Punyashlok Ahilyadevi Holkar Solapur University, Solapur

NAAC Accredited-2022
'B++' Grade (CGPA 2.96)

Name of the Faculty: Science & Technology

CHOICE BASED CREDIT SYSTEM

Syllabus: Entire Computer Science

Name of the Course: B.Sc. I ECS (Sem.– I & II)

(Syllabus to be implemented from June 2022)

Punyashlok AhilyadeviHolkar Solapur University, Solapur

B. Sc.[ECS] - I year (Entire Computer Science)

Preamble: B. Sc. (Entire Computer Science) is a 3 year undergraduate programme with a specialization in the domain of computer science, software and hardware related aspects. B.Sc. (ECS) programme is perfect for students who want to make a career in computers. Major subjects in this programme include Digital Electronics, Computer Programming Theory, Discrete Mathematics, Advanced Programming using Python, C++, Java, etc. The course curriculum is inclusive of theory and practical which makes the students well trained and skillful in the field of programming, software, and network.

The objective of the Programme:

1. To develop problem solving abilities using a computer.
2. To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.
3. To train students in professional skills related to Software Industry.
4. To prepare the necessary knowledge base for research and development in Computer Science.
5. To help students build-up a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products. Programme Outcome: B.Sc. (ECS) programme has been designed to prepare graduates for attaining the following specific outcomes:
6. An ability to apply knowledge of mathematics, statistics and computer science in practice.
7. An ability to enhance not only a comprehensive understanding of the theory but its application too in diverse fields.
8. The program prepares the young professional for a range of computer applications, computer organization, techniques of Computer Networking, Software Engineering, Web Development, Database management and Advance Java.
9. An ability to design a computing system to meet desired needs within realistic constraints such as safety, security and applicability in multidisciplinary teams with a positive attitude.
10. In order to enhance the programming skills of the young IT professionals, the program has introduced the concept of project development in each language/technology learned during the curriculum.

Eligibility for B.Sc. (ECS) Part-I:

1. The candidate passing the Higher Secondary Examination Conducted by the Maharashtra State Board of Higher Secondary Education, with science stream,MCVC with Science Subjects, D. Pharm, Diploma, Engineering, Agricultural Diploma, Dairy Diploma shall be allowed to enter upon the B. Sc. Part-I Course.
2. An examination of any other statutory University or an Examining Body is recognized as equivalent thereto.
3. Repeater Students will be allowed to take fresh admission to the same Class with the same subjects or different subjects.

Programme Learning Outcomes:

These outcomes describe what students are expected to know and can do by the time of graduation. They relate to the skills, knowledge, and behaviours that students acquire in their graduation through the program Programme Learning Outcomes for BSc(Entire Computer Science):

The Bachelor of Science(Entier Computer Science) programme enables students to attain, by the time of graduation:

PLO-1. Demonstrate the aptitude for Computer Programming and Computer based problem solving skills.

PLO-2. Display the knowledge of appropriate theory, practices and tools for the specification, design, implementation

PLO-3. Ability to learn and acquire knowledge through online courses available at different MOOC Providers.

PLO-4. Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study.

PLO-5. Display ethical code of conduct in the usage of Internet and Cyber systems.

PLO-6. Ability to pursue higher studies of specialization and to take up technical employment.

PLO-7. Ability to formulate, model, design solutions, procedure and use software tools to solve real world problems and evaluate.

PLO-8. Ability to operate, manage, deploy, configure computer network, hardware, and software operation of an organization.

PLO-9. Ability to present results using different presentation tools.

PLO-10. Ability to appreciate emerging technologies and tools.

PLO-11. Apply standard Software Engineering practices and strategies in real-time software project development.

PLO-12. Design and develop computer programs/computer -based systems in the areas related to algorithms, networking, web design, cloud computing, IoT and data analytics.

PLO-13. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems

PLO-14. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.

PLO-15. The ability to work independently on a substantial software project and as an effective team member.

PunyashlokAhilyadeviHolkar Solapur University, Solapur

Faculty of Science and Technology
Choice Based Credit System (CBCS),(w.e.f. 2022-23) Revised Structure for B.Sc.(Entire Computer Science)-I

Subject/ Core Course	Name and Type of the Paper			Hrs./week			Total Marks Per Paper	UA	CA	Credits
	Type	Name	Paper / Practical No	L	T	P				
B.Sc.(Entire Computer Science)-I Sem-I										
Ability Enhancement Compulsory Course (AECC)		English Paper I Part-A (communication-skill)		4.0			50	40	10	2.0
*Core Courses .	DSC 1A	Fundamental of Computer	Paper- I	2.5	--	--	50	40	10	4.0
		Basics of Operating System	Paper-II	2.5	--	--	50	40	10	
	DSC 2A	Programming using 'C'	Paper- I	2.5	--	--	50	40	10	4.0
		Python – I	Paper-II	2.5	--	--	50	40	10	
	**DSC 3A /GE-1A	Numerical Methods	Paper- I	2.5	--	--	50	40	10	4.0
		Graph Theory	Paper-II	2.5	--	--	50	40	10	
	***DSC 4A /GE-2A	Basic Electronics	Paper- I	2.5	--	--	50	40	10	4.0
		Advanced Electronics	Paper-II	2.5	--	--	50	40	10	
		Total		**24**	**--**	**--**	**450**	**360**	**90**	**18**
B.Sc.(Entire Computer Science)-I Sem-II										
Ability Enhancement Compulsory Course (AECC)		English Paper I Part-B (communication skill)		4.0			50	40	10	2.0
*Core Courses	DSC 1B	Introduction to Web Technology	Paper-III	2.5	--	--	50	40	10	4.0
		Operating System	Paper-IV	2.5	--	--	50	40	10	
	DSC 2B	Object Oriented Programming using C++	Paper-III	2.5	--	--	50	40	10	4.0
		Python – II	Paper-IV	2.5	--	--	50	40	10	
	**DSC 3B/GE-1B	Linear Algebra	Paper-III	2.5	--	--	50	40	10	4.0
		Discrete Mathematics	Paper-IV	2.5	--	--	50	40	10	
	***DSC 4B/GE-2B	Digital Electronics and Microprocessor	Paper-III	2.5	--	--	50	40	10	4.0
		Introduction to Microcontroller and Embedded System	Paper-IV	2.5	--	--	50	40	10	
	Democracy, Elections and Good Governance			3			50	40	10	NC
Total (Theory)				**24**	**--**	**--**	**450**	**360**	**90**	**18**
Core Practical	DSC 1 A & 1B	Practical I		--	--	4	100	80	20	4.0
	DSC 2 A & 2B	Practical II		--	--	4	100	80	20	4.0

	DSC 3A &3B / GE-1A&1B	Practical III		--	--	4	100	80	20	4.0
	DSC 4A & 4B / GE-2A &2B	Practical IV		--	--	4	100	80	20	4.0
Total (Pract.)						**16**	**400**	**320**	**80**	**16**
Grand Total				**48**		**16**	**1300**	**1040**	**260**	**52**

