4	50	50	100	20	40	3
			30			700			19
Semester – IV									
III	21BCS41C	Core 8 : Algorithms	5	50	50	100	20	40	4
III	21BCS42C	Core 9 : Database Management System	5	50	50	100	20	40	4
III	21BCS43C	Core 10 : Python Programming	5	50	50	100	20	40	4
III	21BCS44P	Core Practical 5 : Python Programming Lab	3	50	50	100	20	40	3
III	21BCS45P	Core Practical 6 : DBMS Lab (SQL)	2	50	50	100	20	40	2
III	21BCS46A	Allied – 4 : Business Accounting	6	50	50	100	20	40	5
IV	21BCS47S	Skill Based Subject – II: Data Analytics with R	4	50	50	100	20	40	3
V	21EXA4GE	@Extension Activities: NCC/NSS/SPORTS//YRC	-	-	-	-	-	-	1
			30			700			26

Part	Sub Code	Title of the Paper	Hrs (wk)	Internal (CA) Marks	External Marks	Total Marks	Ext- Min.	Total Pass Mark	Credits
Semester – V									
III	21BCS51C	Core 11: Operating System	6	50	50	100	20	40	5
III	21BCS52C	Core 12 : Computer Networks	6	50	50	100	20	40	5
III	21BCS53C	Core 13: Internet Technologies	6	50	50	100	20	40	5
III	21BCS54P	Core Practical 7 : Internet Technologies Lab	3	50	50	100	20	40	2
III	21BCS55P	Core Practical 8 : Linux Shell Programming Lab	2	50	50	100	20	40	2
IV	21BCS56S	Skill Based Subject – III: Computer Graphics	4	50	50	100	20	40	3
IV	21BCS5EL	Non-Major Elective Paper – I: Office Automation Tools	3	50	50	100	20	40	2
V	21INV5IV	@ Industrial Visit****							1
			30			700			25
Semester – VI									
III	21BCS61C	Core 14 : C # Programming	6	50	50	100	20	40	6
III	21BCS62C	Core 15 : Artificial Intelligence and Machine Learning	6	50	50	100	20	40	6
III	21BCS63P	Core Practical 9 : C# Programming Lab	3	50	50	100	20	40	3
III	21BCS64P	Core Practical 10 : Open Source Computing Lab	2	50	50	100	20	40	2
III	21BCS65V	Project & Viva – Voce	5	50	50	100	20	40	10
IV	21BCS66S	Skill Based Subject – IV: Open Source Computing	5	50	50	100	20	40	5
IV	21BCS6EL	Non-Major Elective Paper – II: Introduction to Web Design	3	50	50	100	20	40	2
			30			700			34
		Total/Credits				4200			140

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	PROGRAMMING METHODOLOGY	I	21BCS13C

Course Level Outcomes

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

1. Develop simple algorithms and flow charts to solve a problem.
2. Develop problem solving skills coupled with top down design principles.
3. Write efficient and well-structured computer algorithms/programs.
4. Formulate iterative solutions to a problem.
5. Process algorithms coupled with iterative methods.
6. Write text and string of processing efficient algorithms.
7. Interpret searching techniques and use of pointers.
8. Perform recursive techniques in programming.

UNIT I

Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C Programming - Basic Program Structure In C, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II

Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments, Call-By-Value and Call-By-Reference Parameters, Recursion.

UNIT III

Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT IV

Declaration and Initialization-Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT V

Searching Algorithms - Linear Search, Binary Search. Use of files for data input and output. Merging and copying files.

TEXT BOOKS

1. J. R. Hanly and E. B. Koffman ,”Problem Solving and Program Design in C”, Pearson, 2015.

REFERENCES

1. PradipDey, ManasGhosh, "Programming in C", Second Edition, Oxford Higher Education, 2011.

FURTHER READING

1. <https://nptel.ac.in/courses/106105171>

Semester I		Course Code 21BCS13C		Title of the Paper PROGRAMMING METHODOLOGY				Hours 4	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6	CLO7	CLO8
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓	✓
	Communication Skills		✓	✓	✓	✓	✓	✓	
	Critical Thinking				✓	✓	✓		✓
	Reflective Thinking	✓		✓	✓	✓	✓	✓	✓
	Problem Solving	✓		✓	✓	✓	✓		
	Scientific Reasoning							✓	✓
	Team Work	✓	✓		✓	✓			✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	DIGITAL COMPUTER FUNDAMENTALS	I	21BCS14C

Course Level Outcomes

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

1. Outline basics of Number system and Codes.
2. Discuss concepts of Logic Gates and Boolean Algebra
3. Interpret number of Gates using Karnaugh map Minimization
4. Design Combinational Circuits
5. Design procedures for Sequential Circuits

UNIT I

DIGITAL SYSTEMS AND BINARY NUMBERS: Digital Systems - Binary Numbers- Number- Base Conversions - Octal and Hexadecimal Numbers - Complements of Numbers - Signed Binary Numbers - Binary Codes - Binary Storage and Registers.

UNIT II

BOOLEAN ALGEBRA AND LOGIC GATES - Introduction - Basic Definitions - Axiomatic Definition of Boolean Algebra - Basic Theorems and Properties of Boolean Algebra - Boolean Functions - Canonical and Standard Forms - Other Logic Operations.

UNIT III

GATE-LEVEL MINIMIZATION - Introduction - The Map Method - Four- Variable K-Map - Product- of- Sums Simplification – Don't- Care Conditions.

UNIT IV

COMBINATIONAL LOGIC - Introduction - Combinational Circuits - Analysis Procedure - Design Procedure - Binary Adder/Subtractor - Decimal Adder - Decoders - Encoders - Multiplexers - Demultiplexers.

UNIT V

SYNCHRONOUS SEQUENTIAL LOGIC - Introduction - Sequential Circuits - Storage Elements: Latches - Storage Elements: Flip- Flops – RS, JK, T, D Flip-flops

REGISTERS AND COUNTERS - Registers - Shift Registers - Ripple Counters - Synchronous Counters - Ring Counters – Johnson Counters.

TEXT BOOKS

1. Morris Mano M, Michael DCiletti “Digital Design”, Sixth Edition, Pearson, 2018.

REFERENCES

1. Donald P Leech, Albert Paul Malvino and GoutamSaha, “Digital Principles and Applications”, Tata McGraw Hill,2007.
2. Salivahanan S., Suresh Kumar N, Vallavaraj A., “Electronic Devices and Circuits”, Tata McGraw Hill Co. Ltd.,2008.
3. Mehta V. K., “Principles of Electronics”, S Chand and Company Ltd.,2001.

FURTHER READING

1. Bartee, “Digital Computer Fundamentals”, 6th Edition, Publisher Tata McGraw-Hill Education, 1985.
2. <https://nptel.ac.in/courses/106103180>
3. <https://nptel.ac.in/courses/106108099>

Semester I		Course Code 21BCS14C	Title of the Paper DIGITAL COMPUTER FUNDAMENTALS		Hours 3	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills		✓	✓		✓
	Critical Thinking			✓		
	Reflective Thinking				✓	✓
	Problem Solving			✓		✓
	Analytical Reasoning		✓		✓	✓
	Team Work	✓	✓			✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	PROGRAMMING METHODOLOGY LAB	I	21BCS15P

Course Level Outcomes

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

1. Develop problem solving skills coupled with top down design principles.
2. Design the strategies of writing efficient and well-structured computer algorithms/programs.
3. Develop and experiment the skills for formulating iterative solutions to a problem.
4. Experiment array processing algorithms coupled with iterative methods.
5. Experiment text and string processing efficient algorithms.

LIST OF PROGRAMS:

1. Write a program to find the sum, average, standard deviation for a given set of numbers.
2. Write a program for the following String operations without using built-in functions
 - i. String Copy
 - ii. String Compare
 - iii. String Length
 - iv. String reverse
 - v. String Concatenate
 - vi. Palindrome Checking
3. Write a program to generate 'n' prime numbers.
4. Write a program for Matrix addition, Subtraction and Multiplication.
5. Write a program to read a positive integer and reverse it.
6. Write a program to check whether the given number is an Armstrong number or not.
7. Write a program to print the student's Mark sheet assuming Regno, Name, and Marks in five subjects in a structure. Create an array of structures and print the mark sheet in the university pattern.
8. Write a program that take a paragraph and a number (n) as input and word wraps the text to n characters for each line. Your program must ensure that break the word in half. If the word does not fit within the same line, the word should go to the next line.
9. Write a program to perform all manipulations like insertion, deletion and modification in files.
10. Write a program which takes a file as command line argument and copy it into another file. At the end of the second file write i) Number of Characters ii) Number of Words and iii) Number of Lines.
11. Write a program that will print all the reverse the words of a given sentence in-place. For example, if the following is the input **"This is a line of text"** the output will be **"sihT si a enil fo txet."**
12. A factorion is a natural number that equals the sum of the factorials of its decimal digits. For example, 145 is a factorion because $1! + 4! + 5! = 1 + 24 + 120 = 145$. Write a program that will print all the factorions below 1000000.
13. Write a program that encrypts a given string using mirror characters in the alphabet that is $a \leftrightarrow z$,

b↔y, c↔x and so on.

14. Write a program that will print all the palindrome words in a given sentence.

15. Abundant Numbers - A number n is said to be an abundant number if the sum of its proper divisors is greater than the number n. For example, 12 is an abundant number. $1+2+3+4+6=16$. Write a program that will print all the abundant numbers below 1,000,000.

Semester I		Course Code 21BCS15P	Title of the Paper PROGRAMMING METHODOLOGY LAB		Hours 3	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓		✓	✓
	Critical Thinking		✓	✓		✓
	Reflective Thinking		✓	✓	✓	
	Problem Solving		✓		✓	✓
	Analytical Reasoning		✓			✓
	Team Work	✓	✓			✓

AECC-1 @ SEMESTER I

YEAR	SEM.	SUBJECT CODE	TITLE OF THE PAPER	HOURS/ WEEK
2021 -2022 onwards	I	21ENV1GE	ENVIRONMENTAL STUDIES (For all UG courses)	2

COURSE LEARNING OUTCOMES:

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

1. Recognize the role of the environment and the need to conserve it for sustaining life.
2. Enumerate the natural resources
3. Explores the adverse effects of deforestation and over exploitation of natural resources
4. Associate the components of the ecosystem and need for biodiversity conservation.
5. Evaluate the environmental pollution hazards and their effects on the living system.
6. Interpret the different disaster management procedures.
7. Analyse the climatic change and global effects
8. Infer the need for environmental laws in the constitution of India.
9. Relate the growth of the human population and its impact on the environment.

UNIT I

Environment – Introduction – Nature - Scope – Content – Need for study. Natural resources- Forest and energy resources- Use and overexploitation - deforestation. Energy resources- renewable and non-renewable energy resources.

UNIT II

Ecosystem – concept – types- Forest, Grassland, Desert and Aquatic (Pond)- Structure and function of an ecosystem – Producers- consumers and decomposers – Food chain – food web- ecological pyramids- energy flow. Biodiversity and its conservation- *in situ* and *ex situ* conservation- Mega biodiversity centres and hotspots.

UNIT III

Environmental pollution- definition- causes-effects and control measures of air, water, soil, thermal and nuclear pollution. Waste management- Industrial and solid waste. Disaster management – earthquake, cyclone, flood and landslides.

UNIT IV

Social Issues and the environment-Urbanization-Urban problems related to energy and watershed management. Environmental Ethics- Issues and possible solutions- Wasteland reclamation- Climate change - causes and effects. Global warming- Acid rain- Ozone layer depletion- Public awareness. Environmental laws- Environment Protection Act, Wildlife Protection Act, Forest Conservation Act.

UNIT V

Human population and its impact on environment- Population growth- Resettlement and Rehabilitation of project affected persons- Case studies – Sardar Sarovar Project, Maharashtra and Bandipur National Park- Project Tiger, Karnataka, NTPC, India. Role of Indian and Global religions and Cultures in environmental conservation- Case study: sacred groves in Western Ghats (kavu) & Chinese culture. Human and Wildlife Conflict.

PEDAGOGY STRATEGIES

- Board and Chalk lectures
- PowerPoint slide presentations
- Assignments

TEXTBOOKS

1. Sharma, P. D. 2000. Ecology & Environment. Rastogi Publications, Meerut, India.
2. Bharucha, E. 2003. Text book of Environmental Studies. UGC, New Delhi & Bharati Vidyapeeth Institute of Environmental Education and Research, Pune.
3. Arumugam, M. and Kumaresan, V. 2016. Environmental Studies (Tamil version). Saras Publications, Nagercoil.

