PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

NAAC Accredited-2015
'B' Grade (CGPA 2.62)

Name of the Faculty: Science and Technology

Multiple Entry and Multiple Exit Option (NEP-2020)

Syllabus: B. Sc. (ECS) Part-I

Name of the Programme: B.Sc.[ECS]-I (Sem.–I and II)

(Syllabus to be implemented w.e.f. June 2024)

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

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

Syllabus (Semester – I and II)

Multiple Entry and Multiple Exit Option (NEP-2020)

With Effect from June 2024

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

B.Sc. (Entire Computer Science)

Preamble: B. Sc. (Entire Computer Science) is multiple entries and multiple exit option 4-year programme specializing in 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 Computer Programming theory, Mathematical foundation, Advanced Programming using Python, machine learning, Java, etc. The course curriculum is inclusive of theory and practical which makes the students well trained and skillful in programming, software, and networks.

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 a comprehensive understanding of the theory and its application 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 advanced 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. To enhance the programming skills of 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, Diary Diploma shall be allowed to enter upon the B. Sc. Part-I Course.

OR

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 behaviour that students acquire in their graduation through the program

Programme Learning Outcomes for BSc(Entire Computer Science):

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

PLO-1. Demonstrate an 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 an ethical code of conduct in the usage of the Internet and Cybersystems.

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

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

PLO-8. Ability to operate, manage, deploy, and 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 development19

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.

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

Multiple Entry and Multiple Exit Option (w.e.f. 2024-25)

NEP-2020

LEVEL - 4.5

First Year Undergraduate

Subject/ Core Course	Name and Type of the Paper		Hrs./week			Total Marks Per	UA	CA	Credits
	Type	Name	L	T	P	Paper			
B.Sc.(Entire Computer Science)-I Sem-I									
Major	DSC1-1	Digital System	2	--	--	50	30	20	2
	Practical based on DSC1-1		--	--	4	50	30	20	2
	DSC2-1	OOP'S with C++-I	2	--	--	50	30	20	2
	Practical based on DSC2-1		--	--	4	50	30	20	2
	DSC3-1	Python-I	2	--	--	50	30	20	2
	Practical based on DSC3-1		--	--	4	50	30	20	2
Vocational and Skill Enhancement Courses (SEC/VSC)	SEC1	Basics of Web Designing	2	--	--	50	30	20	2
Ability Enhancement Course (AEC)	L1-1	English	2	--	--	50	30	20	2
Indian Knowledge System	IKS		2	--	--	50	30	20	2
Value Education Courses:	VEC1	Constitution of India	2	--	--	50	30	20	2
Co-curricular Courses	CC1		2	--	--	50	30	20	2
Semester-I Total			**16**	**0**	**12**	**550**	**330**	**220**	**22**

B.Sc.(Entire Computer Science)-I Sem-II									
Major	DSC1-2	Basics of Operating System	2	--	--	50	30	20	2
	Practical based on DSC1-2		--	--	4	50	30	20	2
	DSC2-2	OOP'S with C++-II	2	--	--	50	30	20	2
	Practical based on DSC2-2		--	--	4	50	30	20	2
	DSC3-2	Python-II	2	--	--	50	30	20	2
	Practical based on DSC3-2		--	--	4	50	30	20	2
Vocational and Skill Enhancement Courses (SEC/VSC)	SEC2	Advanced Web Designing	2	--	--	50	30	20	2
Generic/ Open Elective Courses	GE1/ OE1	Computational Mathematics	2	--	--	50	30	20	2
Ability Enhancement Course (AEC)	L1-2	English	2	--	--	50	30	20	2
Value Education Courses:	VEC2	Environmental Science	2	--	--	50	30	20	2
Co-curricular Courses	CC2		2	--	--	50	30	20	2
Semester-II Total			**16**	**0**	**12**	**550**	**330**	**220**	**22**
Grand Total			**32**	**0**	**24**	**1100**	**660**	**440**	**44**