*Core Courses: DSC 1A , DSC 1B, DSC 2A and DSC 2B (Core computer science courses)

**Generic Electives: DSC 3A/GE-1A and DSC 3B/GE-1B: Mathematics

***Generic Electives : DSC 4A/GE-2A and DSC 4B/GE-2B: Electronics

Abbreviations:

L: Lectures T: Tutorials P: Practical UA : University Assessment CA : College Assessment CC: Core Course AEC : Ability Enhancement Course DSE : Discipline Specific Elective Paper SEC : Skill EnhancementCourse, AIC: Additional Interdisciplinary Courses GE: Generic Electives

Type: DSC 1A
B.Sc(ECS)-I (Semester I)

Course Title: Fundamental of Computer

(Paper Code:I)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Objectives:

1. The main objective is to introduce IT in a simple language to all undergraduate students, regardless of their specialization.
2. It will help them to pursue specialized programs leading to technical and professional careers and certifications in the IT industry.
3. The focus of the subject is on introducing skills relating to IT basics, computer applications, etc.

Course Outcomes: At the end of this course, the student should be able to

1. To understand basic concepts and terminology of information technology.
2. To a basic understanding of personal computers and their operations.
3. To understand various input and output devices.
4. To understand memory management.

Unit I **18]**

Introduction to computer - Definition of computer, History, characteristics, limitations,concepts of H/W and S/W, classification of computer based on size and purpose, applications of computers in various fields, , computer language – high level, low level, assembly level , compiler, interpreter.

Number System: Introduction of number systems, complements, fixed and floating point representation, character representation, addition, subtraction, conversion Binary, Octal, Decimal, Hexadecimal.

Block diagram – ALU, Memory Unit, Control Unit, Introduction to the motherboard, SMPS, Expansion Slots, Serial and Parallel ports.

Unit II **[12]**

I/O Devices and Concept of Memory:

Input Devices- Input Devices- Keyboard, Mouse, Light pen, Joystick, Trackball,Scanner, Touch screen, MICR (QR code design and reading), OMR, Bar Code reader, Microphone.

Output devices- Monitor-(CRT, LCD, LED), Printers-(Dot Matrix , Ink Jet , Laser, Chain and Drum), Plotters.

Concept of Memory:

Primary Memory (RAM and its types,ROM and its types) and Secondary Storage devices, types of memory based on materials (electronic, optical and magnetic) RAID and its levels.

Reference Books:-

1. Computer Fundaments - P.K. Sinha, BPB Publications, Edition-4th, 2004.
2. Fundamental of computers - V. Raja Raman, PHI Learning; 6th edition, 2014.

Type: DSC1A

B.Sc(ECS)-I (Semester I)

Course Title: Basics of Operating System

(Paper Code:II)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objective:

Students will try to learn:

1. To understand the main components of an OS & their functions.
2. Describe the functions of a modern OS with respect to convenience, efficiency and the ability to evolve.
3. To make aware of different types of OS and their services.
4. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.

Course Objectives:

1. To provide a sound understanding of the Computer operating system, its structures, and its functioning.
2. To understand the services provided by and the design of an operating system.
3. To understand different approaches to memory management.
4. To understand the services provided by and the design of an operating system.
5. To understand what a process is and how processes are synchronized and scheduled.

Unit I **[10]**

Operating system: Definition operating system, Types of Operating Systems-Batch, Multiprogramming, Time Sharing, Real-Time, Distributed, Parallel, OS Services, System components, System Calls.

Process Management: Concept of Process, Process states, Process Control Block, Context switching, Operations on Process.

Unit II **[20]**

Process Synchronization and Deadlocks:

Scheduling- Concept of Process Scheduling, Types of Schedulers, Scheduling criteria, Scheduling algorithms Preemptive and Non-preemptive, FCFS, SJF, Round Robin, Priority Scheduling, Multilevel Queue Scheduling, Multilevel- feedback Queue Scheduling.

Process Synchronization: The Producer Consumer Problem, Race Conditions, Critical Section Problem, Semaphores, and Classical Problems of Synchronization: Reader-Writer Problem, Dinning Philosopher Problem.

Reference Books:

1. Operating System Concepts BySiberchatz and Galvin, Wiley; Edition-8th, 2008.
2. Modern O.S. By Andrews Tanenbaum, Pearson; Edition-4th,2014.

Type: DSC 2A

B.Sc(ECS)-I (Semester I)

Course Title: Programming using 'C'

(Paper Code:I)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5

University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objective:

1. To learn the fundamental programming concepts and methodologies.
2. To learn problem solving techniques using C.
3. To train the student in the basic concepts of the programming language C.
4. To improve the programming skills using C.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. To understand the fundamentals of C programming.
2. To read, Understand, Write and Execute the programs using C.
3. To apply logical thinking toa given program and write the code.
4. To Identify the correct and efficient ways of solving problems.

Unit-I **[15]**

Program: Programming Planning Tools: Algorithm, Flowchart, Pseudocode, Types of Error in Programming.

Introduction to C Language: History of C language/ Evolution of C language, Features / Characteristics of C language, Structure of C Program, C Tokens: Keywords (Reserved Words), Operators, Data Types, Identifier, Variable, Constant, Special Symbols.

Input and Output Operations: Format Code or Format Specifier, Escape Characters (Backslash Character Constants)

Decision Making and Branching: Conditional Operator Statement-if Statement (Two way decision making statement), switch statement: (Multi way decision making statement), goto statement, Iteration logic (Looping Statements)- while loop, do-while loop, for a loop.

Unit-II **[15]**

Array and String: Introduction to Array, Types of an Array- One Dimensional Array, Two Dimensional Array, Multi Dimensional array.

Strings-Introduction to String, Reading and Writing String, The String handling functions.

Function and Pointer: Introduction to Function, Characteristics of Functions, Types of Function, Advantages or Need of Function, Steps to Add User-Defined Function in the Program, Types of Arguments in Function

Types of User-Defined Functions, Recursion, Scope of Variable.

Introduction to Pointer- Advantages of using a pointer, Declaration of Pointer, Initialization of Pointer, Accessing the value of a variable by using its pointer, NULL Pointer, Chain of Pointer (Pointer to Pointer), Pointer Arithmetic, Call by value, Call by Address (Call by pointer or call by reference).

Reference Books:

1. Programming in ANSI C, E. Balagurusamy, McGraw Hill Education India Private Limited;, Edition-7th, 2017.
2. Programming through C Language, Dr. Tulashiram B. Pisal and Mr. Balasaheb J. Kshirsagar, InSc Publishing House(IPH), Edition-1st, 2021.
3. The 'C' programming language, Brian Kernighan, Dennis Ritchie, Pearson, Edition-2nd , 2015.
4. Let Us C - Y.C. Kanetkar, BPB, Edition-17th .