Online/E-Resources:

1. <https://www.edx.org/course/subject/environmental-studies>
https://www.coursera.org/courses?_facet_changed=true&domains=life-sciences%2Cphysical-science-and-engineering%2Csocial-sciences&query=environmental%20science%20and%20sustainability&userQuery=environmental%20science%20and%20sustainability
<https://www.open.edu/openlearn/nature-environment/free-courses>

COURSE LEVEL MAPPING OF PROGRAMME LEVEL OUTCOME:

Program Level Outcomes (PLO)	Course Level Outcome (CLO)								
	1	2	3	4	5	6	7	8	9
Disciplinary Knowledge		√	√		√	√			
Communication Skills		√		√				√	√
Critical Thinking	√		√		√		√		
Research related skills	√		√			√		√	
Analytical reasoning	√			√		√		√	
Problem Solving		√	√			√			√
Team Work				√	√		√		√
Moral and ethical awareness		√		√		√		√	√

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	C++ PROGRAMMING	II	21BCS23C

Course Level Outcomes

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

1. Develop the concepts of data, abstraction and encapsulation.
2. Write programs using classes and objects.
3. Recall the principles of Inheritance and Polymorphism and their use and program level implementation.
4. Perform exception and basic exception handling mechanisms in a program
5. Write typical object-oriented constructs of specific object-oriented programming language.

UNIT I

Basics-Introduction to Object Oriented Programming and its Basic Features, Basic Components of C++, Characteristics of Object-Oriented Language, Structure of a C++ Program, Differences Between Object Oriented and Procedure Oriented Programming. Flow Control Statements in C++, Functions - Scope of Variables, Inline Functions, Recursive Functions, Arrays.

UNIT II

C++ pointers, Pointers to functions, Dynamic memory allocation and Deallocation, Overview of Object-Oriented programming Principles: Encapsulation, Data abstraction, C++ Classes, Objects, User defined Types, Constructors and Destructors, 'this' pointer, Friend functions, Overloading, Type Conversion.

UNIT III

Class Inheritance, Base and Derived Classes, Virtual Base Class, Virtual Functions, Polymorphism, Static and Dynamic Bindings, Base and Derived Class Virtual Functions, Dynamic Binding through Virtual Functions, Pure Virtual Functions, Abstract Classes, Virtual Destructors.

UNIT IV

Exception Handling- Benefits of Exception Handling, Throwing an Exception, the Try Block, Catching an Exception, Exception Objects, Exception Specifications, Rethrowing an Exception, Uncaught Exceptions.

UNIT V

Stream Classes Hierarchy, Stream I/O, File Streams, Overloading the Extraction and Insertion Operators, Error Handling during File Operations, Formatted I/O.

TEXT BOOKS

1. Robert Lafore," Object Oriented Programming in C++", 4th Edition, Pearsons Publishes, 2011.
2. Walter Savitch,"Problem solving with C++: The Object of Programming", 4th Edition, Pearson Education.
3. Herbert Schildt," C++: The Complete Reference", 4thEdition.

REFERENCES

1. Sourav Sahay," Object Oriented Programming with C++", 2nd Edition, Oxford University Press.
2. B. Stroutstrup," The C++ Programming Language", 3rd Edition, Pearson Education.
3. Ashok N Kamthane," Programming in C++", Pearson 2ndEdition.

FURTHER READING

1. <https://nptel.ac.in/courses/106105151>

Semester II		Course Code 21BCS23C	Title of the Paper C++ PROGRAMMING		Hours 4	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills		✓	✓	✓	✓
	Critical Thinking		✓	✓	✓	
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving			✓	✓	✓
	Analytical Reasoning					✓
	Team Work		✓			✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	COMPUTER SYSTEM ARCHITECTURE	II	21BCS24C

Course Level Outcomes

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

1. Discuss the basic structure, operation and characteristics of digital Computer
2. Explain arithmetic and logic unit as well as the concept of the concept of pipelining.
3. Interpret hierarchical memory system including cache memories and virtual memory.
4. Demonstrate different ways of communicating with I/O devices and Standard I/O interfaces.
5. Discuss the Parallelism and its Architecture.

UNIT I

Fundamentals of Digital Electronics: Data Types, Complements, Fixed-Point Representation, Floating-Point Representation, Other Binary Codes, Error Detection Codes.

Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus & Memory Transfer, Arithmetic Microoperations, Logic Microoperations, and Shift Microoperation.

UNIT II

Basic Computer Organization: Instruction codes, Computer Registers, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instruction, Input - Output & Interrupts.

UNIT III

Processor and Control Unit: Hardwired vs. Micro programmed Control Unit, General Register Organization, Stack Organization, Instruction Format, Data Transfer & Manipulation.

UNIT IV

Memory and I/O Systems: Peripheral Devices, I/O Interface, Data Transfer Schemes, Program Control, Interrupt, DMA Transfer, I/O Processor. Memory Hierarchy, Processor vs. Memory Speed, High-Speed Memories, Cache Memory, Associative Memory, Virtual Memory.

UNIT V

Parallelism: Instruction-level-parallelism – Parallel processing challenges– Flynn's classification.

TEXT BOOKS

1. M. Morris Mano, "Computer System Architecture", 3rd Edition, Prentice Hall.
2. David A. Patterson and John L.Hennessey," Computer Organization and Design", Fifth edition, Morgan Kauffman/Elsevier, 2014.

REFERENCES

1. John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach", 4th Edition.
2. William Stallings, "Computer Organization and Architecture", Prentice Hall.

FURTHER READING

1. <https://nptel.ac.in/courses/106103180>

Semester II		Course Code 21BCS24C	Title of the Paper COMPUTER SYSTEM ARCHITECTURE		Hours 4	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills		✓	✓		✓
	Critical Thinking				✓	
	Reflective Thinking			✓	✓	✓
	Problem Solving			✓		
	Scientific Reasoning				✓	✓
	Analytical Reasoning				✓	
	Team Work				✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	C++ PROGRAMMING LAB	II	21BCS25P

Course Level Outcomes

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

1. Experiment the concepts of data, abstraction and encapsulation.
2. Write programs using classes and objects.
3. Experiment the principles of Inheritance and Polymorphism and their use and
4. Program level implementation.
5. Perform exception and basic event handling mechanisms in a program.
6. Device object-oriented constructs of C++ object-oriented programming language.

Students are required to understand the object-oriented concepts using C++. They are required to practice the concepts learnt in the theory. Some of the programs to be implemented are listed as follows:

Part A

1. Number of vowels and number of characters in a string.
2. Write a function using call by reference and access the function in main program.
3. Demonstration of array of object.
4. Using this pointer to return a value (return by reference).
5. Demonstration of virtual function.
6. Demonstration of static function.
7. Accessing a particular record in a student's file.
8. Demonstration of operator overloading.

Part B

9. Write a program to create a database for students that contains Name, Enrolment no, Department, Programme using Constructors, destructors, input and output functions; input and output for 10 people using different methods.
10. Create a class holding information of the salaries of all the family members (husband, wife, son, and daughter). Using friend functions give the total salary of the family.

Semester II		Course Code 21BCS25P	Title of the Paper C++ Programming LAB		Hours 2	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving			✓	✓	✓
	Scientific Reasoning					✓
	Analytical Reasoning		✓		✓	✓
	Team Work			✓	✓	

AECC-2 @ SEMESTER II

YEAR	SUBJECT TITLE	SEM.	SUB CODE
2021 -22 Onwards	VALUE EDUCATION – GANDHIAN THOUGHTS (For all UG courses)	II	21VAL2GE

COURSE LEVEL OUTCOMES:

On successful completion of the course, the student will be able to:

1. Interpret Gandhiji's experiments to his spiritual pursuits and search for purity, political activities through fasting protests, and even his role as an educator using diet and meals as teaching exercises.
2. Lead a life marked with humility and truthfulness and subsequent realization of the Truth as the purpose of human life.
3. Infer lessons that are fundamental to living in harmony and social progress such as respect, empathy, equality, solidarity and critical thinking.
4. Promote tolerance and understanding above and beyond our political, cultural and religious differences.
5. Create special emphasis on the defense of human rights, the protection of ethnic minorities
6. Emerge as responsible citizens with clear conviction to practice values and ethics in life.
7. Transform themselves to become good leaders.
8. Realize their role and contribution to the nation building.

UNIT I

Birth and Parentage - Childhood - At the High school - Stealing and Atonement - Glimpses of Religion - Gandhi's choice - Experiments in Dietetics - Acquaintance with Religions - The Great Exhibition.

UNIT II

The first case - Preparing for South Africa - same experiences - on the way to Pretoria – Coolie - Natal Indian Congress - Education of Children - Brahmacharya.

UNIT III

Simple life - The Boer war - Sanitary Reform and Famine Relief - Lord Curzon's Darbar - A month with Gokhale - Experiments in Earth and water treatment - Indian opinion - Coolie Locations or Ghettoes - The Black plague.

UNIT IV

The Magic spell of a Book - The Zulu Rebellion - The Birth of Satyagraha - More experiments in Dietetics - Kasturba's Courage - Domestic Satyagraha- Fasting - Shanti Niketan - Woes of Third-Class passengers.

UNIT V

Kumbhamela - LakshmanJhula - Founding of the Ashram - Abolition of Indentured Emigration - The Kheda Satyagraha - The Rowlatt Bills - Navajivan and young India - Congress Initiation - The Birth of Khadi.

TEXT BOOKS

1. M.K. GANDHI, "The Story of My Experiments with Truth", An Autobiography
Apple publishing International(P) Ltd, Chennai.

. மகாத்மா காந்தியின் சுயசரிதை - சத்தியசோதனை தமிழாக்கம் -

2. -ரா.வேங்கடராமன், நவஜீவன் பரசுராயம், அகமதாபாத

PEDAGOGY STRATEGIES

- Board and Chalk lecture
- Powerpoint slide presentations
- Seminar
- Assignments
- Quizes
- Group discussion

COURSE LEVEL MAPPING OF PROGRAM LEVEL OUTCOMES.

			CourseLevel Outcomes (CLO)							
			1	2	3	4	5	6	7	8
ProgramLevel Outcomes (PLO)	1	Reflective thinking	✓	✓			✓		✓	✓
	2	Communication skills		✓		✓	✓	✓	✓	✓
	3	Critical thinking	✓			✓		✓	✓	✓
	4	Multi-cultural Competence				✓	✓	✓	✓	✓
	5	Analytical reasoning		✓	✓	✓		✓		
	6	Problem solving		✓	✓	✓		✓	✓	✓
	7	Team work	✓		✓		✓	✓	✓	
	8	Leadership readiness/qualities			✓		✓	✓		✓
	9	Moral and ethical awareness	✓		✓		✓	✓		✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	SOFTWARE ENGINEERING	III	21BCS31C

Course Level Outcomes

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

1. Explain the design of complex systems.
2. Apply software engineering principles and techniques.
3. Create efficient, reliable, robust and cost-effective software solutions.
4. Work as an effective member or leader of software engineering teams.
5. Manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals, Identify and analyze the common threats in each domain.

UNIT I

Software Development Approaches: Introduction; Evolving Role of Software; Software Characteristics; Software Applications. **Software Design Processes:** Introduction; What is Meant by Software Engineering? Definitions of Software Engineering; The Serial or Linear Sequential Development Model; Iterative Development Model; The incremental Development Model

UNIT II

Software Design Principles: Introduction, System Models: Data-flow Models, Semantic Data Models, Object Models, Inheritance Models, Object Aggregation, Service Usage Models, Data Dictionaries; Software Design: The Design Process, Design Methods, Design description, Design Strategies, Design Quality; Architectural Design: System Structuring, The Repository Model, The Client–Server Model, The Abstract Machine Model, Control Models, Modular Decomposition, Domain-Specific Architectures.

UNIT III

Object Oriented Design: Introduction; Object Oriented Design: Objects, Object Classes & Inheritance, Object Identification, An Object -Oriented Design Example, Object Aggregation; Service Usage; Object Interface Design: Design Evolution, Function Oriented Design, Data–Flow Design; Structural Decomposition: Detailed Design.

UNIT IV

Software Reliability: Introduction; Software Reliability Metrics; Programming for Reliability: Fault Avoidance, Fault Tolerance, and Software Reuse.