Abbreviations:

Discipline Specific Course:DSC;

Value Education Courses: VEC;

Generic / Open Electives: OE;

Vocational Skill and Skill Enhancement Courses: VSEC;

Vocational Skill Courses: VSC;

Co-curricular Courses: CC;

Skill Enhancement Courses: SEC;

Ability Enhancement Courses: AEC;

Indian Knowledge System: IKS

Total Credits for B.Sc[ECS]-I (Semester I and II): 44

Medium of instruction: English

I. Practical Examination is the Semester wise after theory Examination.

II. Duration of Practical Examination as per respective BOS guidelines.

III. Separate passing is mandatory for Theory, Internal and Practical Examination.

Exit Option at Level 4.5: Students can exit after Level 4.5 with under certificate course in Computer Programming if he/she complete the courses equivalent to a minimum of 44 credits and an additional.

4 credits core NSQF course/Internship.

Course Structure:

Lectures and Practicals should be conducted as per the scheme of lectures and practicals indicated in the course structure.

Teaching and Practical Scheme

I. Contact session for teaching 60 minutes each.

II. One Practical Batch should be of 20 students.

Assessment

I. The final practical examination will be conducted by the University appointed examiners internal as well as external at the end of the semester for each lab course and marks will be submitted to the university by the panel.

II. The practical examination will be conducted semester-wise to maintain the relevance of the respective theory course with the laboratory course.

III. The final examinations shall be conducted at the end of the semester.

Practical Examination:

I. Each paper carries 30 Marks.

II. ***Duration of Practical Examination:*** 2 Hrs.

III. ***Nature of Question Paper:*** There will be four questions of 10 marks each. Students will attempt

any two out of four questions.

IV. Certified Journal carries 5 Marks and Viva carries 5 Marks.

Standard of Passing:

I. Minimum 12 marks in each subject. There shall be separate passing for theory (semester end exam and Internal) and practical also.

Board of Paper Setters /Examiners:

For each semester-end examination, there will be a board of Paper setters and examiners for every course. While appointing paper setters/examiners, care should be taken to see that there is at least one person specialized in each unit of the course.

Credit system implementation:

As per the University norms.

Fees Structure:

As approved by the PAHS University fee fixation committee.

Intake Capacity: 80

Type: DSC1-1

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

Course Title: Digital System

(Paper Code: I)

Credits: Theory – (2) Practical – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objectives:

1. To provide a comprehensive introduction to the fundamental axioms, theories and conventions underlying the operation of digital systems.
2. To equip students with the necessary skills which will allow them to analyse, design, test, and simulate the operation of basic digital circuits.
3. To utilize a variety of digital logic design and simulation tools.

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

1. Identify a digital system and its main characteristics, and differentiate between digital and analogue systems
2. Describe the concepts of binary numbers and binary encoding, and perform conversions between binary, decimal, and hex numbers and between binary codes
3. Perform basic mathematical operations using binary numbers, and design digital systems capable of performing such operations.
4. Describe theorems and axioms of Boolean Algebra, and utilize them effectively in the process of designing digital systems.
5. Model, analyse, design, test, and simulate the operation of combinational and sequential circuits using analytic and modular methodologies and tools.
6. Explain the concept of memory in digital systems, and design basic memory modules .

Unit I: **[18]**

History and Overview: History, applications of digital systems, digital signals and analog signals, advantages and disadvantages of digital systems.

Number System: The binary, decimal and hexadecimal number system, conversion of binary, decimal and hexadecimal number system

Signed and unsigned binary numbers, representing signed binary numbers using the 2’s

complement method, performing basic mathematical operations using binary numbers (addition, subtraction, multiplication, division), design of adder and subtractor circuits.

Block diagram – ALU, memory unit, control unit, motherboard, SMPS, expansion slots, serial and parallel ports.