Type.: DSC 2A
B.Sc(ECS)-I (Semester I)

Course Title: Python - I

(Paper Code: Paper-II)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objective:

1. To learn the fundamentals of python Programming
2. To learn different data structures used in Python
3. To learn different control statements used in logic development.
4. To learn the various operations on the array, list, tuple, string, set, and dictionary.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Understand the basic concepts and applications of Python.
2. Design, create, build, and debug python applications.
3. Explore Integrated Development Environment (IDE).
4. Write and apply decision structures for different operations.
5. Write loop structures to perform iterative tasks.

UNIT I **[18]**

Introduction: features of python, steps for execution of python program, python virtual machine, memory management, garbage collection, Installation of python software, setting the path to operating system environment, writing the first python program, executing a python program.

Datatypes in python: Datatypes, type conversion- implicit and explicit, comments, literals, constants, Identifiers, naming conventions, operators, operator precedence and associativity, input and output statements, command-line arguments.

Control Statements: if statement, if..else statement, if..elif..else statement, while loop, for loop, else suite, infinite loop, nested loops, word indentation, break statement, continue statement, pass statement, assert statement, return statement.

UNIT II **[12]**

Arrays in Python: Concept of array, advantages of array, creating an array, importing array module, indexing and slicing on arrays, methods of array module, types of arrays. **String, List, Tuple, Set and Dictionary:** Creating string, manipulating different operations on string, creating list, manipulating different operations on list, list comprehensions, creating tuple, manipulating different operations on tuple, creating set, manipulating different operations on set, creating dictionary, manipulating different operations on dictionary.

Reference Books:

1. Python: The Complete Reference by Martin C. Brown.
2. Core Python Programming, Dreamtech publications, by R. Nageswara Rao.
3. Python Programming, A modular approach, First Edition, Pearson, by Taneja Sheetal
4. Learning with Python, Dreamtech publications, by Allen Downey
5. Python Programming for the Absolute Beginner by Michael Dawson-Cengage Learning.

Type.:DSC 3A
B.Sc(ECS)-I (Semester I)
Course Title: Numerical Methods

(Paper Code: Paper-I)

Credits: Theory – (2)Practicals – (2)

Total Lectures: 30 Hrs **Contact Hrs (L) :**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Objectives:

1. The course is designed to have a grasp of important concepts of Numerical Methods in a scientific way.
2. The learner is expected to solve as many examples as possible to get complete clarity and under standing of the topics covered.

Course Outcomes:

1. Ability to appreciate real world applications which use these concepts.
2. Skill to formulate a problem through Mathematical Modeling and programming.

Unit I **[18]**

Polynomial Interpolation Approximation and Errors:

Interpolation and Extrapolation: Argument and entries, equally spaced data and not equally spaced data, Finite difference operators: forward difference operator, backward difference operator, divided difference operator, Relation between these operators, Interpolation and Extrapolation, Newton's forward difference interpolation formula, Newton's backward difference interpolation formula, Lagrange's interpolation formula, problems, Newton's divided difference interpolation formula (only formula without proof), problems.

Errors: Absolute error, relative error and percentage error

Normalized Floating point representation of real numbers, arithmetic operations on the numbers in normalized floating point notation: addition, subtraction, multiplication and division.

Unit II **[12]**

Numerical Integration and Ordinary Differential Equation:

Numerical Integration General quadrature formula for equidistant ordinates (without proof), Trapezoidal rule, Simpson's 1/3 rd rule and Simpson's 3/8 th rule, derivation of rules and examples

Ordinary Differential Equation: Degree and order of a differential equation, Definition of an ordinary differential equation, Picard's Methods ,Taylor Series Method, Euler's method, Runge-Kutta second order method, Runge-Kutta fourth order method, examples.

Reference Books:

1. Introduction to Numerical Analysis by S. S. Sastri, Tata McGraw Hill
2. Computers and Numerical Methods by Balguruswamy, (TMH).
3. Numerical Methods for Scientific and Engineering Computation
   M. K. Jain, S. R. K. Iyengar and R. K. Jain, New Age International Publisher
4. Applied Numerical Methods with MATLAB for Engineers and Scientists Steven C Chapra, Tata McGraw Hill, 2/e (2010).

Type.:DSC 3A
B.Sc(ECS)-I (Semester I)
Course Title:Graph Theory
(Paper Code: Paper-II)

Credits: Theory –(2) **Practicals– (2)**
Total Lectures: 30 Hrs **Contact Hrs (L) :2.5**
University Evaluation: 40 Marks **Internal Evaluation: 10 Marks**

Course Objectives:

1. Cultivate clear thinking and creative problem solving.
2. To apply graph theory in solving practical problems.

Course Outcomes: On successful completion, of course, the learner/student will be able to:

1. Understand the notion of mathematical thinking, and mathematical proofs and to apply them in problem solving.
2. Ability to understand and apply concepts of graph theory in solving real world problems and ability to reasonlogically.

Unit I **[12]**

Graph and Operations on Graphs:

Introduction, the definition of the graph, basic terminology, types of graphs: simple, multi, pseudo, null, regular, r-regular, complete, bipartite, complete bipartite, weighted graphs, Degree of a vertex, the total degree of a graph, shaking hand lemma and elementary results, Matrix Representation of graphs: Adjacency and incidence matrix.

Operations on Graphs: Union, intersection, ring sum, a product of two graphs, complement of a graph, Graph Coloring.

Unit II **[18]**

Connected Graphs and Trees

Connected Graph: Definition of connected and disconnected graphs, walk, trail, path, length of a path, the distance between two vertices, Dijkstra's shortest path algorithm and problems.

Euler and Hamiltonian Graph: Euler trail, Euler circuit, Euler graph, Hamiltonian Path, Hamiltonian circuit, Hamiltonian Graph, Travelling Salesman Problem, Chinese Postman problem (for both Eulerian and non Eulerian graph), problems.

Tree: Definition of the tree, spanning trees, shortest spanning tree, Kruskal's algorithm for shortest spanning tree, Binary tree: definition of a binary tree, root, internal vertices, intermediate vertices, and elementary results.

Reference Books:

1) Elements of graph theory- Bhave and Raghunathan

2) Discrete Mathematical structure for Computer Science-Alan Doerrand K.Levessuer

3) Elements of Discrete Mathematics- C.L.Liu

4) Discrete mathematics & its applications- K. Rosen

5) Discrete Mathematics and applications K. H. Rosen, Tata McGraw Hill Publishing Company

6) Discrete Mathematical Structures with Applications to Computer Science, J. P. Trembley, R. Manohar (Tata McGraw Hill Publishing Company)

Type.:DSC 4A
B.Sc(ECS)-I (Semester I)
Course Title:Basic Electronics
(Paper Code: Paper-I)
Credits: Theory –(2)Practicals– (2)
Total Lectures: 30 Hrs **Contact Hrs (L) :2.5**
University Evaluation: 40 Marks **Internal Evaluation: 10 Marks**

Course Objective:

The subject aims to provide the student with:

1. An understanding of basic Electronics abstractions on which analysis and design of electrical and electronic circuits and systems are based, including Digital circuits.
2. The capability to use abstractions to analyze and design simple electronic Circuits.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Learn how to develop and employ circuit models for elementary electronic components, e.g., resistors, inductors, capacitors, diodes and transistors.
2. Gain an intuitive understanding of the role of power flow and energy storage in electronic circuits.
3. Learn how to develop different power supplies in the computer system.