UNIT V

Software Testing Techniques: Introduction; Software Testing Fundamental; Testing Principles; White Box Testing; Control Structure Testing; Black Box Testing; Boundary Value Analysis; Testing GUIs; Testing Documentation and Help Facilities; Software Testing Strategies: Introduction; Organizing for Software Testing; Software Testing Strategy, Unit Testing: Unit Test Considerations, Top-Down Integration, Bottom-Up Integration.

TEXT BOOKS

1. R. S. Pressman, "Software Engineering", 8th edition, Tata McGraw-Hill.

REFERENCES

1. Sommerville, Ian, "Software Engineering", Pearson Education.
2. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa Publications.
3. Pfleeger, Shari Lawrence, "Software Engineering Theory and Practice", Second Edition, Prentice-Hall, 2001.
4. "Object Oriented & Classical Software Engineering", Fifth Edition, SCHACH, Tata McGraw-Hill.

FURTHER READING

1. <https://nptel.ac.in/courses/106101061>

Semester III		Course Code 21BCS31C	Title of the Paper SOFTWARE ENGINEERING		Hours 5	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓		
	Critical Thinking					✓
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving			✓	✓	✓
	Analytical Reasoning			✓		✓
	Team Work				✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	DATA STRUCTURES	III	21BCS32C

Course Level Outcomes

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

1. Recall the fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles.
2. Infer complexity of basic operations like insert, delete, search on these data structures.
3. Choose a data structure to suitably model any data used in computer applications.
4. Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.
5. Implement and know the applications of algorithms for sorting, pattern matching.

UNIT I

Basic concepts - Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction Performance analysis, Linear and Non-Linear data structures, Singly Linked Lists-Operations, Concatenating, circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two-dimensional arrays, sparse matrices-array and linked representations.

UNIT II

Stack- Operations, Array and Linked Implementations, Applications- Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation, Queue- Definition and Operations, Array and Linked Implementations, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue).

UNIT III

Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Priority Queue – Introduction to Heap.

UNIT IV

Graphs, Graph ADT, Graph Representations, Graph Traversals, Searching, Static Hashing-Introduction, Hash tables, Hash functions, Overflow Handling.

UNIT V

Red-Black and Splay Trees, Comparison of Search Trees, Pattern Matching Algorithm-TheKnuth-Morris-Pratt Algorithm, Tries (examples).

TEXT BOOKS

1. E. Horowitz, S. Sahni and Susan Anderson-Freed, "Fundamentals of Data structures in C", 2nd Edition, Universities Press.
2. Lipschutz, "Schaum's outline series Data structures", Tata McGraw-Hill.

REFERENCES

1. M. A. Weiss, "Data structures and Algorithm Analysis in C", 2nd edition, Pearson.
2. Sedgewick and Wayne, "Algorithms", 4th Edition, Addison Wesley, 2011.

FURTHER READING

1. <https://nptel.ac.in/courses/106103069>

Semester III		Course Code 21BCS32C	Title of the Paper DATA STRUCTURES		Hours 5	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Critical Thinking				✓	
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Analytical Reasoning				✓	✓
	Team Work	✓		✓		✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	PROGRAMMING IN JAVA	III	21BCS33C

Course Level Outcomes

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

1. Design the structure and model of the Java programming language.
2. Use the Java programming language for various programming technologies.
3. Develop software in the Java programming language.
4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.
5. Perform file handling techniques in Java.

UNIT I

Introduction: Benefits of OOPS- Java History-Java Features- Java Virtual Machine - Java Environment- Java Tokens- Constants- Variables- Data Types – Operators and Expressions-Decision Making and Branching- Decision Making and Looping.

UNIT II

Classes, Objects and Methods: Classes and Objects - Constructors- Method Overloading- Static Members-Inheritance- Overriding Methods- Final Variables, Final Methods and Final Classes - Finalize Method- Abstract Methods and Abstract Classes –Visibility Control - Arrays - Strings.

UNIT III

Interfaces, Packages and Thread: Defining Interface- Extending Interfaces Implementing Interfaces – Packages.

Multithreaded Programming: Thread Life Cycle - Thread Exceptions – Thread Priority - Synchronization.

UNIT IV

Exception Handling: Types of Errors – Exceptions- Syntax of Exception Handling Code- Multiple Catch Statements- Using Finally Statements.

Applets: Life cycle of an Applet-Development and Execution of a Simple Applet.

Graphics: Introduction

UNIT V

File Handling: Managing Input / Output Files in Java: Concept of Streams -Stream Classes- Character Stream - Classes-Reading / Writing Characters- Reading / Writing Bytes-Handling Primitive Data Types- Random Access files.

TEXT BOOKS

1. E.Balagurusamy, “Programming with JAVA”, 4th edition., Tata McGraw Hill, New Delhi. Units I, II, III and IV.
2. C.Muthu, “Programming with JAVA”, Second Edition, Vijay Nicole Imprints Private Limited, Chennai, 2011. - Unit-V.

REFERENCES

1. PatricNaughton, and Herbert Schildt, "Java - The Complete Reference", Tata McGraw Hill Publishers, 2011.

FURTHER READING

1. <https://nptel.ac.in/courses/106105191>

Semester III		Course Code 21BCS33C	Title of the Paper PROGRAMMING IN JAVA		Hours 5	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning		✓			
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Analytical Reasoning		✓		✓	
	Team Work	✓	✓		✓	

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	JAVA PROGRAMMING LAB	III	21BCS34P

Course Level Outcomes

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

- Explain the basics of JAVA programs and its execution.
 - Interpret concepts like packages and interfaces.
 - Infer life cycle of the applets and its functionality.
 - Demonstrate the usage of util package.
 - Develop java programs using interfaces.
1. Program on strings: Check the equality of two strings, Reverse a string.
 2. Program using loops: to find the sum of digits of a given number
 3. Program to display a multiplication table.
 4. Program to display all prime numbers between 1 to n.
 5. Program to demonstrate all math class functions.
 6. Program on files to copy a file to another file using Java to package classes.
 7. Program to demonstrate method over-riding.
 8. Program to implement method overloading.
 9. Programs to implement single inheritance.
 10. Program to display employee pay slip using inheritance concepts.
 11. Program to implement interface concept.
 12. Program to implement Multi-threaded programming.

Semester III		Course Code 21BCS34P	Title of the Paper JAVA PROGRAMMING LAB		Hours 3	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning			✓		
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Analytical Reasoning		✓		✓	✓
	Team Work		✓	✓	✓	

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	ASSEMBLY LANGUAGE PROGRAMMING LAB	III	21BCS35P

Course Level Outcomes

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

1. Discuss Assembly Language
2. Describe general structure of an assembly language program
3. Infer Data representation in assembly.
4. Explain the instructions of an assembly language program.
5. Interpret the usage of DOS interrupt signals.

List of Experiments:

1. Data transfer instructions like:
 - i. Byte and word data transfer in different addressing modes
 - ii. Block move (with and without overlap)
 - iii. Block interchange
2. Arithmetic & logical operations like:
 - i. Addition and Subtraction of multi precision nos.
 - ii. Multiplication and Division of signed and unsigned Hexadecimal nos.
 - iii. ASCII adjustment instructions
 - iv. Code conversions
 - v. Arithmetic programs to find square cube, LCM, GCD, factorial
3. Bit manipulation instructions like checking:
 - i. Whether given data is positive or negative
 - ii. Whether given data is odd or even
 - iii. Logical 1 and 0
 - iv. Bitwise and nibble wise palindrome
4. Branch/Loop instructions like:
 - i. Arrays: addition/subtraction of nos.
 - ii. Finding largest and smallest nos.
 - iii. Ascending and descending order
 - iv. Near and Far Conditional and Unconditional jumps, Calls and Returns
5. Programs on String manipulation like string transfer, string reversing, searching for a string, etc.
6. Programs involving Software interrupts
7. Programs to use DOS interrupt INT 21h Function calls for Reading a Character from keyboard, Buffered Keyboard input, Display of String on console.

Semester III		Course Code 21BCS35P	Title of the Paper ASSEMBLY LANGUAGE PROGRAMMING LAB		Hours 2	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills		✓	✓	✓	
	Reflective Thinking		✓	✓	✓	✓
	Scientific Reasoning				✓	✓
	Team Work			✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	MICROPROCESSOR & ASSEMBLY LANGUAGE PROGRAMMING	III	21BCS37S

Course Level Outcomes

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

1. Discuss Intel 8085 architecture and its addressing modes.
2. Explain the concepts of 8-bit processors.
3. Demonstrate Assembly language programming concepts.
4. Examine interfaces and interrupts of Intel 8085.
5. Interpret the architecture and functionalities of 8086.

UNIT I

Introduction to Microprocessors – Evolution of Microprocessors- Buses -Microprocessor Architecture: Intel 8085 – ALU - Timing and control unit – Registers - Pin configuration - Instruction cycle: Fetch and Execute operation - Machine cycle and state.

UNIT II

Instruction set of INTEL 8085: Introduction – Instruction and Data formats - Addressing Modes - Status Flags - Intel 8085 Instructions: Data transfer group – Arithmetic group- Logical group - Branch group - Stack, I/O machine control group - Assembly language: stacks – subroutines – MACRO.

UNIT III

Assembly language programming - Addition of two 8-bit numbers - 8-bit subtraction -Decimal addition of two 8-bit numbers - Addition of two 16-bit numbers - 8-bit decimal subtraction - Finding Square from look-up table - Finding largest number in a data array - Arrange a data array in ascending and descending order - Sum of series of 8-bit numbers.

UNIT IV

Peripheral devices and their interfacing: Address space partitioning - Memory and I/O interfacing - Interrupts of Intel 8085 - I/O ports: Programmable Peripheral interface - Programmable DMA controller.

UNIT V

8086 Architecture and assembly language Programming: Basic - 8086 Configuration - minimum mode and maximum mode - CPU Architecture - Internal Operation – Machine language Instructions –instruction Execution timing – Assembler instruction format.

TEXT BOOKS

1. B.Ram, "Fundamentals of Microprocessors and Microcomputers", Dhanpat Rai Publications Pvt.Ltd., 1998.
2. Y.C.Liu and G.A.Gibson, "Microcomputer Systems: The 8086/8088 family Architecture, programming and Design", Prentice Hall of India, New Delhi, 1986.

REFERENCES

1. Ramesh Gaonkar, "*Microprocessor* Architecture, Programming & application with **8085**", Fifth Edition, Penram Publications.
2. Brey, B.B, "The INTEL Microprocessors", Prentice Hall of India Private Limited, 2002.

FURTHER READING

1. <https://nptel.ac.in/courses/106108100>

Semester III		Course Code 21BCS37S	Title of the Paper MICROPROCESSOR & ASSEMBLY LANGUAGE PROGRAMMING		Hours 4	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Problem Solving			✓		
	Scientific Reasoning				✓	✓
	Self-directed Learning	✓		✓	✓	✓
	Team Work			✓		
	Research Related skills		✓			✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	ALGORITHMS	IV	21BCS41C

Course Level Outcomes

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

1. Recall the principles of algorithm design.
2. Analyses algorithms and estimate their worst -case and average- case behavior (in easy cases).
3. To become familiar with fundamental data structures and with the manner in which these Data structures can best be implemented; become accustomed to the description of Algorithms in both functional and procedural styles.
4. Apply their theoretical knowledge in practice.
5. Provide knowledge on advanced Algorithm Analysis Technique.

UNIT I

Introduction: Basic Design and Analysis Techniques of Algorithms, Correctness of Algorithm. Algorithm Design Techniques: Iterative Techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

UNIT II

Sorting and Searching Techniques: Elementary Sorting techniques– Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques- Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques- Medians & Order Statistics, complexity analysis

UNIT III

Graphs Algorithms: Graph Algorithms– Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees. String Processing.

UNIT IV

Lower Bounding Techniques: Decision Trees, Balanced Trees, and Red-Black Trees.

UNIT V

Advanced Analysis Technique: Randomized Algorithm, Distributed Algorithm, Heuristics.

TEXT BOOKS

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein,” Introduction to Algorithms”, 3rd Edition, PHI, 2009.
2. Sara basse& A.V. Gelder,” Computer Algorithm – Introduction to Design and Analysis”, 3rd Edition, Pearson, 1999.

REFERENCES

1. J. Kleinberg and E. Tardos. "Algorithm Design", Pearson, 2006.
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. "The Design and Analysis of Computer Algorithms", Pearson Education (Singapore), 2006.