Unit II: [12]

Boolean Algebra fundamentals: definition of boolean algebra and logic gates, types of logic gates(AND, OR, NOT, NAND, NOR, XOR, XNOR), DeMorgan's Theorem, implementation of logic circuits using boolean equation and truth table.

Microprocessor: History and overview, comparative study of 8085, 8086 and Pentium processor, architecture of Pentium microprocessor, features of Pentium microprocessor, applications of Pentium. addressing modes, instruction set, types of programming languages, assembly language programming, applications of Pentium.

Reference Books:

1. Digital Systems: Principles and Applications by Ronald J. Tocci, Neal S. Widmer, and Greg Moss
2. Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog by M. Morris R. Mano, and Michael D. Ciletti
3. Fundamentals of Digital Logic with VHDL Design, by Stephen Brown and Zvonko Vranesic
4. Digital Fundamentals, by Thomas L. Floyd
5. Digital Logic Design, by B. Holdsworth
6. System Programming byJohn J. Donowon
7. System programming and Operating System by Dhamdhere
8. System Software by Beck:

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

Course Title: Practical based on DSC1-1

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L): 4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Tools / Software: NASM (Netwide Assembler)/ MASM (Microsoft Macro Assembler) / TASM (Turbo Assembler)etc.

1. **Steps for creating source code.**
2. **Programs Involving Arithmetic and logical operations**

I. Write an ALP to add 2 Multibyte no.

II. Write an ALP to subtract two Multibyte numbers.

III. Write an ALP to multiply two 16-bit numbers.

IV. Write an ALP to divide two numbers.

V. Write an ALP to multiply two ASCII no.

VI. Develop and execute and assembly language program to perform the conversion from BCD to binary.

VII. Write an ALP to convert binary to BCD.

VIII. Write an ALP to find the square of a number.

IX. Write an ALP to find the cube of a number.

Programs Involving Bit manipulation instructions like checking

I. Write an ALP to separate odd and even numbers

II. Write an ALP to separate positive and negative numbers

III. Write an ALP to find logical ones and zeros in a given data

Write an ALP to demonstrate the

I. AND Gate

II. OR Gate

III. NOT Gate

IV. NAND Gate

V. NOR Gate

VI. XOR Gate.

Type: DSC2-1
B.Sc(ECS)-I (Semester I)
Course Title: OOP'S with C++ - I
(Paper Code:II)

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

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2
University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objective:

Students will be learn:

1. The basic programming and OOPs concepts
2. Creating C++ programs
3. Tokens, expressions and control structures in C++
4. Arranging same data systematically with arrays
5. Classes and objects in C++

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

1. Describe OOPs concepts
2. Use functions and pointers in your C++ program
3. Understand tokens, expressions, and control structures
4. Explain arrays and strings and create programs using them

Unit-I **[15]**

Introduction to (Object Oriented Programming)OOP: Introduction to algorithm and flowchart, Introduction to OOP, Features OOP's- Class, Object, Data Abstraction and encapsulation, Data hiding, Message passing,
polymorphism, inheritance, persistency, delegation, extensibility.
Comparison between POP(Procedural Oriented Programming) and OOP, Advantages of OOP's, Application of OOP

Introduction to C++:

History of C++, C++ basics(C++ tokens)- Keywords, identifiers, data types, constants, operators, special symbols, control flow statements- Decision and iterative statements, Types of Variables- Value, pointer and reference, Structure of C++ program, Basics Input and Output- cin, cout objects, Introduction to array, pointer and template, Function and its types,

Default argument, Parameter passing methods, inline function

Unit-II **[15]**

Classes and Objects:

Introduction to class and object, Defining class (class specification), Creating object, Access specifier (Visibility modes)-public, protected, private, Class members- data members, member function, Defining member function inside and outside the class, Static data members, static member functions, Pointer to object, Array of object, Returning objects from functions, Passing object as parameter by value, by pointer, by reference, Dynamic memory allocation (new, delete), Friend function and friend class, nesting of classes, Constructors, characteristics of constructor, Types of constructor- default, parameterized and copy Constructor overloading, Constructor with default argument,

Destructor, characteristics of destructor, Static polymorphism (Function and Operator overloading) , rules to overload operator, unary and binary operator overloading, overloading operator using member function and friend function, Type conversion (type casting)- implicit and explicit.