Unit I **[15]**

Introduction to components: Classification of Components (tree chart), Passive and Active Components.

Passive Components: Resistors: Classification of Resistor (Fixed and Variable)

Fixed Resistor: Carbon composition, Construction, Application.

Variable Resistor: Potentiometer, Construction, Working and Application, Colour coding of Resistor with examples.

Capacitors**:** Classification of capacitor (Electrolyte and Non Electrolyte), Electrolyte: construction of electrolyte,

Non-electrolyte (mica, ceramic): construction of Non electrolyte, Applications of both electrolyte and Non-electrolyte capacitors.

Variable Capacitor: Gang capacitor, Application of variable capacitors

Inductors**:** types of inductors (iron core and air core), its applications,

Transformers**:** Types of transformers (step up and step down), Construction of step down transformer, and its applications.

Unit I **[15]**

Semiconductor Devices and DC Power Supplies: Classification of Materials on thebasis of Energy Band Diagram.

Introduction to Semiconductor:P type semiconductor (Construction), N type Semiconductor (Construction), Formation of P-N Junction Diodes, Working of P-N Junction Diode, Applications of P-N junction Diodes.

Introduction to Bipolar Junction Transistors (BJT):Symbol, types, construction and Working of PNP and NPN transistor, applications of BJT

FET: Introduction of FET, classification of FET (J-FET and MOS FET), construction of J-FET and MOS FET, applications of FET.

DC Power Supplies: Introduction to power supplies,Rectifier: Definition, types of rectifier (Half, Full, Bridge), Construction and Working of rectifiers

Regulated power supplies (by using 3- pin positive (78xx) and negative voltage (79xx) regulator): Block diagram and working of regulated power supplies, applications of regulated power supplies.

SMPS: block diagram, working and applications.

Inverter: block diagram, working and applications.

Reference Books:

1. Principle of Electronics- V.K.Mehta (new e/d)
2. Electronics Principle- Malvino
3. Basic Solid State Electronic- B.L.Theraja
4. Principle of Electronics-P.C.Narayan Rao (Vol.I, II, III) New Age International
5. Basic Electronics and Linear Circuits: N.N. Bhargava, D.C. , Kulshreshtha, S.C. Gupta Tata McGraw Hill Publishing Company
6. Electronic Devices and circuits: Boy stead, Tata McGraw Hill

Type: DSC 4A
B.Sc(ECS)-I (Semester I)
Course Title:Advanced Electronics
(Paper Code: Paper-II)

Credits: Theory –(2) **Practicals– (2)**
Total Lectures: 30 Hrs **Contact Hrs (L) :2.5**
University Evaluation: 40 Marks **Internal Evaluation: 10 Marks**

Course Objective:

The subject aims to provide the student with:

1. An understanding of Advanced Electronics abstractions on which analysis and design of Integrated circuits(ICs) and Logic Families.

2. The capability to use abstractions to analyze display devices and sensors. Designing of PCB technologies electronics devices.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Learn how to develop the Integrated circuits (IC) in electronics systems. E.g. Computer systems, Microprocessor, Microcontroller, Mobile, etc.

2. Learn how to Manufacturing Resistors, Capacitors, Diode and Transistor in IC.

3. An understanding of different Display devices, Sensors and PCB technologies used in Computer System.

Unit I **[15]**

Integrated Circuits and Logic Families: Introduction to IC, Linear or Digital IC's (only Define), Fabrication process of IC (Explain in detail with diagram).

Steps of IC fabrication process:

1) Substrate Preparation (Wafer production)

2) Epitaxial Growth (N-Type and P- Type Layer)

3) Insulation Layer (Sio2)

4) Photolithography: i) Masking and Etching

5) Diffusion (Doping or Ion Implantation)

6) Isolation Process

7) Metallization

Monolithic IC's: Component Fabrication: Transistor, Diode, Resistor and Capacitor. Applications of IC's

Logic Families: Types of IC Families, TTL Subfamilies (74XX), TTL Logic Gate (AND OR NOT), MOS Families

Unit II **[15]**

Display Devices, Sensors and PCB Technology: LED (Construction, working and Application).
Seven segment LED Display (Construction, working and Application).
LCD Display (Construction, working and Application).
LED Display (Construction, working and Application).
Sensors: Introduction to Sensor
Types of Sensors: Temperature sensor (Thermistor), Photodiode, LDR, Opto coupler. (Construction, working and Application of Sensors)
Introduction to PCB Technology: Single layer and Multilayer PCB Technology.
Surface Mounting Devices (SMD), Surface Mounting Technology (SMT), Advantages and Applications of SMT and SMD.

Reference Books:

1. B.L.Theraja, Basic Solid State Electronic. S.Chand& Company Ltd.
2. V.K. Mehta, Principles of Electronics, S.Chand& Company Ltd.
3. Electronic Instrumentation byK.S.Kalsi, TMH Publication.
4. Electronic Measurements by U.A.Bakshi
5. and V.U.Bakshi, Technical Publication
6. Transducers and Display Systems by B.S.Sonde

Type: DSC 1B
B.Sc(ECS)-I (Semester II)

Course Title: Introduction to Web Technology

(Paper Code:I)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Objective:

1. Give the distinguishing characteristic of scripting language.
2. Discuss the reasons for and effectsof nonstandard client-side scripting language characteristics, such as limited data types,dynamic variable types and properties, and extensive use of automatic type conversion.
3. Develop event-driven programs that use HTML intrinsic event attributes, DOM events, listeners, and DOM-generated events.
4. Use the DOM to modify a document's attributes andstyle properties as well as to modify its parse-tree representation.

Course Outcomes:

1. Explain the history of the internet and related internet concepts that are vital inunderstanding web development.
2. Discuss the insights of internet programming and implement complete applications overthe web.
3. Demonstrate the important HTML tags for designing static pages and separate designfrom content using Cascading Style sheet.
4. Utilize the concepts of JavaScript.

Unit I **[10]**

Introduction to Web Design: Introduction to Networking, Introduction to Internet, Applications of the Internet. Introduction to HTML, Structure of HTML, Creating and opening HTML file, Singular and paired tags, Text formatting tag, Anchor tag, Lists, Image, Image Map, Table, Frames and Frameset, form.

Unit II **[20]**

Introduction to CSS and JavaScript: Introduction to CSS, Types of CSS, Use of CSS, Selectors, Properties, Values.

CSS Properties: Background, Text, Fonts, Link, List, Table, Box Model, Border, Margin, Padding, Display, Positioning, Floating, Opacity, Media type, Backgrounds, Animations, Multiple Column Layout, Navigation bar.