FURTHER READING

1. <https://nptel.ac.in/courses/106106131>

Semester IV		Course Code 21BCS41C	Title of the Paper ALGORITHMS		Hours 5	Credits 4
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓			
	Communication Skills		✓	✓	✓	✓
	Critical Thinking			✓		
	Moral and Ethical Reasoning		✓	✓		
	Reflective Thinking		✓		✓	✓
	Problem Solving			✓	✓	✓
	Scientific Reasoning		✓			
	Analytical Reasoning		✓			✓
	Team Work	✓	✓		✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	DATABASE MANAGEMENT SYSTEMS	IV	21BCS42C

Course Level Outcomes

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

1. Discuss database systems and database management systems software.
2. Model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.
3. Formulate, using SQL, solutions to a broad range of query and data update problems.
4. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
5. Infer the basics of transaction processing and concurrency control.
6. Explain database storage structures and access techniques.

UNIT I

Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modeling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II

SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views; Altering Schemas.

UNIT III

Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF, 4NF, 5NF;

UNIT IV

Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Query Processing Translation of SQL into Query Plans; Basics of Transactions, Concurrency and Recovery.

UNIT V

DATABASE PROGRAMMING Introduction: Embedded SQL; Dynamic SQL, JDBC; PHP and MySQL, Object Relational Modeling: Hibernate, Active Record in Rails (ruby on Rails) - BIG DATA Introduction; OLAP vs. OLTP; Map Reduce and Hadoop; Spark; Other Systems: HBase.

TEXT BOOKS

1. Elmasri and Navathe, "Fundamentals of Database Systems". 7th Edition, Addison-Wesley, 1992.

REFERENCES

1. Raghu Ramakrishnan, Johannes Gehrke," Database Management Systems", McGraw Hill Education.
2. A. Silberschatz, Henry. F. Korth, S. Sudarshan," Data base System Concepts", McGraw Hill Education.

FURTHER READING

1. <https://nptel.ac.in/courses/106104135>

Semester IV		Course Code 21BCS42C	Title of the Paper DATABASE MANAGEMENT SYSTEMS			Hours 5	Credits 4
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓
	Communication Skills		✓	✓	✓	✓	✓
	Moral and Ethical Reasoning					✓	
	Reflective Thinking		✓	✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓	✓
	Scientific Reasoning		✓				✓
	Analytical Reasoning				✓	✓	✓
	Team Work	✓		✓		✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	PYTHON PROGRAMMING	IV	21BCS43C

Course Level Outcomes

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

1. Develop and Execute Simple Python programs.
2. Structure a Python program into functions.
3. Use Python lists, tuples to represent compound data
4. Develop Python Programs for file processing
5. Provide Knowledge on File Processing and Exceptions

UNIT I

Introduction to Python, Python, Features of Python, Execution of a Python Program, Writing Our First Python Program, Data types in Python, Python Interpreter and Interactive Mode; Values and Types: int, float, Boolean, string, and list; Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators, Comments; Modules and Functions, Function Definition and use, Flow of Execution, Parameters and Arguments.

UNIT II

Operators in Python, Input and Output, Control Statements. Boolean Values and operators, Conditional (if), Alternative (if-else), Chained Conditional (if-elif-else); Iteration: state, while, for, break, continue, pass. Functions: Introduction- Built-in function -Function Composition- User-defined functions- Parameters – function calls- Return Values- Local and Global Scope-Recursion-anonymous function. Illustrative Programs: Square Root, gcd, Exponentiation and factorial of a given no n.

UNIT III

Arrays in Python, Strings and Characters. Strings: String Slices, Immutability, String Functions and Methods, String Module; Lists as Arrays. Illustrative Programs: Sum an Array of Numbers, Linear Search, Binary Search.

UNIT IV

Lists: List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters, Advanced List Processing - List Comprehension. Tuples: Creating tuples-accessing values in tuples-tuples are mutable-Tuple Assignment, Tuple as Return Value-Tuple Operations-Built-in tuple function. Dictionaries: Operations and Methods, Illustrative Programs: Bubble Sort.

UNIT V

Files and Exception: Text Files, Reading and Writing Files, Exceptions: Introduction-exception with arguments-User-Defined Exceptions. Illustrative Programs: Word Count, Copy File.

TEXT BOOKS

1. E. Balagurusamy," Introduction to computing and problem solving using python", McGraw Hill India, 2016.

REFERENCES

1. Mark Lutz," Learning Python".
2. Tony Gaddis, "Starting Out With Python".
3. Kenneth A. Lambert, "Fundamentals of Python".
4. James Payne, "Beginning Python using Python 2.6 and Python 3".

FURTHER READING

1. <https://nptel.ac.in/courses/106106145>

Semester IV		Course Code 21BCS43C	Title of the Paper PYTHON PROGRAMMING		Hours 5	Credits 4
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	
	Critical Thinking		✓			
	Moral and Ethical Reasoning					✓
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	
	Scientific Reasoning					✓
	Analytical Reasoning					✓
	Team Work	✓	✓		✓	✓

YEAR	SUBJECT TITLE	SEM	SUBJECT CODE
2021-2022 onwards	PYTHON PROGRAMMING LAB	IV	21BCS44P

Course Level Outcomes

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

1. Develop and Execute Simple Python programs.
 2. Differentiate types of operations in Python.
 3. Use Python lists, tuples to represent compound data
 4. Demonstrate working with dictionaries in python
 5. Develop Python Programs for file processing
-
1. Write a program to demonstrate different number data types in Python.
 2. Write a program to perform different Arithmetic Operations on numbers in Python.
 3. Write a program to create, concatenate and print a string and accessing substring from a given string.
 4. Write a program to check given string is palindrome or not.
 5. Write a program to create, append, and remove lists in python.
 6. Write a program to arrange a numbers in ascending order.
 7. Find Mean, Median, Mode of a list of numbers.
 8. Write a program to demonstrate working with tuples in python.
 9. Write a program to demonstrate working with dictionaries in python.
 10. Write a python program to find largest of three numbers.
 11. Write a Python program to construct the following pattern, using a nested for loop


```

      *
      * *
      * * *
      * * * *
      * * * * *
      * * * *
      * * *
      * *
      *
```
 12. Write a Python script that prints n prime numbers.

13. Write a python program to define a module to find Fibonacci Numbers and import the Module to another program.
14. Write a program to demonstrate working with text file in python.

Semester IV		Course Code 21BCS44P	Title of the Paper PYTHON PROGRAMMING LAB		Hours 3	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Critical Thinking		✓			✓
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Scientific Reasoning					✓
	Analytical Reasoning					✓
	Team Work	✓			✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	DBMS LAB (SQL)	IV	21BCS45P

Course Level Outcomes

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

1. Explain basic database concepts, applications, data models, schemas and instances.
2. Demonstrate the use of constraints and relational algebra operations.
3. Describe the basics of SQL and construct queries using SQL.
4. Examine the importance of normalization in databases.
5. Design a commercial relational database system (MyAccess/MySQL) by writing SQL using the system.

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing, querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model

Analyze the organization and identify the entities, attributes and relationships in it. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion.

Experiment 4: Normalization

Apply the First, Second and Third Normalization levels on the database designed for the Organization

Experiment 5: Installation of MySQL and practicing DDL commands

Installation of MySQL. Creating databases, how to create tables, altering the database, dropping tables and databases if not required. Try truncate, rename commands etc.

Experiment 6: Practicing DML commands on the Database created for the example Organization

DML commands are used to for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

Experiment 7: Querying

Practice queries (along with sub queries) involving ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Experiment 8 and Experiment 9: Querying (continued...)

Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Experiment 10: Triggers

Work on Triggers. Creation of, insert trigger, delete trigger, update trigger. Practice triggers using the above database.

Note: MyAccess/MySQL may be used.

The following concepts must be introduced to the students:

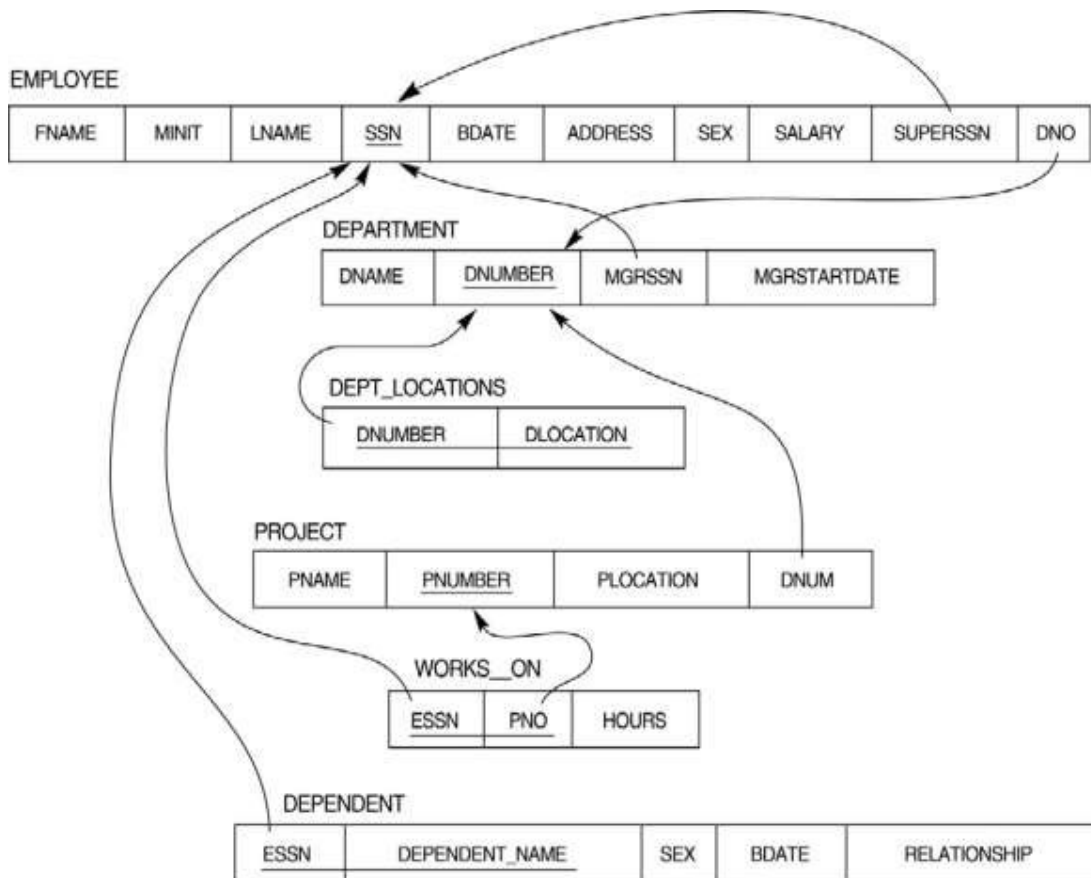
DDL Commands

- Create table, alter table, drop table

DML Commands

- Select , update, delete, insert statements
- Condition specification using Boolean and comparison operators (and, or, not, =, <, >, <=, >=, <=)
- Arithmetic operators and aggregate functions(Count, sum, avg, Min, Max)
- Multiple table queries (join on different and same tables)
- Nested select statements
- Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.)
- Categorization using group by having
- Arranging using order by

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create tables with relevant foreign key constraints.
2. Populate the tables with data.
3. Perform the following queries on the database:
 - a. Display all the details of all employees working in the company.
 - b. Display ssn, lname, fname, address of employees who work in department no7.
 - c. Retrieve the birth date and address of the employee whose name is 'Franklin T. Wong'
 - d. Retrieve the name and salary of every employee.
 - e. Retrieve all distinct salary values.
 - f. Retrieve all employee names whose address is in 'Bellaire'.
 - g. Retrieve all employees who were born during the 1950s.

h. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000 (inclusive)

i. Retrieve the names of all employees who do not have supervisors.

j. Retrieve SSN and department name for all employees.

Semester IV		Course Code 21BCS45P	Title of the Paper DBMS LAB (SQL)		Hours 2	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Critical Thinking		✓	✓		
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Scientific Reasoning		✓	✓		✓
	Analytical Reasoning			✓	✓	
	Team Work	✓		✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 ONWARDS	DATA ANALYTICS WITH R	IV	21BCS47S

Course Level Outcomes

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

1. Discuss the data analytics, Statistical model for data analytics.
2. Explain basics of R Programming language.
3. Demonstrate Data visualization using R.
4. Apply statistical techniques using R Programming for data analytics and decision making.
5. Interpret Reinforcement.