Reference books:

1. OOP in C++ – E-balagurusamy
2. Mastering C++-K. R. Venugopal
3. The Complete Reference C++-Herbert Schildt

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

Course Title: Practical based on DSC2-1

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L):4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

1) Write different programs in 'C++' language that shows use of array, pointers variable, reference variable, cin and cout objects, scope resolution operators, basic operators
2) Write a program that shows use of class and object.
3) Write a program that shows parameter passing techniques in C++
4) Write a program that shows defining member function inside and outside of class body
5) Write a program that demonstrate use of inline function
6) Write a program to implement function overloading concept
7) Write a program to implement parameterized and copy constructor
8) Write a program that shows use of static data member and static member function.
9) Write a program that shows use of nesting classes.
10) Write a program that shows passing and returning object from function.
11) Write a program that shows use of new and delete operator
12) Write a program that shows explicit type conversion
13) Write a program to overload different unary and binary operators by using friend and member function.
14) Write a program to calculate factorial of given number by using recursion.
15) Write a program for addition, subtraction, multiplication and division of two complex numbers by using return by object method.
16) Create 2 distance classes "class A" stores distance in meter and cm and "Class B" stores distance in feet and inches and add two distances by friend function and display the result.
17) Generate the result for 5 students with following data - Name, exam no, Theory marks in 5 subjects, grade(Use array of object concept).
18) Write a program for constructor overloading.
19) Write a program to calculate root of quadratic equation by using default argument constructor.

20) Write a program to demonstrate friend function, friend class, member function of a class is friend to another class.

21) Write a program to count no. of objects created by using static data member & member function.

22) Write a program to overload unary operators (++, - -, -).

23) Write a program to overload binary operator.(+, -, *, /, %) by using member function and friend function.

Type.: DSC3-1

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

Course Title: Python - I

(Paper Code: Paper-IV)

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

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2

University Evaluation: 30 Marks **Internal Evaluation:** 20 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 the Integrated Development Environment (IDE).
4. Write and apply decision structures for different operations.
5. Write loop structures to perform iterative tasks.

UNIT I

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

Functions: Difference between function and method, defining a function, calling function, returning result from a function, returning multiple values from a 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

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.

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

Course Title: Practical based on DSC3-1

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L):4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

1. Write a Python program to find the sum of a list of numbers using a 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 a 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 a substring 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.

Type: SEC1

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

Course Title: Basics of Web Designing

Credits: Theory – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2.

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objective:

1. Give the distinguishing characteristics of scripting language.
2. Discuss the reasons for and effects of 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 and style 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 in understanding web development.
2. Discuss the insights of internet programming and implement complete applications over the web.
3. Demonstrate the important HTML tags for designing static pages and separate design from 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 and 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 and 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: DSC1-2

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

Course Title: Basics of Operating Systems

(Paper Code: V)

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

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objective:

Students will try to learn:

1. To understand the main components of an OS and 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 of 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 :

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 By Siberchatz and Galvin.
2. Modern O.S. By Andrews Tanenbaum.

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

Course Title: Practical based on DSC1-2

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L):4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

1. write a Python program for implementation of Priority scheduling algorithms
2. write a Python program for the implementation of round-robin scheduling algorithms
3. write a Python program for the implementation of FCFS and SJF scheduling algorithms.
4. write a Python program for the implementation of SJF scheduling algorithms.
5. write a Python Program to implement the producer–consumer problem using semaphores.
6. Write a Python program to simulate the concept of the Dining-Philosophers problem.
   write a c program to implement Threading and Synchronization Applications.