JavaScript:

Introduction to JavaScript, JavaScript Variables & Data types, Operators, Built-in functions in JavaScript,Control structure in JavaScript,DOM, Math, Array, History, Navigator, Location, Windows, String, Date, Document objects, user defined function, Form Validation, event & event handling in JavaScript.

Reference Books:-

1. HTML5 Black Book Kogent Learning Solutions Inc Dream-tech.
2. Beginning JavaScript and CSS Development with jQuery Richard York.
3. Beginning HTML and CSS Rob Larsen

Type: DSC 1B
B.Sc(ECS)-I (Semester II)

Course Title: Operting System

(Paper Code:II)

Credits: Theory – (2) **Practical's – (2)**

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objectives:

1. Students should be able to use system calls for managing processes, memory and the file system.
2. Describe the choices to be made in designing file systems.
3. Compare different approaches to file organization, recognizing the strengths and weaknesses of each.

Course Objectives:

1. Describe and analyze memory management and its allocation policies.
2. Identify the use and evaluate the storage management policies concerning different storage management technologies.
3. To understand different approaches to memory management.

Unit I **[20]**

Deadlocks: Concept of deadlock, Dead Lock Characterization, Resource Allocation Graph, Methods of deadlock Handling- Deadlock Prevention, Deadlock Avoidance -banker's algorithm, Deadlock Detection and Recovery.

Memory Management: Logical and Physical Address Space, Dynamic Loading, Overlays, Swapping,

Memory allocation: Contiguous Memory allocation – Fixed and variable partition – Internal and External fragmentation and Compaction, Paging, Segmentation.

Virtual Memory: Concept of virtual memory, Demand paging, Page fault, Page Replacement policies: Optimal (OPT), First in First Out (FIFO), Least Recently used (LRU).

Unit II **[10]**

File Management: File Management: File concept, Access methods, File types, File operation, Allocation methods (contiguous, linked and indexed).

Disk Management: disk structure, disk scheduling (FCFS, SSTF, SCAN, CSCAN).

Reference Books:

1. Operating System Concepts BySiberchatz and Galvin.
2. Modern O.S. By Andrews Tanenbaum.

Type.: DSC 2B
B.Sc(ECS)-I (Semester II)

Course Title: Object Oriented Programming using C++

(Paper Code: Paper-III)

Credits: Theory – (2) **Practical's – (2)**

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Objectives:

1. To understand how C++ improves C with object-oriented features.
2. To learn how to write inline functions for efficiency and performance.
3. To learn the syntax and semantics of the C++ programming language.
4. To learn how to design C++ classes for code reuse.
5. To learn how to implement copy constructors and class member functions.
6. To understand the concept of data abstraction and encapsulation.
7. To learn how to overload functions and operators in C++.
8. To learn how containment and inheritance promote code reuse in C++.
9. To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
10. To learn how to design and implement generic classes with C++ templates.
11. To learn how to use exception handling in C++ programs.

Course Outcomes:

1. escribe the procedural and object oriented paradigm with concepts of streams classes, functions, data and objects.
2. Understand dynamic memory management techniques using pointers, constructors, destructors, etc.
3. Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
4. Classify inheritance with the understanding of early and late binding, usage of exception handling, and generic programming.
5. Demonstrate the use of various OOPs concepts with the help of programs

Unit I **[15]**

Basics: Introduction to Object Oriented Programming, Overview of OOP features- Class, Object, Data Encapsulation, Data Abstraction, Polymorphism, Inheritance, Message Passing etc., Differences Between Object Oriented and Procedure Oriented Programming, Structure of a C++, Type Conversion, Dynamic Memory Allocation and De-Allocation, Inline function, Friend Functions, this Pointer, Constructors and Destructors, Operator Overloading, overloading unary and binary operator using friend function and member function.

Unit II **[15]**

Introduction and definition of Inheritance, Base and Derived Classes, Types of derivations, Types of inheritance- Single inheritance, Multiple inheritance, Multi-level inheritance, Hierarchical inheritance, Hybrid inheritance., Runtime Polymorphism- Virtual and pure virtual functions, Virtual Destructors, Stream, Stream Classes Hierarchy, File Stream classes- ifstream class, ofstream class, fstream class, Error Handling during File Operations.

Reference Books:

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

Type.: DSC 2B
B.Sc(ECS)-I (Semester II)

Course Title: Python - II

(Paper Code: Paper- IV)

Credits: Theory – (2) **Practical's – (2)**

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objective:

1. To learn the use of functions in programming.
2. To understand the use of modules and packages in the application hierarchy.
3. To understand python programming using the object-oriented programming principles.
4. To learn handling of various exceptions during the application development.
5. To understand the working with different file operations.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Write and implement a functional approach to application development.
2. Write and implement a modular approach to application development.
3. Design an application using object-oriented paradigm.
4. Create error free applications by applying the exception handling concept.
5. Design an application that contains the use of different files for data processing.

Unit I **[12]**

Functions: Difference between function and method, defining function, calling function, returning result from function, returning multiple values from function, functions are objects, formal and actual arguments, types of arguments, local, nonlocal and global variables, global keyword, recursive functions, anonymous functions or lambdas, using lambdas with filter(), map() and reduce() functions.

Modules and packages: what are modules in python, import statement, from…import statement, creating our own modules, importing modules, working with built-in modules- Math module, time module and random module. what are packages, creating and importing module from packages.

Unit II **[18]**

Python Object Oriented: Difference between procedure oriented and object oriented programming. Features of object oriented programming- classes and objects, inheritance, polymorphism, encapsulation, abstraction. Creating class, self-variable, constructor, types of variables, namespaces, types of methods, passing member of one class to another class, inner classes. Types of inheritance, super() method, method overloading, method overriding, abstract classes and interfaces.

Exception Handling: Error in python program, exceptions, steps in exception handling using try, except, else and finally blocks, types of exceptions-built-in exceptions and user defined exceptions, assert statement.

File Input Output: concept of files, Types of files in python, opening a file- the file opening modes, closing a file, working with text files containing strings, working with binary files, with statement, pickling and unpickling, seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories.

Reference Books:

1. Python: The Complete Reference by Martin C. Brown.
2. Core Python Programming, Dreamtech publications, by R. Nageswara Rao.
3. Python Programming, A modular approach, First Edition, Pearson, by Taneja Sheetal
4. Learning with Python, Dreamtech publications, by Allen Downey
5. Python Programming for the Absolute Beginner by Michael Dawson-Cengage Learning.

Type: DSC 1B
B.Sc(ECS)-I (Semester II)

Course Title: Linear Algebra

(Paper Code:III)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Outcome:

1. Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.
2. Understand the basics of combinatorics, and be able to apply the methods from these subjects in problem solving.
3. Be able to use effectively algebraic techniques to analyse basic discrete structures and algorithms.
4. Understand asymptotic notation, and its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.
5. Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

Unit I **[15]**

Solution of System of linear Equations : Matrix definition and elementary results, types of matrices, Trace of matrix & properties of trace of matrix,Rank of matrix (problem for 2x2,3x3) & properties of rank matrix, linear equations, System of linear equations, matrix representation of system, augmented matrix, row echelon form and reduced row echelon form, definitions of solution, trivial solution and non-trivial solution, homogeneous system and non-homogeneous system of linear equations, consistent and inconsistent system of linear equations, solution of system by using Gauss elimination method (with row pivoting) and Gauss−Jordan elimination method, problems.