UNIT I

Introduction to Data Analysis: Overview of Data Analytics, Need of Data Analytics, Nature of Data-Classification of Data: Structured, Semi-Structured, Unstructured-Characteristics of Data-Applications of Data Analytics.

UNIT II

R Programming Basics: Overview of R programming, Environment setup with R Studio-R Commands- Variables and Data Types, Control Structures-Array, Matrix, Vectors, Factors, Functions- R packages.

UNIT III

Data Visualization using R: Reading and getting data into R (External Data):Using CSV files, XML files, Web Data, JSON files, Databases, Excel files.

Working with R Charts and Graphs: Histograms, Box plots, Bar Charts, Line Graphs, Scatter plots, Pie Charts.

UNIT IV

Statistics with R: Random Forest, Decision Tree, Normal and Binomial distributions, Time Series Analysis, Linear and Multiple Regression, Logistic Regression, Survival Analysis.

UNIT V

Prescriptive Analytics: Creating data for analytics through designed experiments, creating data for analytics through active learning, Creating data for analytics through reinforcement learning.

TEXT BOOK

1. W. N. Venables, D.M. Smith and the R Development Core Team ,”An Introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics”, Version 3.0.1(2013-05-16). URL: <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>

REFERENCES

1. Jared P Lander, R for everyone, “Advanced analytics and graphics”, Pearson Education, 2013.
2. Dunlop, Dorothy D., and Ajit C. Tamhane,” Statistics and data analysis: from elementary to intermediate”, Prentice Hall, 2000.
3. G Casella and R.L. Berger, “Statistical Inference”, Thomson Learning, 2002.

FURTHER READING

1. <https://nptel.ac.in/courses/110106064>
2. Anthony Fischetti, “Data Analysis with R: Load, wrangle, and analyze your data using the world's most powerful statistical programming language”, PACKT Publishing, 1 January, 2015.

Semester IV		Course Code 21BCS47S	Title of the Paper Data Analytics with R		Hours 4	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Problem Solving			✓	✓	
	Scientific Reasoning	✓	✓			
	Information Literacy	✓			✓	✓
	Analytical Reasoning				✓	✓
	Self-directed Learning	✓		✓	✓	✓
	Team Work	✓			✓	✓
	Research Related skills	✓		✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	OPERATING SYSTEMS	V	21BCS51C

Course Level Outcomes

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

1. Outline operating system, Process concepts and Deadlock concepts.
2. Discuss Real Storage Management.
3. Explain the concepts of Virtual Storage Management and Job and processor scheduling
4. Demonstrate Distributed Computing and disk performance optimization.
5. Infer File and Database systems.

UNIT I

INTRODUCTION AND PROCESS CONCEPTS: what is an Operating System? Definition of process – process states – process state transition – interrupt processing – mutual exclusion – semaphores – deadlock and indefinite postponement.

UNIT II

STORAGE MANAGEMENT: REAL STORAGE – storage organization, management and hierarchy - storage management strategies – contiguous Vs non-contiguous storage allocation – single user contiguous storage allocation – fixed partition multiprogramming – variable partition multiprogramming – multiprogramming with storages wrapping.

UNIT III

VIRTUAL STORAGE MANAGEMENT: Introduction - Virtual storage management strategies – page replacement strategies –working sets – demand paging – page size. **PROCESSOR MANAGEMENT: JOB AND PROCESSOR SCHEDULING:** Scheduling Objectives-Preemptive non-preemptive scheduling – priorities – deadline scheduling – FIFO – RR – SJF – SRT – HRN.

UNIT IV

DISTRIBUTED COMPUTING: Classification of sequential and parallel processing - pipelining –vector processing - array processors – dataflow computers – multiprocessors – fault tolerance. **AUXILIARY STORAGE MANAGEMENT: DISK PERFORMANCE OPTIMIZATION:** Operation of moving head disk storage – need for disk scheduling – seek optimization – FCFS – SSTF – SCAN – RAM disks – optical disks.

UNIT V

FILE AND DATABASE SYSTEMS: File system – functions – organization – allocating and freeing space – file descriptor – access control matrix – backup and recovery – file servers – distributed file system.

TEXT BOOK

1. H.M.Deitel, “Operating Systems”, 2nd Edition, Pearson Education Publ., 2004.

REFERENCES

1. Achyut S Godbole, "Operating Systems", TMH Publ, 2002.
2. Andrew S. Tanenbaum, "Modern Operating Systems", 4th Edition, Pearson Prentice Hall of India, 2014.
3. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, "Operating System Concepts", Ninth Edition, John Wiley and Sons Inc, 2012.

FURTHER READING

1. <https://nptel.ac.in/courses/106106144>

Semester V		Course Code 21BCS51C	Title of the Paper OPERATING SYSTEMS		Hours 6	Credits 5
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning		✓	✓	✓	
	Reflective Thinking	✓	✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓
	Scientific Reasoning	✓		✓	✓	✓
	Analytical Reasoning					✓
	Team Work	✓			✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	COMPUTER NETWORKS	V	21BCS52C

Course Level Outcomes

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

1. Distinguish the structure of Data Communications System and its components and different network terminologies.
2. Demonstrate contemporary issues in network technologies.
3. Discuss the layered model approach explained in OSI and TCP/IP network models
4. Identify different types of network devices and their functions within a network.
5. Explain routing mechanisms, IP addressing scheme and internetworking concepts.
6. Analyze IP and TCP Internet protocols.
7. Outline major concepts involved in design of WAN, LAN and wireless networks.
8. Infer network configuration and maintenance.
9. Examine the fundamentals of network security issues.

UNIT I

Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers

UNIT II

Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques- FDM, TDM, Transmission Media.

UNIT III

Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, Connection- Oriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV

Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols- Stop, Go-Back-N, Point to Point Protocol on Internet.

UNIT V

Multiple Access Protocol and Network Layer: Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols. Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols. Transport Layer and Application Layer Functions and Protocols-Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of WWW & HTTP Protocol.

TEXT BOOKS

1. W. Stallings, “Data and Computer Communications”, 7th Edition, Prentice Hall, 2004.
2. W. Richard Stevens, “TCP/IP Illustrated”, Volume 1, Addison-Wesley.

REFERENCES

1. A. Forouzan, “Data Communications and Networking”, Fourth edition, THM Publishing Company Ltd., 2007.
2. A.S. Tanenbaum, “Computer Networks”, Fourth edition, PHI Pvt. Ltd., 2002.

FURTHER READING

1. <https://nptel.ac.in/courses/106106091>

Semester V		Course Code 21BCS52C		Title of the Paper COMPUTER NETWORKS					Hours 6	Credits 5
Course Level Outcome (CLO)		CL O1	CLO2	CLO3	CLO4	CLO5	CLO6	CLO7	CLO8	CLO9
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Critical Thinking									
	Moral and Ethical Reasoning		✓	✓	✓	✓		✓		✓
	Reflective Thinking	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓	✓	✓	✓	✓
	Scientific Reasoning	✓		✓		✓	✓			✓
	Analytical Reasoning				✓			✓		✓
	Team Work				✓			✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	INTERNET TECHNOLOGIES	V	21BCS53C

Course Level Outcomes

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

1. Create a basic website using HTML and Cascading Style Sheets.
2. Design and implement dynamic web page with validation using JavaScript objects and by applying different event handling mechanisms.
3. Create rich client presentation using AJAX.
4. Design and implement simple web page in PHP, and to present data in XML format.
5. Develop front end web page and connect to the back end databases

UNIT I

Markup Language (HTML5): Basics of Html -Syntax and tags of HTML- Introduction to HTML5 - Semantic/Structural Elements -HTML5 style Guide and Coding Convention- HTML SVG and Canvas – HTML API' s - Audio & Video - Drag/Drop - Local Storage - Web socket API- Debugging and validating HTML.

UNIT II

Cascading Style Sheet (CSS3): The need for CSS – Basic syntax and structure Inline Styles Embedding Style Sheets - Linking External Style Sheets - Introduction to CSS3 – Backgrounds - Manipulating text - Margins and Padding - Positioning using CSS -Responsive Web Design.

UNIT III

Introduction to JavaScript: Core JavaScript - Client-Side JavaScript - Lexical Structure: Character Set - Comments - Literals Identifiers and Reserved Words Optional Semicolons types, values, and variables - Expressions and Operators. Statements: Expression Statements - Compound and Empty Statements - Declaration Statements Conditionals - Loops - Jumps -Objects - Creating Objects - Querying and Setting Properties -Deleting Properties -Testing Properties - Enumerating Properties - Object Attributes - Object Methods Arrays: Creating Arrays - Reading and Writing Array Elements - Adding and Deleting Array Elements - Iterating Arrays

UNIT IV

Functions: Defining Functions -Invoking Functions - Function Arguments and Parameters-Functions As Values - Functions As Namespaces - Closures - Function Properties, Methods, and Constructor - Functional Programming - **Classes and Prototypes:** Classes and Constructors - Java-Style Classes in JavaScript - Augmenting Classes - Classes and Types - Object-Oriented Techniques in JavaScript - Subclasses – Modules- **JavaScript in Web Browsers:** Client-Side JavaScript - Embedding JavaScript in HTML.

UNIT V

Window Object: Timers - Browser Location and Navigation – Browsing History - Browser and Screen Information – Dialog Boxes.

Scripting Documents: Overview of the DOM - Selecting Document Elements - Document Structure and Traversal - Attributes - Element Content - Creating, Inserting, and Deleting Nodes- Generating a Table of Contents.

Scripting CSS: Overview of CSS - Important CSS Properties - Scripting Inline Styles - Querying Computed Styles - Scripting CSS Classes - Scripting Style sheets

TEXT BOOKS

1. Thomas A. Powell, “HTML & CSS: The Complete Reference”, Fifth Edition, 2010
2. David Flanagan, “JavaScript: The Definitive Guide”, Sixth Edition, O'Reilly Media, 2011.

REFERENCES

1. Harvey & Paul Deitel & Associates, Harvey Deitel and Abbey Deitel, “Internet and World Wide Web - How to Program”, Fifth Edition, Pearson Education, 2011.
2. James Lee, Brent Ware, “Open Source Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP”, Addison Wesley, Pearson, 2009.
3. Thomas A Powell, Fritz Schneider, “JavaScript: The Complete Reference”, Third Edition, Tata McGraw Hill, 2013.
4. Thomas A Powell, “Ajax: The Complete Reference”, McGraw Hill, 2008.

FURTHER READING

1. <https://nptel.ac.in/courses/106105084>

Semester V		Course Code 21BCS53C	Title of the Paper INTERNET TECHNOLOGIES		Hours 6	Credits 5
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Critical Thinking					✓
	Moral and Ethical Reasoning	✓	✓		✓	✓
	Reflective Thinking	✓	✓	✓	✓	✓
	Problem Solving	✓	✓	✓		✓
	Scientific Reasoning		✓	✓	✓	
	Analytical Reasoning		✓		✓	✓
	Team Work	✓	✓	✓	✓	✓

YEAR	SUBJECT TITLE	SEM	SUBJECT CODE
2021-2022 onwards	INTERNET TECHNOLOGIES LAB	V	21BCS54P

Course Level Outcomes

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

1. Demonstrate Internet technology with web development using tools like HTML, CSS and JavaScript.
2. Create a web page which includes form design and frame.
3. Display the content in Table.
4. Create a Basic CSS file to style a HTML Document.
5. Develop different types of CSS Files to Design a HTML Document.
6. Discuss the Script Language to develop a webpage.