Type.: DSC2-2

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

Course Title: OOP'S with C++ - II

(Paper Code:II)

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

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objective:

Students will try to learn:

1. Inheritance and Polymorphism in C++
2. Files management and templates in C++
3. Handling exceptions to control errors.

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

1. Explain Inheritance, Polymorphism and create programs using them
2. Describe and use constructors and destructors
3. Understand and employ file management
4. Demonstrate how to control errors with exception handling.

Unit I: **[12]**

Inheritance and Runtime Polymorphism: Introduction to inheritance, benefits, defining derived class, Types of derivations-Public, Private and Protected, Types (Forms) of Inheritance- Single, Multi-level, Multiple, Hierarchical, Hybrid, Multi- path (Virtual base class), Behavior of constructors and destructor in inheritance, Overloaded member functions, Pointer to base class, Pointer to derived class, Object composition-delegation

Runtime polymorphism: Introduction to runtime polymorphism, Virtual functions-Characteristics, Use of virtual function, Pure virtual function-Characteristics, Use, Abstract class, virtual destructors

Unit II: **[18]**

Stream and Files: Introduction to streams in C++, Stream classes, File stream classes, Formatted and unformatted I/O functions and Manipulators.

File Manipulations: Opening, closing, reading, writing, Appending, File opening modes-Opening files, using open() and constructor, Error handling during file manipulations, Command line arguments.

Exception Handling and Template: Introduction to Exception handling, Exception handling mechanism-try, catch, throw keywords, Custom exception.

Introduction to Function and Class template, inheritance of class template, class template containership.

Reference Books:

1. OOP in C++ – E-balagurusamy
2. Mastering C++ - K.R. Venugopal
3. Structured approach using C++ – Behrouz A. Forouzan
4. The Complete ReferenceC++- Fourth Edition. Herbert Schildt

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

Course Title: Practical based on DSC2-2

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L):4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

1. Write a program to implement single inheritance.
2. Write a program to implement multi-level inheritance
3. Write a program to implement multiple inheritance
4. Write a program to implement hierarchical inheritance
5. Write a program to implement hybrid inheritance
6. Write a program to implement multi-path inheritance
7. Write a program that shows use of pointer to base class
8. Write a program that shows use of pointer to derived class
9. Write a program that shows use of virtual function.
10. Write a program that shows use of pure virtual function.
11. Write a program that shows use of abstract class
12. Write a program that shows use of virtual destructor
13. Write a program that shows behavior of constructor and destructor in inheritance.

Streams and Files

1. Write a program that shows use of istream class.
2. Write a program that shows use of ostream class.
3. Write a program that shows use of different manipulators.
4. Write a program to read, write and append data into file.
5. Write a program that checks two files are identical or not.
6. Write a program that shows use of random access of file.
7. Write a program that shows use of command line argument.

Exception Handling and template

1. Write a program that shows use try, catch and throw
2. Write a program that shows use multiple catch blocks.
3. Write a program that shows use of custom exception.
4. Write a program that shows use of function template
5. Write a program that shows use of class template

Type.: DSC3-2

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

Course Title: Python - II

(Paper Code: Paper- VIII)

Credits: Theory – (2)	**Practical – (2)**
Total Lectures: 30 Hrs.	**Contact Hrs. (L):** 2
University Evaluation: 30 Marks	**Internal Evaluation:** 20 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 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 an 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)**

Modules and packages: Introduction to modules and packages, import statement, from…import statement, creating our own modules, working with built-in modules- Math module, time module and random module.

Functions: Difference between function and method, defining a function, calling function, returning result from a function, returning multiple values from a 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: the 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.