Determinants: Definition of determinant, properties of determinant function, cofactor of an element, cofactor matrix, adjoint of matrix, determinant of a matrix by using cofactor expansion method, inverse of a matrix by using adjoint method, Solution of system of linear equations by using Cramer's rule, problems, Area of triangle by using determinants.

Unit II **[15]**

Complex Number: Definition of Complex Number, Algebraic Operations and their properties, Conjugate of complex number, modulus and argument of complex number and their properties, square root of complex number, polar form of complex number, De Moivre's Theorem, and its application.
Eigen values and Eigen Vectors: Characteristic polynomial, characteristic equation, characteristic matrix, cayley Hamilton theorem and its application Eigen value and Eigen vectors (definition and problems for 2 x 2 ,3x3 matrix),Properties of Eigen Value & Eigen vectors), diagonalization of a matrix.

Reference Books:

1. Linear Algebra and it's applications, Gilbert Strang
2. Linear Algebra and it's applications David C Lay, Pearson Education India, third edition
3. Matrices, shanti Narayan

Type: DSC 1B
B.Sc(ECS)-I (Semester II)

Course Title: Discrete Mathematics

(Paper Code:IV)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objectives:

1. To familiarize the prospective learners with a mathematical structure that is fundamentally discrete.
2. To introduces set and functions, solving recurrence relations and different counting principles.
3. To study or describe objects or problems in computer algorithms and programming languages.

Course Outcome:

1. To understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.
2. To understand the basics of combinatorics, and be able to apply the methods from these subjects in problem solving.
3. To use effectively algebraic techniques to analyse basic discrete structures and algorithms.
4. To understand asymptotic notation, and its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.
5. To understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

UNIT I **[15]**

Relations: Set, subset, power set, Cartesian product, relation, types of relation, void, universal, identity, reflexive, symmetric, equivalence, anti symmetric, partial ordering, asymmetric, inverse relation, Matrix representation and graphical representation of relation, in degree and out degree, closure of relation, transitive closure, Warshall's algorithm

Function: Definition of function as relation, domain, co-domain and range of function, injective (one-one) function, surjective function (Onto Function), bijective function, composition of function

UNIT II **[15]**

Recurrence Relations: Explicit formula, recursive formula, recurrence relation , Homogeneous Recurrence Relation with constant coefficients, homogeneous solution, Linear Recurrence Relation with constant coefficients, particular solution, total solution.

Counting principles: Cardinality of a set, Pigeonhole principle, Addition principle, Multiplication principle, Inclusive-exclusive principles for two sets & three sets, Problems

Reference Books:

1. Combinatorics - V.Krishnamurthy
2. Discrete Mathematical structure for Computer Science, Alan Doerr and K Levassuer
3. Elements of Discrete Mathematics - C.L.Liu
4. Discrete mathematics & its applications- K. Rosen
5. Discrete Mathematics and applications K. H. Rosen,Tata McGraw Hill Publishing Company

Type: DSC 4B
B.Sc(ECS)-I (Semester II)

Course Title: Digital Electronics and Microprocessor

(Paper Code:III)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

1. To learn Boolean algebra and logic gates
2. To study digital logic families and their important features
3. To develop a designing and analyzing attitude about sequential circuits
4. To develop a designing and analyzing attitude about combinational circuits
5. To learn 8085 Microprocessor Architecture and Assembly language Programming.

Course Outcomes:

1. Design and construct logic as well as arithmetical circuits
2. Calculate various important parameters of Digital logic families
3. Design & analyze combinational logic circuits
4. Design & analyze sequential logic circuits
5. To Execute 8085 Microprocessor Assembly language programming.

UNIT I **[15]**

Logic Gates and Combinational Logic circuits: Introduction to logic gates, OR, AND, NOT, NAND, NOR, XOR, XNOR, Pin function of IC 7432, 7408, 7404, 7400, 7402, 7486 Applications of Logic Gates.

Combinational circuit: Introduction to the combinational circuit, Half adder, full adder, Half subtractor, Multiplexer(4:1), Demultiplexer(1:4), Encoder (4:2), Decoder(2:4), Applications of Combinational Logic Circuits.

Sequential circuits: Concept of sequential circuits,

Flip-flops: RS, Clocked RS, JK, Master Slave JK, D Flip-flop, Pin configuration of IC-7474 , Counter-synchronous, asynchronous(3 bit up counter and Down Counter), modulus of Counter (Mod 2, Mod 5 ,Mod 10), Pin configuration of IC 7490, Shift register (SISO, SIPO, PIPO, PISO) Pin Configuration of IC 7495), Applications of Sequential Logic Circuits.

UNIT II **[15]**

Fundamentals of 8085 Microprocessor: Introduction to microprocessor, Basic system bus architecture, Concept of T state Machine cycle, Instruction cycle, pin function of 8085 microprocessor, internal architecture of 8085 microprocessor.

Instruction set of 8085: instructions Format, Classification of instruction set, Addressing modes, Assembly language programming of 8085(addition, subtraction, division, multiplication) Intel 8085 microprocessor features

Reference Books:

1. Morris Mano Computer System Architecture (3rd Edition) PHP
2. Digital principle & applications – Malvino Leech.3
3. Digital principle – Floyed.
4. Digital electronics – C. F. Strangio
5. Microprocessor Architecture programming and Application,-Ramesh Gaonkar

Type: DSC 4B
B.Sc(ECS)-I (Semester II)

Course Title: Introduction to Microcontroller and Embedded System

(Paper Code:IV)

Credits: Theory – (2) Practical's – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.5
University Evaluation: 40 Marks **Internal Evaluation:** 10 Marks

Course Objective:

1. To develop specialists in hardware-software co-design for application specific Electronic systems.
2. To prepare students for demonstrating the acquired knowledge.
3. To encourage the student to develop skills for accepting challenges of upcoming technological advancements.

Course Outcomes:

1. Design, test and critically evaluate embedded solutions to real-world situations using digital components (sequential and combinational).
2. Recognize the key features of embedded systems in terms of computer hardware and be able to discuss their functions. You will be aware of the key factors affecting computing hardware evolution.
3. Design, test and critically evaluate embedded solutions to real-world situations using (embedded) computer systems interfaced with digital hardware

UNIT I **[15]**

Fundamentals of Microcontroller: Introduction to 8051 Microcontroller, Difference between Microprocessor and Microcontroller,Overview and features of MCS-51 Family, Salient features and block diagram of 8051, Pin description, Internal memory, Registers and SFRs, Port structure, Clock and Reset circuit. Timer/counter, Serial port

Instruction set of 8051:

Instruction set, Instruction types - Data transfer, Arithmetic, Logical instruction execution and timing. Simple programs based on ports using Embedded C (LED on/off, Port in Port Out)

UNIT II **[15]**

Introduction to Embedded System: Definition of an embedded system, Types of Embedded System, Basic architecture of embedded system, Concept of Hardware and Software Design in Embedded System, characteristics of Embedded System, Advantages and Application of Embedded System.