HTML

1. Write a HTML program for creation of web site with forms, frames, links etc.
2. Design a web site using HTML and DHTML. Use basic text formatting and image tags
3. Create a personal website using HTML and DHTML
4. Write a HTML program to display a traditional newspaper with the use of table tags

CSS 3.0

1. Implement a CSS programs describing layers, inline, internal and external style sheets
2. Develop a webpage using CSS to set the background color, font, and paragraph
3. Develop a webpage using external CSS to import classes for various HTML tags (Use link and import)
4. Develop a webpage in various styles using CSS

Java Script

1. Develop a web page to validate the registration, user login, user profile and payment by credit card pages using JavaScript by importing a .js file
2. Develop a web page to count the number of words and number of vowels in a passage
3. Develop a web page to display a digital clock at the status bar using JavaScript
4. Develop a tool tip text (for form validation) for a webpage

Semester V		Course Code 21BCS54P	Title of the Paper INTERNET TECHNOLOGIES LAB			Hours 3	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓	✓
	Critical Thinking						✓
	Moral and Ethical Reasoning	✓			✓	✓	✓
	Reflective Thinking	✓	✓	✓	✓	✓	✓
	Problem Solving	✓	✓	✓	✓	✓	
	Scientific Reasoning						✓
	Analytical Reasoning	✓				✓	✓
	Team Work	✓	✓			✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 ONWARDS	LINUX SHELL PROGRAMMING LAB	V	21BCS55P

Course Level Outcomes

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

Describe the basics of Linux Operating System

1. Create programs in the Linux environment using Linux utilities and commands.
2. Demonstrate Linux shell commands to write own shell scripts.
3. Develop applications using shell programming.
4. Develop Linux utilities to perform File processing, Directory handling and User Management
5. Develop shell scripts using pipes, redirection, filters, Pipes and display system configuration
6. Develop simple shell scripts applicable to file access permission network administration
7. Apply and change the ownership and file permissions using advance Unix commands.
8. Create shell scripts for real time applications.

List of Programs

1. Write a shell script to stimulate the file commands: rm, cp, cat, mv, cmp, wc, split, diff.
2. Write a shell script to generate factorial of n numbers.
3. Write a Shell Script to implement the following: pipes, Redirection and tee commands.
4. Write a shell script for displaying current date, user name, file listing and directories by getting user choice.
5. Write a shell script to implement the filter commands.
6. Write a shell script to remove the files which has file size as zero bytes.
7. Write a shell script to find the sum of the individual digits of a given number.
8. Write a shell script to find the greatest among the given set of numbers using command line arguments.
9. Write a shell script for palindrome checking.
10. Write a shell script to print the multiplication table of the given argument using for loop
11. Write a shell script for Sorting numbers/Names
12. Write a Shell script to find out whether the given number is odd or even.

Semester V		Course Code 21BCS55P		Title of the Paper LINUX SHELL PROGRAMMING LAB					Hours 2	Credits 2
Course Level Outcome (CLO)		CLO 1	CLO 2	CLO 3	CLO4	CLO5	CLO6	CLO7	CLO8	CLO9
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Communication Skills		✓	✓	✓	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning			✓	✓	✓	✓	✓		✓
	Reflective Thinking		✓	✓	✓	✓	✓		✓	✓
	Problem Solving		✓	✓	✓	✓	✓	✓	✓	✓
	Scientific Reasoning			✓	✓	✓		✓	✓	✓
	Analytical Reasoning							✓		✓
	Team Work	✓	✓				✓	✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 onwards	COMPUTER GRAPHICS	V	21BCS56S

Course Level Outcomes

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

1. Recall the concepts and relevant mathematics of computer graphics.
2. Use various algorithms to scan, convert the basic geometrical Primitives, transformations, area filling, clipping.
3. Describe the importance of viewing and projections.
4. Design basic graphics application programs.
5. Demonstrate fundamentals of animation and Virtual reality technologies
6. Design applications that display graphic images to given specifications.
7. Create a typical graphics pipeline.

UNIT I

Introduction to computer graphics: Brief Survey of Computer Graphics – Graphics Systems: Video Display Devices – Types –Raster-Scan Systems and Random-Scan Systems – Input Devices –Hard-Copy Devices – Graphics Software.

UNIT II

Output primitives and their attributes Line-Drawing (DDA and Bresenham's) Algorithms – Circle-Generating (Midpoint) Algorithm –Area-Filling (Boundary-Fill and Flood-Fill) Algorithms - Line Attributes - Color and Grey scale Levels – Character Attributes.

UNIT III

Two-dimensional transformations and viewing: Basic Transformations - Matrix Representations and Homogeneous Coordinates – Composite Transformations - Other Transformations –Window-to- Viewport Coordinate Transformation.

UNIT IV

Three-Dimensional Transformations: Translation- Rotation- Scaling - Other Transformations. Three-dimensional viewing: Viewing Pipeline and Coordinates - Transformation from World to Viewing Coordinates – Projections – Parallel Projection- Perspective Projection.

UNIT V

Visible Surface Detection Methods: Back-Face Detection, Depth Buffer, Scan line, BSP-Tree Method. Computer Animation: Design of Animation sequence, General Computer Animation Functions key Frame Animation, Animation Sequence, Motion Control methods, Morphing, Warping (only Mesh Warping).

TEXT BOOK

1. D. Hearn and M.P. Baker, "Computer Graphics", 2nd Edition, Pearson Education, Prentice Hall, 2005.

REFERENCES

1. S. Harrington, "Computer Graphics", 2nd Edition, Tata McGraw-Hill Book Co., 1987
2. Foley, Van Dam, Feiner and Hughes, "Computer Graphics Principles & practice", Pearson Education.

FURTHER READING

1. <https://nptel.ac.in/courses/106102065>

Semester V		Course Code 21BCS56S		Title of the Paper COMPUTER GRAPHICS			Hours 4	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6	CLO7
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning					✓		
	Problem Solving		✓	✓	✓	✓	✓	✓
	Information Literacy				✓	✓		
	Analytical Reasoning					✓		
	Self-directed Learning	✓		✓	✓	✓	✓	✓
	Team Work		✓		✓	✓	✓	
	Research Related skills		✓			✓	✓	

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	Non-Major Elective - Paper-I OFFICE AUTOMATION TOOLS	V	21BCS5EL

Course Level Outcomes

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

1. **Save Manpower:** Automation aims at saving manpower. It eases the process of work and thereby reduces work load of employees.
2. **Save Time:** Office automation enables promptness. The work that would take hours can be done in no time with the aid of machines.
3. **Maintain Accuracy:** As the task to be performed is planned well ahead in advance in the form of programmed schedules, automation enables completion of work with accuracy.
4. **Eliminate Monotony:** In case of routine or repetitive job the data which is already feed in the system can be retrieved instead of typing again and again. Repetitive processes may sometimes lead to monotony and office machines eliminate the monotony of these repetitive processes.
5. **Add security:** Automation also minimizes the chances of fraud in office work.
6. **Storage of Facts and Data:** Once the data entered in the computers, can be stored forever. At a time it can store large volume of facts and data for future reference.
7. Enhance the productivity through Automation.
8. Automation eliminates heavy physical labour. The employees have to work less which in turn increases their job satisfaction.
9. Improves the quality of decision making in business through Automation. Computers have become indispensable in certain industries like defense, airlines, insurance, etc.

Introduction to open office/MS office/Libre office

Word Processing: Formatting Text, Pages, Lists, Tables

Spreadsheets: Worksheets, formatting data, creating charts and graphs, using formulas and functions, macros, Pivot Table (6L)

Presentation Tools: Adding and formatting text, pictures, graphic objects, including charts, objects, formatting slides, notes, hand-outs, slide shows, using transitions, animations (4L)

TEXT BOOKS

1. Sushila Madan, Introduction to Essential tools, JBA,2009.
2. Anita Goel, Computer Fundamentals, Pearson, 2012

Practical List for WORD:

1. Create a telephone directory.
 - The heading should be 16-point Arial Font in bold
 - The rest of the document should use 10-point font size
 - Other headings should use 10-point Courier New Font.
 - The footer should show the page number as well as the date last updated.
2. Design a time-table form for your college.
 - The first line should mention the name of the college in 16-point Arial Font and should be bold.
 - The second line should give the course name/teacher's name and the department in 14-point Arial.
 - Leave a gap of 12-points.
 - The rest of the document should use 10-point Times New Roman font.
 - The footer should contain your specifications as the designer and date of creation.
3. Create the following one-page document.
 - a. Compose a note inviting friends to a get-together at your house, including a list of things to bring with them.
 - b. Design a certificate in landscape orientation with a border around the document.
4. Create the following document: A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.
5. Convert following text to a table, using comma as delimiter

Type the following as shown (do not bold).

Color, Style, Item

Blue, A980, Van

Red, X023, Car

Green, YL724, Truck

Name, Age, Sex

Bob, 23, M

Linda, 46, F

Tom, 29, M

6. Prepare a grocery list having four columns (Serial number, the name of the product, quantity and price) for the month of April, 06.

- Font specifications for Title (Grocery List): 14-point Arial font in bold and italics.
- The headings of the columns should be in 12-point and bold.
- The rest of the document should be in 10-point Times New Roman.
- Leave a gap of 12-points after the title.

7. XYZ Publications plans to release a new book designed as per your syllabus. Design the first page of the book as per the given specifications.

- a. The title of the book should appear in bold using 20-point Arial font.
- b. The name of the author and his qualifications should be in the center of the page in 16-point Arial font.
- c. At the bottom of the document should be the name of the publisher and address in 16-point Times New Roman.
- d. The details of the offices of the publisher (only location) should appear in the footer.

8. Create the following one-page document.

- a. Design a Garage Sale sign.
- b. Make a sign outlining your rules for your bedroom at home, using a numbered list.

9. Enter the following data into a table given on the next page.

Salesperson	Dolls	Trucks	Puzzles
Amit	1327	1423	1193
Shivi	1421	3863	2934
Om	5214	3247	5467
Ananya	2190	1278	1928
Anupama	1201	2528	1203
Maharshi	4098	3079	2067

Add a column Region (values: S, N, N, S, S, S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order:

Q1. Create a student worksheet containing roll numbers, names and total marks. Open a document in Word and insert the excel worksheet using: -

- i) Copy/Paste
- ii) Embedding
- iii) Linking

Q2. The term wise marks for APS class of 20 students are stored in 3 separate sheets named term1, term2 and term3. Create 4th worksheet that contains student names and their total and average marks for the entire year. Give proper headings using headers. Make the column headings bold and italic. The 4th worksheet should contain college name as the first line. Make it bold, italic and center it.

Q3. Using a simple pendulum, plot 1-T and 1-T² graph.

I	t1	t2	t3	Mean(t)	$T=t/20$	T ²
70						
80						
90						

Q4. Consider the following employee worksheet

Full Name (First Last)	Grade 1/2/3	Basic Salary	HRA	PF	Gross	Net	(VA) Vehicle Allowance

HRA is calculated as follows:

Grade HRA % (of Basic)

1 40%

2 35%

3 30%

Gross = Basic + HRA + VA

Net = Gross – PF

PF is 8% for all Grades

VA is 15000, 10000 and 7000 for Grades 1, 2 and 3.

- Find max, min and average salary of employees in respective Grade
- Count no. of people where VA > HRA
- Find out most frequently occurring grade.
- Extract records where employee name starts with “A” has HRA > 10000
- Print Grade wise report of all employees with subtotals of net salary and also grand totals. Use subtotal command.
- Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.

Q5. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs. 40. It was resolved to increase the sell of more of more items and getting the profit of Rs40,000/-. Use Goal Seek to find out how many items you will have to sell to meet your profit figure.

Q6. To study the variation in volume with pressure for a sample of an air at constant temperature by plotting a graph for P – V and P-I/V. Sample observations are:-

Pressure(P)	Volume(V)	I/V	PV	P/V
75	20			
78.9	19			
83.3	18			
88.2	17			

Q7. Plot the chart for marks obtained by the students (out of 5) vs. frequency (total number of students in class is 50).

Q8. Create the following worksheet(s) containing a year wise sale figure of five salesmen in

Salesman	2002	2003	2004	2005
MOHAN	10000	12000	20000	50000
MITRA	15000	18000	50000	60000
SHIKHA	20000	22000	70000	70000
ROHIT	30000	30000	100000	80000
MANGLA	40000	45000	125000	90000

Apply the following Mathematical & Statistical functions:

- I. Calculate the commission for each salesman under the condition:
 - a. If total sales are greater than Rs. 3, 00,000/-, then commission is 10% of total sale made by the salesman.
 - b. Otherwise, 4% of total sale.
- II. Calculate the maximum sale made by each salesman.
- III. Calculate the maximum sale made in each year.
- IV. Calculate the minimum sale made by each salesman.
- V. Calculate the minimum sale made in each year.
- VI. Count the no. of sales persons.
- VII. Calculate the cube of sales made by Mohan in the year 2002.
- VIII. Find the difference in sales by salesman Mitra between the year 2002 and 2003. Find the absolute value of difference.
- IX. Also calculate the Mode, Stddev, Variance, Median for the sale made by each salesman.
- X. Calculate the year wise Correlation coefficient between the sales man Mohan and Mitra year wise

Q9. The following table gives a year wise sale figure of five salesmen in Rs.