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

Course Title: Practical based on DSC3-2

Credits: Practical's – (2)

Total Lectures: 60 Hrs. **Contact Hrs. (L):4**

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

1) Write a function to return the addition and subtraction of two numbers using a function that return two values.
2) Write a Python program to demonstrate the different methods of array modules.
3) Write a Python program to demonstrate the types of array.
4) Write a Python program to understand the positional arguments of a function
5) Write a Python program to understand the keyword arguments of a function
6) Write a Python program to understand the default arguments in a function
7) Write a Python program to understand Variable length arguments in a function.
8) Write a Python program to understand the Anonymous (lambda) Function.
9) Write a Python program to understand local, non-local and global variables.
10) Write a Python program to create a module and import it.
11) Write a Python program to create a package and import it.
12) Write a Python program to demonstrate the instance method, class method and static method.
13) Write a Python program to demonstrate inner classes.
14) Write a Python program to demonstrate Constructors in Inheritance.
15) Write a Python program to demonstrate method overloading.
16) Write a Python program to demonstrate method overriding.
17) Write a Python program to read all the strings from the text file and display them.
18) Write a Python program to append data to an existing file and display them.
19) Write a Python program to count a number of lines, words and characters in a file.
20) Write a Python program to copy an image file into another file.
21) Write a Python program to apply different manipulation operations of directories.
22) Write a Python program to handle the ZeroDivisionError exception.
23) Write a Python program to handle syntax errors given by the eval( ) function.
24) Write a python program to handle the IOError produced by the open( ) function.
25) Write a python program to illustrate the use of raising an exception

Type: SEC-2

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

Course Title: Advanced Web Designing

Credits: Theory – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):**2

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objective:

1. To identify the capabilities of JavaScript and jQuery and their role in web design and the document object model.
2. To respond to user events using jQuery, creating interactivity.
3. To provide a collection of syntax for template designs.
4. To develop and apply appropriate website or web application information architectures.
5. To design effective user interfaces.

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

1. Understand the concepts of jQuery, bootstrap.
2. Build interactive web applications using JQuery and bootstrap.
3. Develop solution to complex problems using appropriate method, technologies, frameworks, web services and content management
4. Extend this knowledge to .Net Platforms,Java Technologies,Full Stack Development .

Unit I: **(12)**

JQUERY: Introduction to jQuery, Difference between document Ready and Window Load, Adding jQuery to Your Web Pages, jQuery Syntax, jQuery Selectors, jQuery Event Methods, Effects-Hide and Show, Fading, Sliding, Animation, Button in jQuery, How to handle forms, Callback Functions, Chaining, Get and Set Content and Attributes, Add Elements, Add Several New Elements, Remove Elements, Get and Set CSS Classes, css() Method, The noConflict() Method

Unit II: **[18]**

BOOTSTRAP: Introduction to Bootstrap Framework, History of Bootstrap, Advantages, Responsive web page, Major Features of Bootstrap, Setting up Environment, Syntax of

Bootstrap, Introduction to Bootstrap container and its types, Introduction to Bootstrap Grid, **Bootstrap Forms:** Form Layouts, Inline Form, form controls (input, textarea, checkbox, radio, select), Static Control, Media Objects, Filters.

Bootstrap Components: Jumbotron, Button, Grid, Table, Alert , Wells , Badge and label, Panels , Input Groups, Input Types, Modals, Popover, Scrollspy, Pagination, Pager , Image , Glyphicon , Carousel , Progress Bar , List Group , Dropdown , Collapse, Tabs/Pill, Navbar.

Reference Books:

1. Nicholas C Zakas, “Professional JavaScript for Web Developers”, 3rd Edition, Wrox/Wiley India, 2012.
2. David Sawyer Mcfarland, “JavaScript & jQuery: The Missing Manual”, 1st Edition, O’Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014.
3. Benjamin Jakobus and Jason Marah, Mastering Bootstrap 4, Packt Publishing, 2016.
4. Jacob Lett, Bootstrap 4 Quick Start: Responsive Web Design and Development Basics for Beginners, 2018.