Introduction to Embedded c and Development Tools: Structure of Embedded C, I/O port programming

Development tools for Embedded System: Introduction to Keil Micro vision, Steps involved in Programming with Keil Micro Vision simulation. Introduction to flash magic, Steps involved in programming of the microcontroller. Basic Concepts of Designing of microcontroller based embedded system

Reference Books:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinley's, "The 8051 microcontroller and Embedded systems using Assembly and C", Second Edition Pearson
2 Kenneth Ayala, Delmar Cengage Fearning, "The 8051 Microcontroller Architecture, Programming and Applications" , Third Edition TMH
3. Raj Kamal Embedded systems Architecture, Programming and Design TMH

Type: Core Practical

Practical based on: DSC 1A and 1B

(Paper Code: Practical-I**)**

Tools / Software: Microsoft Office / Notepad++ / Turbo C++ Editor / DosBox

1. DOS – external and internal commands, batch files.
2. MS – WORD – Creating new documents, typing, deleting, selecting text, undo, Redo, formatting text Paragraphs, line spacing, margins, page setup, headers and footers.
3. Writer's tools – spelling checker, format painter, creating mail merge document.
4. MS – EXCEL - Creating worksheet, Graphs, resizing graphs, formulas, if Statement, types of functions.
5. MS-Powerpoint-Creating presentation, slideshow, adding slides, inserting clip arts, smart art, images, sound files linking etc.
6. Internet – creating e – mail accounts, browsing.
7. Design HTML page to display student Information.
8. Design HTML page for all lists.
9. Design HTML page for display Table .
10. Design HTML page for Image map, frameset tags.
11. Create a web page using the Internal/ Linked/ External style sheet.
12. Create web page using Text formatting properties, CSS Borders, Margin Properties, Color properties.
13. Create web page Using DIV and SPAN tag properties.
14. Write a JavaScript code working with functions: the alert Box, the confirm Box , the prompt Box etc.
15. Solve the Following program using JAVA Script to check the given number is:
    I. even or odd
    II. Prime or not
    III. palindrome or not
    IV. a sum of digit of given number
16. create standard calculator using JavaScript.
17. write a C program for implementation of Priority scheduling algorithms
18. write a C program for implementation of Round Robin scheduling algorithms
19. write a C program for implementation of FCFS and SJF scheduling algorithms.
20. write a C program for the implementation of SJF scheduling algorithms.
21. write a C-program to implement the producer – consumer problem using semaphores.
22. Write a C program to simulate the concept of Dining-Philosophers problem.
23. write a c program to implement Threading and Synchronization Applications.

24. write a C program to implement banker‟s algorithm for deadlock avoidance.
25. write a C program to implement algorithm for deadlock detection.
26. write a C program for implementation memory allocation methods for fixed partition
27. Write a C program to simulate the following contiguous memory allocation techniques
    a) Worst-fit b) Best-fit c) First-fit
28. write a program to simulate the MVT algorithm
29. write a c program to implement Paging technique for memory management.
30. write a C program for implementation of FIFO page replacement algorithm.
31. write a c program to implement LRU page replacement algorithm.
32. write C program to implement LFU page replacement algorithm.

Type: Core Practical

Practical based on: DSC 2A and 2B

(Paper Code: Practical-II**)**

Practical based on: DSC 2A

Tools / Software: Turbo C++ Editor / DosBox

1. Write a program to check a given number is PRIME or not
2. Write a program to check a given number is PALINDROME or not
3. Write a program to check a given number is PERFECT or not
4. Write a program to check a given number is ARMSTRONG or not
5. Write a program to check a given number is STRONG or not
6. Write a program to display prime numbers between 1 to 1000
7. Write a program to display the first 100 prime numbers.
8. Write a program to find the smallest and largest element from an array
9. Write a program to search for an element in an array
10. Write a program to display diagonal elements from a given matrix.
11. Write a program to display the transpose of a given matrix.
12. Write a program to display the addition and multiplication of two matrices.
13. Write a program to count vowels from a given string
14. Write a program to find the length of a given string without using the library function
15. Write a program to display the factorial of a given number using recursion
16. Write a program to display the Fibonacci series using recursion
17. Write a program to swap two numbers using
    a) Call by value b) Call by reference
18. Write a program to display the following patterns

a)
```
P
PR
PRO
PROG
PROGR
PROGRAM
```

b)
```
   A
  ABA
 ABCBA
ABCDCBA
 ABCBA
  ABA
   A
```

c)
```
    1
  2 3
4 5 6
```

19. Define a class to represent a bank account. Include the following members:

Data Members: a) Name of depositor b) Account number
c) Type of account d) Balance amount in the account

Member function: a) To assign initial values b) To deposit an amount
c) To withdraw an amount after checking the balance
d) To display name and balance

20. Write a Program using class to process Shopping List for a Departmental Store. The list includes details such as the CodeNo and Price of each item and performs the operations like Adding, Deleting Items to the list, and Printing the Total value of an Order.

21. Write a Program that creates & uses an array of the object of a class. (for eg. implementing the list of students of a College having details such as Roll no., Name, class, Contact No.)

22. Write a Program to find the Maximum out of Two Numbers using the friend function. Note: Here one number is a member of one class and theother number is a member of some other class.

23. Write a Program to swap private data members of classesnamed class_1, class_2 using the friend function.

24. Write a Program using a copy constructor to copy data of anobject to another object.

25. Write a Program illustrating how the constructors areimplemented and the order in which they are called when theclasses are inherited. Use three classes named alpha, beta, gamma such that alpha, beta is the base class and gamma is the derived class inheriting alpha & beta.

26. Write a Program to design a student class representing student roll no. and a test class (derived class of student) representing the scores of the student in various subjectsand a sports class representing the score in sports. The sports and test class should be inherited by a result class having thefunctionality to add the scores and display the final result fora student.

27. Write a program to maintain the records of a person with details (Name and Age) and find the eldest among them. Theprogram must use this pointer to return the result.

28. Write a program illustrating the use of a virtual function.

Practical based on: DSC 2B

Tools/(Open Source) Softwares: Python Interpreter, Idle Graphics Window, Command Prompt, System Prompt, PyCharm, Visual Studio Code, Jupyter Notebook.