Salesman	2000	2001	2002	2003
S1	10000	12000	20000	50000
S2	15000	18000	50000	60000
S3	20000	22000	70000	70000
S4	30000	30000	100000	80000
S5	40000	45000	125000	90000

- I. Calculate total sale year wise.
- II. Calculate the net sales made by each salesman
- III. Calculate the commission for each salesman under the condition:
 - a. If total sale is greater than Rs. 4, 00,000/-, then commission is 5% of total sale made by the salesman.
- IV. Otherwise, 2% of total sale.
- V. Calculate the maximum sale made by each salesman.
- VI. Calculate the maximum sale made in each year.
- VII. Draw a bar graph representing the sale made by each salesman.
- VIII. Draw a pie graph representing the sale made by salesmen in year 2001.

Q10. Consider the following worksheet for APS 1st year students: -

S.No.	Name	PH	CH	BY	MT	CS	Total	%	Grade
							Marks		
1									
2									

Grade is calculated as follows: -

If % ≥ 90 Grade A

If % ≥ 80 & < 90 Grade B

If % ≥ 70 & < 80 Grade C

If % ≥ 60 & < 70 Grade D

Otherwise, students will be declared fail.

- i. Calculate Grade using if function
- ii. Sort the data according to total marks
- iii. Apply filter to display the marks of the students having more than 65% marks.
- iv. Draw a pie chart showing % marks scored in each subject by the topper of the class.
- v. Draw the doughnut chart of the data as in (iv)
- vi. Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.
- vii. Extract all records where name
 - a. Begins with "A"
 - b. Contains "A"
 - c. Ends with "A"

Practical List for Power Point:

1. Create five Power point slides. Each slide should support different format. In these slides explain areas of applications of IT. Make slide transition time as 10 seconds.
2. Create five Power Point slides to give advantages/disadvantages of computer, application of computers and logical structure of computer.
3. Create five Power Point slides detailing the process of internal assessment. It should be a self-running demo.

Semester V		Course Code 21BCS5EL			Title of the Paper OFFICE AUTOMATION TOOLS			Hours 3	Credits 2	
Course Level Outcome (CLO)		CLO 1	CLO 2	CLO 3	CLO 4	CLO 5	CLO 6	CLO 7	CLO 8	CLO 9
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Reflective Thinking	✓	✓	✓	✓	✓	✓		✓	
	Problem Solving	✓	✓	✓	✓		✓	✓		✓
	Information literacy						✓		✓	
	Analytical Reasoning			✓						✓
	Self-directed learning	✓	✓	✓	✓		✓	✓		✓
	Team work							✓	✓	✓
	Lifelong learning	✓	✓	✓	✓		✓	✓		✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	C# PROGRAMMING	VI	21BCS61C

Course Level Outcomes

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

1. Explain the structure and model of the programming language C # (note).
2. Use the programming language C # for various programming technologies (understanding).
3. Develop software in C # (application) .
4. Evaluate user requirements for software functionality required to decide whether the programming language C # can meet user requirements (analysis).
5. Propose the use of certain technologies by implementing them in the C # programming language to solve the given problem (synthesis).
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (Evaluation).

UNIT I

Introduction to C# - What is C# - Characteristics of C# - Applications of C# - Differs C# and C++ - Differs C# and java.

Introduction to .net Framework – The .net Strategy – What is .net Framework – What is CLR – Benefits of .net approach.

UNIT II

Overview of C# - Simple C# program - C# Program Structure – C# tokens – Keywords – Identifiers – Literals – Variables – Data Types – Value Types – Punctuators – Boxing and Un boxing.

Operators and Expressions in C# - Type Conversions – Operator Precedence and Associativity - Math Functions in C#.

UNIT III

Branching and Looping – if statement – if-else-statement – Nesting If-else – Else if Ladder – Switch statement - While statement – Do-while statement - for statement – for each Statement.

Methods in C# - Declaring Methods – invoking Methods – nesting of methods – Pass by value – Pass by reference – Methods Overloading.

UNIT IV

Handling Array – Array List – String – String Methods – Inserting String methods- Substring Methods – Regular Expressions.

Class-object-inheritance – polymorphism – Abstract class –Abstract methods.

UNIT V

Creating Console applications – creating windows applications – Form Designing – GUI components.

TEXT BOOKS

1. E. Balagurusamy, "Programming in C# a primer", 3rd Edition, Tata McGraw-Hill, 2010.

REFERENCES

1. R B Wtaker,"C# player's guide".
2. Joel Murach, Anne Boehm, "Murach's C#", 2012.

FURTHER READING

1. Jon Skeet, "C# in Depth", 4th Edition, 2019

Semester VI		Course Code 21BCS61C	Title of the Paper C# PROGRAMMING			Hours 6	Credits 6
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓	✓
	Critical Thinking		✓			✓	
	Moral and Ethical Reasoning			✓		✓	
	Reflective Thinking		✓	✓	✓	✓	✓
	Problem Solving			✓	✓		✓
	Scientific Reasoning		✓	✓		✓	
	Analytical Reasoning			✓			
	Team Work			✓	✓	✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	VI	21BCS62C

Course Level Outcomes

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

1. Discuss the role of basic knowledge representation in AI.
2. Describe the modern view of AI as the study of agents that receive percepts from the Environment and perform actions.
3. Demonstrate awareness of informed search and exploration methods.
4. Deduce the use of NLP applications.
5. Interpret the concept of machine learning and its types.

UNIT I

Hours Fundamentals of Artificial intelligence and Intelligent Agent What is AI? Acting humanly: The Turing Test approach- Thinking humanly: The cognitive modeling approach- Thinking rationally: The “laws of thought” approach-Acting rationally: The rational agent approach- State of Art (Applications of AI) Agents and Environments the Concept of Rationality the Nature of Environment the Structure of Agents. Case Study: Create a new health care market with AI.

UNIT II

Problem Solving by searching Problem-Solving Agents Well defined problem and solutions- Formulating problems- Example Problems Toy Problems-Searching for Solution Uninformed Search Strategies Concept of BFS Concept of DFS- Depth-limited search- Iterative deepening DFS- Bidirectional search- Informed (Heuristic) Search Strategies Concept of Greedy BFS- A* search: Minimizing the total estimated solution cost- Case Study: Applications of AI in transportation.

UNIT III

Natural language processing Language Models N-gram character models- N-gram word models- Text classification by data compression- Information retrieval the page rank algorithm- The HITS algorithm- Information extraction Finite state automata for information extraction- Probabilistic model for information extraction- Examples: Applications of Natural Language Processing. Case Study: Automated Voice Assistants, Chat bots.

UNIT IV

Machine Learning Machine Learning in the bigger picture Areas of machine learning and grades for supervision Supervised Learning strategies - regression versus classification Unsupervised problem solving-clustering.

UNIT V

Types of Machine Learning: Supervised, Unsupervised- Semi-Supervised Learning- Reinforcement Learning- How Supervised Learning works. Why the model works on new data. Case Study: Recommendation Based Systems, At Microsoft, AI is a Big, Big Deal.

TEXT BOOKS

1. Stuart Russel, Peter Norvig,” Artificial Intelligence: A Modern Approach”, Third Edition, Prentice Hall, 2010.
2. Claudio Delrieux, “Machine Learning for Developers”.
3. Andriy Burkov,” The Hundred-Page Machine Learning”.

REFERENCES

1. Rich and Knight,” Artificial Intelligence”, 2nd Edition.
2. Tom M Mitchell, “Machine Learning”.
3. Nils J. Nilsson,” Artificial Intelligence: A New Synthesis”.
4. Artificial Intelligence in the real world, The Economist Intelligence Briefing Paper, Wipro [Case Study]
5. Getting Smarter by the day: How AI is elevating the performance of global companies. [Case Study]

FURTHER READING

1. <https://nptel.ac.in/courses/106105077>

Semester VI		Course Code 21BCS62C	Title of the Paper ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING		Hours 6	Credits 6
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓
	Critical Thinking		✓		✓	
	Moral and Ethical Reasoning			✓	✓	
	Reflective Thinking		✓	✓	✓	✓
	Problem Solving		✓	✓		✓
	Scientific Reasoning			✓	✓	
	Analytical Reasoning		✓	✓		✓
	Team Work				✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	C# PROGRAMMING LAB	VI	21BCS63P

Course Level Outcomes

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

1. Read, write, execute, and debug C# applications
2. Create variables and user defined data types
3. Infer the uses of different decision and control structures.
4. Create user-defined methods
5. Construct and manipulating arrays
6. Apply object-oriented programming techniques.
7. Infer the programming concept/logic using C#
8. Demonstrate the working of numerical concepts in C#.
9. Develop simple application in C#.

C# PROGRAMMING LAB

LIST OF PRACTICALS

1. Write a C# program to check given number is palindrome or not.
2. Write a C# program to check given number is prime or not.
3. Write a C# program to check given number is Armstrong or not.
4. Write a C# console application program to calculate the sum of digits in a given number.
5. Write a C# console application program to print reverse a number.
6. Write a C# console application program to swap of TWO numbers without third variable.
7. Write a C# console application program convert a decimal number to binary number.
8. Write a C# program to convert a decimal number to characters. (Ex. 1 converts to ONE).
9. Write a C# program to print Floyd's triangle using numbers.


```

1
11
1111
11111
111111
1111111

```
10. Write a C# program to print Pascal triangle.

```

A
ABA
ABCBA
ABCDcba
ABCDEDCBA

```

Semester VI		Course Code 21BCS63P			Title of the Paper C# PROGRAMMING LAB				Hours 3	Credits 3
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6	CLO7	CLO8	CLO9
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning			✓			✓		✓	
	Reflective Thinking	✓		✓	✓		✓	✓	✓	
	Problem Solving	✓	✓		✓		✓	✓		✓
	Team Work	✓				✓		✓		✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	OPEN SOURCE COMPUTING LAB	VI	21BCS64P

Course Level Outcomes

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

1. Use open source tools like PHP and MySQL.
2. Create basic programs using PHP Concepts.
3. Apply In-built and create user defined function in PHP programming.
4. Design and develop a web site using form controls for presenting web based content.
5. Use cookies to store and receive data.
6. Create Session Variable and handle session.

List of Experiments:

1. Write a PHP program to validate the Textbox
2. Write a PHP program to draw different shapes
3. Write a PHP program to perform the string manipulation
4. Write a PHP program using MySQL table
5. Write a PHP program to perform user registration form using HTML tags
6. Write a PHP program to display date and time
7. Write a PHP program to check user login
8. Write a PHP program to create a college website
9. Write a PHP program for cookies and session scripts
10. Write a PHP program to perform file read, write, open and append operation.

Semester VI		Course Code 21BCS64P	Title of the Paper OPEN SOURCE COMPUTING LAB			Hours 2	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓
	Communication Skills	✓	✓	✓	✓	✓	✓
	Critical Thinking	✓		✓			
	Moral and Ethical Reasoning				✓	✓	✓
	Reflective Thinking	✓	✓	✓	✓	✓	✓
	Problem Solving		✓	✓	✓	✓	
	Scientific Reasoning	✓		✓		✓	✓
	Analytical Reasoning	✓			✓		✓
	Team Work	✓			✓	✓	✓

Concepts and Installation of MySQL-MySQL structure and syntax -Types of MySQL tables and Storage engines - MySQL commands - Integration of PHP with MySQL - Connection to the MySQL Database - Creating and Deleting MySQL database using PHP - Updating, Inserting, deleting records in the MySQL database - Hosting Website (Using 'C' panel, Using FileZilla Software).

TEXT BOOK

1. VikramVaswani, "PHP: A Beginners guide", Tata McGraw Hill, First edition, 2010.

REFERENCES

1. Matt Doyle, "Beginning PHP 5.3", Wiley India pvt. Ltd, First edition, 2010.
2. Steve Suehring, Tim Converse, Joyce Park, "PHP 6 and MySQL 5", Wiley India pvt. Ltd., First edition, 2010. Anx. UCS (3 B.Sc. Computer Science) 2013Onwards.

FURTHER READING

1. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites", 3 July, 2009.
2. Rasmus Lerdorf, "Programming PHP", 2002.