Type: GE / OE -1

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

Course Title: Computational Mathematics

Credits: Theory – (2)

Total Lectures: 30 Hrs. **Contact Hrs. (L):** 2

University Evaluation: 30 Marks **Internal Evaluation:** 20 Marks

Course Objectives:

1. To introduce the concepts of mathematical logic.
2. To introduce the concepts of relations, counting and functions.
3. To perform the operations associated with counting, functions, and relations.
4. To use Graph Theory for solving problems.
5. To introduce the matrix and its operations.

Course Outcomes:

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

1. Ability to apply mathematical logic to solve problems.
2. Understand sets, relations, counting, matrix, and graph.
3. Able to use logical notation to define and reason about fundamental mathematical concepts such as relations, counting, matrices and graphs.
4. Able to model and solve real-world problems using graphs and trees.

Unit I **(15)**

Logic and Proofs: Propositional logic, Applications of Propositional logic, propositional equivalences, Predicates and Quantifiers, Rules of inference.

Relations: Relations and their properties, Representing relation, Closures of relations, Partial orderings .

Unit II **(15)**

Counting: The basics of counting, The pigeonhole principle, Permutation and Combinations, Applications of recurrence relations, Solving recurrence relations, Divide and Conquer algorithms and recurrence relations.

Graphs: Graphs and Graphs models, Graph terminology and special types of graphs, Representing graphs and Graph isomorphism, Connectivity, shortest path problems.

Matrices: Introduction, operations, inverse, Rank of a matrix, solution of simultaneous linear equations, Eigen values and Eigen Vectors.

Reference Books:

1. Modern Algebra - S. Arumugam and A. Thangapandi Isaac, Scitech publications, 2005.
2. Invitation to Graph Theory- S.Arumugam and S.Ramachandran, Scitech Publications, 2005, Chennai.
3. Discrete Mathematical Structures with applications to Computer Science Tremblay and Manohar, McGrawHill, 1997.
4. Mathematical Structure for Compute Sience, Discrete Mathematics and its Appications, Judith L.Gersting, W.H.Freeman and Company, Seventh Edition, 2014.

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

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

(NEP-2020)

Semester-I		
Sr. No.	**Name of the Old Paper (w.e.f. 2022-2023)**	**Name of the New Paper (w.e.f. 2043-2025)**
1	English Paper I Part-A (communication skill)	English (Sem-I)
2	Fundamental of Computer	No Equivalence
3	Basics of Operating System	Basics of Operating System (Sem-II)
4	Programming using 'C'	No Equivalence
5	Python – I	Python – I
6	Numerical Methods	Computational Mathematics
7	Graph Theory	No Equivalence
8	Basic Electronics	No Equivalence
9	Advanced Electronics	No Equivalence
Semester-II		
Sr. No.	**Name of the Old Paper (w.e.f. 2022-2023)**	**Name of the New Paper (w.e.f. 2024-2025)**
1	English Paper I Part-B (communication skill)	English (Sem-II)
2	Introduction to Web Technology	Basics of Web Designing

3	Operating System	No Equivalence
4	Object Oriented Programming using C++	OOP'S with C++-I
5	Python – II	Python – II
6	Linear Algebra	No Equivalence
7	Discrete Mathematics	No Equivalence
8	Digital Electronics and Microprocessor	Digital System
9	Introduction to Microcontroller and Embedded System	No Equivalence

CA

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology.

Nature of Question Paper for CBCS Pattern

B. Sc. / B.C.A.(Part- I) w.e.f. AY 2024-25

Time:

Total Marks: 20

Internal Evaluation System for 20 Marks

Choose any two of the following

Home Assignment / Unit Test / Tutorial /Seminar

Pattern of Examination

External Evaluation + Internal Evaluation

30 Marks + 20 Marks = 50 Marks

Passing Criteria:

1. Written Exam – 12 out of 30

2. Continuous Assessment (CA) – 08 out of 20