1) Write a python program to find the sum of a list of numbers using for loop.
2) Write a python program to display stars in right angled triangular form using nested for loops.
3) Write a python program to display multiplication table from 1 to 10 using nested for loops.
4) Write a python program to display numbers from 10 to 6 and break the loop when the number about to display 5.
5) Write a python program to display numbers from 1 to 5 using the continue statement.
6) Write a python program to find the first occurrence of sub string in a given main string.
7) Write a python program to display elements in a list in reverse order.
8) Write a python program to accept elements in the form of a tuple and display their sum and average.
9) Write a python program to create a dictionary with employee details and retrieve the values upon giving keys.
10) Write a function to test whether a number is prime or not.
11) Write a function to return the addition and subtraction of two numbers using a function return two values.
12) Write a python program to demonstrate the different methods of array module.
13) Write a python program to demonstrate the types of array.
14) Write a python program to understand the positional arguments of a function
15) Write a python program to understand the keyword arguments of a function
16) Write a python program to understand the default arguments in a function
17) Write a python program to understand Variable length arguments in a function.
18) Write a python program to understand Anonymous (lambda) Function.
19) Write a python program to understand local, nonlocal and global variables.
20) Write a python program to create a module and import it.
21) Write a python program to create a package and import it.
22) Write a python program to demonstrate the instance method, class method and static method.
23) Write a python program to demonstrate inner classes.
24) Write a python program to demonstrate Constructors in Inheritance.
25) Write a python program to demonstrate method overloading.
26) Write a python program to demonstrate method overriding.
27) Write a python program to read all the strings from the text file and display them.

28) Write a python program to append data to an existing file and display them.
29) Write a python program to count a number of lines, words and characters in a file.
30) Write a python program to copy an image file into another file.
31) Write a python program to apply different manipulation operations of directories.
32) Write a python program to handle the ZeroDivisionError exception.
33) Write a python program to handle syntax errors given by eval( ) function.
34) Write a python program to handle IOError produced by open( ) function.
35) Write a python program to illustrate the use of raising an exception

Type: Core Practical

Practical based on: DSC 3A and 3B

(Paper Code: Practical-III**)**

Tools / Software: Scilab/MATLAB / MATHEMATICA / MAPLE/ C / C++ / Python

1. Interpolation and extrapolation by using Newton's forward difference interpolation formula and Newton's backward difference interpolation formula.
2. Interpolation and extrapolation by using Lagrange's interpolation formula and Newton's divided difference interpolation formula.
3. Numerical Integration.
4. Solution of Ordinary Differential Equations by using Euler's Method, Runge Kutta Second Order and Fourth Order Method.
5. Adjacency and incidence matrix.
6. Operations on graphs. (Union, Intersection, complement)
7. Operations on graphs. (Ringsum and product)
8. Shortest path and its weight by using Dijkstra's shortest path algorithm.
9. Travelling salesman problem, Chinese Postman problem for both Eulerian and non Eulerian graphs.
10. Shortest Spanning tree by using Kruskal's Algorithm.
11. Solution of a system of linear equations by using direct methods. (Gauss Elimination Method and Gauss Jordan Elimination Method)
12. Determinant of a matrix by using the cofactor expansion method.
13. Inverse a matrix by using the adjoint method.
14. Cramer's rule
15. Transitive closure of relation by using Warshall's algorithm.
16. Eigen values and Eigen vectors (for 2x2 and 3x3 matrix)
17. Counting Principles.

Type: Core Practical

Practical based on: DSC 4A and 4B

(Practical-IV)

Tools / Software: Digital trainer kit/circuit board, Anshuman 8085 Kit, Anshuman 8051 kit, Keil Micro-vision Simulator, Programmer/Burner 8051 Kit.

1. Study of instruments & measurement techniques (Multimeter, CRO, and F.G.)
2. Study of electronic components
3. Half Wave Rectifier
4. Full wave rectifier.
5. Positive Voltage regulator by using three pin IC (7805)
6. Negative Voltage regulator by using three pin IC (7905)
7. Transistor as a switch
8. Study of Logic Gates
9. Interconversion of Gates by using NAND
10. Interconversion of Gates by using NOR
11. Demorgans Theorem
12. Study of Half adder
13. Study of Full Adder
14. Study of Flip Flop R-S (using NAND) and D Flip Flop
15. Arithmetic Operation using 8085
16. Addressing Modes-I
17. Block transfer using 8085
18. Arithmetic Operation using 8051
19. Interfacing of LED with 8051 Microcontroller using Simulator/Anshuman Kit

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

Equivalent Subject for Old Syllabus B.Sc. (ECS) - I (Semester–I and II)

Semester-I		
Sr. No.	**Name of the Old Paper (w.e.f. 2019-2020)**	**Name of the New Paper (w.e.f. 2022-2023)**
1	English Paper I Part-A (communication skill)	English Paper I Part-A (communication skill)
2	Fundamentals of Programming using C and C++ – I	Programming using 'C'
3	Fundamentals of Programming using C and C++ – II	Object Oriented Programming using C++
4	Computer System Architecture – I	No Equivalence
5	Computer System Architecture – II	No Equivalence
6	Fundamental of Computer system – I	Fundamental of Computer
7	Fundamental of Computer system – II	Operating System – I
8	Numerical Methods – I	Linear Algebra(Sem-II)
9	Numerical Methods – II	Numerical Method
Semester-II		
Sr. No.	**Name of the Old Paper (w.e.f. 2019-2020)**	**Name of the New Paper (w.e.f. 2022-2023)**
1	English Paper I Part-B (communication skill)	English Paper I Part-B (communication skill)
2	Programming in JAVA – I	Python-I
3	Programming in JAVA – II	Python-II
4	Discrete Structures – I	Graph Theory (Sem-I)
5	Discrete Structures – II	Discrete Mathematics
6	Introduction to Web Designing – I	Introduction to Web Designing
7	Introduction to Web Designing – II	No Equivalence
8	Digital Electronics – I	Digital Electronics and Microprocessor
9	Digital Electronics – II	No Equivalence

Nature of Question Paper

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Nature of Question Paper for Choice Based Credit System(CBCS) Semester Pattern

Faculty of Science (w.e.f. June 2022 for B.Sc[ECS]-I and from June 2023 B.Sc[ECS]-II)

Time: 2 Hrs. Total Marks:40

Instructions: (Instructions may differ from subject to subject)

1. All questions are compulsory.
2. Draw **net diagrams** and give equestion whenever necessary.
3. Figur to the right indicates **full marks**.
4. Use of logarithmic table and calculator is allowed.

Q.No.1) Multiple choice questions. **(8)**

1. _____

a) b) c) d)

2.

3.

4.

5.

6.

7.

8.

Q.No.2) Answer any four of the following. **(8)**

I.

II.

III.

IV.

Q.No.3) Write a short note on any two of the following. **(8)**

I.

II.

III.

Q.No.4) Answer any two of the following. **(8)**

I.

II.

III.

Q.No.5) Answer any one of the following. **(8)**

I.

II.

For Science faculty: CA- Contineous Assessment (Internal Examinations) of Total Marks: 10

Pattern / Examination nature may be as follows:

One internal examination of 10 marks or two examinations of 5 marks each.

Open book examination / Home Assignment / Classroom test / Seminar / Field Work report / Project Report etc.