Semester VI		Course Code 21BCS66S	Title of the Paper OPEN SOURCE COMPUTING			Hours 5	Credits 5
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5	CLO6
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning			✓		✓	
	Problem Solving		✓	✓	✓	✓	✓
	Scientific Reasoning		✓	✓			
	Information Literacy	✓		✓		✓	✓
	Analytical Reasoning				✓	✓	
	Self-directed Learning	✓		✓			
	Team Work		✓	✓	✓	✓	
	Research Related skills					✓	✓

YEAR	SUBJECT TITLE	SEM.	SUBJECT CODE
2021-2022 Onwards	Non-Major Elective - Paper-II INTRODUCTION TO WEB DESIGN	VI	21BCS6EL

Course Level Outcomes

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

1. Infer basics of Web.
2. Demonstrate HTML Structure-tables-frames.
3. Interpret WWW Architecture and protocols.
4. Apply CSS rules to elements on the page (text, images, and hyperlinks) to change their properties such as colour, font type, font size or border colour for example.
5. Discuss PHP Basic syntax for variable types and calculations.

UNIT I

INTRODUCTION: Basics of web- web page – web site – Internet Vs WWW- Domain Names - Do's and Don'ts of creating a web site - Choosing a web host and getting your own website

UNIT II

HTML: Structure of HTML Program – Core Elements – Links and Addressing – Images –Layout Design – Tables – Frames – Form

UNIT III

Internet Standards – Introduction to WWW – WWW Architecture – SMTP – POP3 – File Transfer Protocol - Overview of HTTP, HTTP request – response — Generation of dynamic webpages.

UNIT IV:

CSS: Style sheet Basics – Adding Style to a Document – Creating Style Rules – Fonts – Text Formatting – Padding, Margins & Borders – Colors and Backgrounds – Tables – Element Positioning.

UNIT V:

Introduction of PHP: What is PHP? Advantages of PHP – An overview of PHP – PHP Language structure: the building blocks of PHP –variables-Data types –typecasting – operator and Expressions –Constants.

TEXT BOOKS

1. Bryan Pfaffenberger, Steven M Schafer, Chuck White and Bill Karow, “HTML, XHTML, & CSS Bible”, Wiley Dream tech, New York, 2004.
2. Thomas A Powell, “HTML –The Complete Reference”, Tata McGraw Hill, New Delhi, 2006.

REFERENCES

1. Eric A Meyer, "Cascading Style Sheets: The Definitive Guide", O'Reilly Publications, UK, 2000
2. Dick Oliver, Michael Morrison, "HTML and CSS", Pearson Education, 2006
3. Julie C. Meloni, "PHP, MYSQL and Apache", Dorling Kindersley (India) Pvt Ltd, 2005.

FURTHER READING

1. <https://nptel.ac.in/courses/106106222>
2. C. Xavier, "Web Technology and Design", First edition, NEW AGE, 1 August, 2018.

Semester VI		Course Code 21BCS6EL	Title of the Paper NME: INTRODUCTION TO WEB DESIGN		Hours 3	Credits 2
Course Level Outcome (CLO)		CLO1	CLO2	CLO3	CLO4	CLO5
Programme Learning Outcome (PLO)	Disciplinary Knowledge	✓	✓	✓	✓	✓
	Moral and Ethical Reasoning			✓	✓	
	Reflective Thinking	✓	✓	✓	✓	✓
	Problem Solving			✓	✓	
	Information literacy	✓	✓		✓	✓
	Analytical Reasoning		✓	✓		
	Self-directed learning	✓	✓		✓	✓
	Team work	✓			✓	✓
	Lifelong learning	✓			✓	✓

7. Teaching-Learning Methodologies

The teaching-learning process should be in-line with the course objective and outcomes. Teaching has to ensure that the suggested outcomes are ensured for each course and overall programme. Teaching-aids should be used wherever required to facilitate proper and impactful learning. Blended learning is recommended with the use of MOOC platforms and classroom teaching.

To meet the set objectives of the course and enable students achieve the expected outcomes of the course the teaching-learning process should be appropriately chosen. Though the teachers are best positioned to create innovative models suitable for teaching the course, certain well accepted and widely tested processes are suggested to achieve the desired outcomes

CLASSROOM TEACHING - Regular classroom and face to face teaching and tutorial scan be primarily used for imparting theoretical foundations of Computer Science. Applications of the same may be explained from time to time so that the student can appreciate the theory.

LABORATORY - Lab exercises in programming and usage of package / software tools should be made mandatory and integral part. Opensource software/Packages should be preferred over proprietary tools wherever available.

SEMINARS - Guest lectures and seminars involving industry experts and eminent teachers should be arranged to help the students understand the practices in the industry and developments in the field.

MOOCS - Teacher should choose appropriate lecture materials and videos on similar courses available online through Massive Open Courses Online in the world wide web (such as NPTEL) to provide good perspective of the course and use cases and promote blended learning.

PROJECT - Wherever possible the laboratory assignments can be designed in the form of a mini project. For example, the database course lab assignments can be designed to build a complete system for library management. Similarly, summer/ Semester breaks can be utilized for guiding students to develop live projects with industry orientation/ industry problem. Teamwork work

1. **ASSIGNMENTS** - Home assignments should be designed to make student collect information from various sources and solve unfamiliar problems and make comparisons of solutions.
2. **MAJOR PROJECT** - The major project should be defined based on the student proposals keeping in mind that opportunity to demonstrate the knowledge and skills gained during the course. One-One mentoring support should be provided.
3. **Simulation** - Packages to provide simulated environments to teach various components of networking and hardware working should be used wherever feasible.

8. Assessment Methods

The committee recommends that assessment should be viewed not only merely as a testing by the institution to evaluate the students' progress, but also as a valuable tool for a student to learn what is expected of him/her, where their level of knowledge and skill is lacking, and perhaps most importantly, what he/she could do to improve these levels with the valuable inputs of the lecturers. Assessment methods are the strategies, techniques, tools and instruments for collecting information to determine the extent to which students demonstrate desired learning outcomes. In the Bachelor's programmes leading to degrees such as BSc with Computer Science and BSc in Computer Science, the assessment and evaluation methods focus on testing the conceptual understanding of the basic ideas of computer hardware and software, development of programming skills and experimental techniques, retention and ability to apply the knowledge acquired to real -life applications, and to solve new problems and communicate the results and findings effectively. Based on the Learning Objectives defined for each course as proposed in detail, assessment methods can be designed to monitor the progress in achieving the Learning Objectives during the course and test the level of achievement at the end of the course. Several methods can be used to assess student learning outcomes. Relying on only one method to provide information about the program will only reflect a part of students' achievement.

Modular Assessment

As the courses are broken up into a smaller more cohesive learning outcomes a module will consist of a number of these smaller, finer grained assessments of which the majority can be considered to be formative assessments that aid the learning process rather than assessments aimed at solely being used to evaluate the student.

Continuous Assessment

The continuous assessment occurs on a regular and continuous basis. It is an ongoing formative and summative process, involves monitoring students, is integrated with teaching, involves a systematic collection of marks or grades into a final score, and may be used to determine the students' final grades.

Direct assessment methods ask students to demonstrate their learning, while indirect methods ask students to reflect on their learning. Tests, essays, presentations, etc. are

generally direct methods of assessment, and indirect methods include surveys and interviews. For each Learning Objective, a combination of direct and indirect assessment methods should be used.

Formative Assessment

While *formative assessment* is to gather feedback from formal or informal processes that can be used by the instructor and the students to gather evidence for the purpose of improving learning, *summative assessment* measures the level of success or proficiency that has been obtained at the end of an instructional unit, by comparing it against some standard or benchmark. Nevertheless, the outcome of a *summative assessment* can be used formatively when students or faculty use the results to guide their efforts and activities in subsequent courses. Daily programming assignments or home-assignments is a good way of implementing *formative assessment* and gives an idea of how well the students understood and could apply each programming concept. Another way of *formative assessment* can be that at the end of each class period, a student response system can be used to ask students one or more questions about the topic taught on that day. Regular tutorial Assignment, Term - paper, Seminar Presentation, Surprise Quizzes, Open-book Quizzes should be adopted for formative assessments. It is suggested that 25-30% weightage be given *Formative Assessments* in case of theory components while 30-40% weightage be given to the Programming/Laboratory/Projects/Dissertation components of the various courses.

During the semester, at least three smaller formative assessments shall be given for each course. To pass a course a student had to achieve marks between 70% in two of the assessment opportunities. The philosophy is that the student could fail one opportunity and take the experience gained from that opportunity to pass subsequent assessments.

Summative Assessment

For the traditional summative assessment, it is the semester tests based. The students need to attend two semester tests which consist of half of the content they learned for each test. Students are admitted to an examination for individual courses if they attain the minimum semester mark of 40%. Summative Assessment for the theory papers, can be a combination of Mid -Semester Test, Individual /Team Project report, Oral Presentations of Seminar/Projects, Viva -Voce Examination for dissertation and End Semester closed book

examination. Summative Assessment methods shall be different for theory courses and Practical Courses.

It is suggested that the examination questions should be asked keeping the learning outcomes in mind and also covering all the Units. Term papers, problem solving assignments, Lab, projects, Internship experience, group projects are recommended for achieving the expected outcomes. Wherever possible, students need to do minor projects in practical classes to learn the technology and also to apply the technology for problem solving. As this is a technology-oriented programme and new technologies are introduced quite often, care should be taken to familiarize the students with the recent advances through seminars or term papers and case studies. This should be given due weightage during continuous evaluation process. To achieve this objective, the following are suggested

- i. The end examination papers should be covering all units of the syllabus. Questions should be balanced and evaluate the comprehension, analytical and problem -solving skills.
- ii. The students should be evaluated on teamwork in addition to the technical skills through projects.
- iii. Ability to self-learning and solving new problems should be assessed through assignments, Seminars and project work.
- iv. It is recommended that 25-30% weightage of marks shall be devoted for formative assessment.
- v. It is recommended that 40% weightage be given for practical and laboratory work.
- vi. Peer evaluation component is recommended for project evaluation and seminar.
- vii. Online course certification should be encouraged and equivalent grade for the same need to be worked to achieve the outcome of self-learning.

Government Arts College (Autonomous), Coimbatore -641 018.

B.Sc. Degree examination, Nov/Dec -2021

Computer Science - Semester - I

Programming Methodology (21BCS13C)

Time: 2 Hrs

Max. Marks: 50

PART - A

I. Choose the best answer

(5 * 1 = 5 Marks)

1. Find out the correct order
 - a) char<int<double
 - b) int>char>float
 - c) char>int>float
 - d) double>char>int
2. Which of the following is not a data type
 - a) float
 - b) int
 - c) real
 - d) imaginery
3. Can we pass a variable argument list to a function at run-time?
 - a) True
 - b) False
4. Predict the output,

```
#include <stdio.h>
int main()
{ int i;
  i = 1, 2, 3;
  printf("i = %d\n", i);
  getchar();
  return 0;
}
```

 - a) i=1
 - b) i=2
 - b) i=3
 - d) i=123
5.

```
#include int main() { display(); return 0; } void display() {
printf("IndiaBIX.com"); }
```

 - a) display() is called before it is defined
 - b) display() doesn't get invoked
 - c) none of these
 - d) no errors

II. Answer any three questions

(3 * 2 = 6 Marks)

6. Define an array.
7. What is function?
8. List any eight Library functions.
9. What is the use of return statement
10. Write a C program to covert centigrade to Fahrenheit.

PART – B

(5 * 3 = 15 Marks)

III. Answer ALL questions – (Short Answers not exceeding 100 words)

11. A) Write short notes on actual and formal parameters (OR)
B) Write a C program to find out the maximum of N different numbers.
12. A) Write short notes on function with no arguments and no return data (OR)
B) Write a C program to add two matrices.
13. A) Write short notes on one dimensional array. (OR)
B) Write a C program to arrange the numbers in the arranging order.
14. A) What is string? Briefly write about strings. (OR)
B) What is recursion? Describe with a suitable example.
15. A) Write a C program to convert centigrade to Fahrenheit (OR)
B) List any five library string functions. Explain each with suitable example.

PART - C

(3 * 8 = 24 Marks)

IV. Answer any THREE questions

16. Write a C program to find out whether the string the given string is a palindrome or not.
17. Explain in detail about functions with argument and with return statement.
18. Describe in detail with two dimensional arrays with example.
19. Describe how to pass array arguments.
20. Write a C program to generate n numbers of prime numbers.

***** ALL THE BEST *****