**Punyashlok Ahilyadevi Holkar Solapur University, Solapur**

पुण्यश्लोक अहिल्यादेवी होळकर
सोलापूर विद्यापीठ

।। विद्यया संपन्नता ।।

**NAAC Accredited-2022**
**‘B$^{++}$’Grade (CGPA2.96)**

# Name of the Faculty: Science and Technology

(As per New Education Policy 2020)

## Syllabus: - Entire Computer Science

## Name of the Program: B.Sc. I (ECS) (Sem. – I and II)

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

## Punyashlok Ahilyadevi Holkar Solapur University, Solapur

## B. Sc. (Entire Computer Science) - I

**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:**

- To develop problem-solving abilities using a computer.
- To build the necessary skill set and analytical abilities for developing computer-basedsolutions for real-life problems.
- To train students in professional skills related to Software Industry.
- To prepare the necessary knowledge base for research and development in Computer Science.
- 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:
- An ability to apply knowledge of mathematics, statistics and computer science in practice.
- An ability to enhance a comprehensive understanding of the theory and its applicationin diverse fields.
- 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
- 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.
- To enhance the programming skills of young IT professionals, the program hasintroduced the concept of project development in each language/technology learned during the curriculum.

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

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**

An examination of any other statutory University or an Examining Body is recognized as equivalent thereto. 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, bythe 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, and implementation

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

**PLO-4.** Ability to link knowledge of Computer Science with other two chosen auxiliarydisciplines 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 technicalemployment.

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

**PLO-8.** Ability to operate, manage, deploy, and configure computer network, hardware, andsoftware 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 softwareproject 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 therebyinnovate novel solutions to existing problems

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

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

## B.Sc. (ECS) Part - I

| **Subject/ Core Course** | | **Name and Type of the Paper** | | | **Hrs./week** | | | **Total Marks Per Paper** | **UA** | **CA** | **Credits** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Type** | **Code** | **Name** | **L** | **T** | **P** | | | | |
| **Sem-I** | | | | | | | | | | | |
| Major **(Select any one group)** | **Group - I** | DSC1-1 | Go6-0101 | OOP'S with C++-I | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0101-P | Practical based on DSC1-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC2-1 | Go6-0102 | Python Programming-I | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0102-P | Practical based on DSC2-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC3-1 | Go6-0103 | Digital Electronics and Microprocessors | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0103-P | Practical based on DSC3-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| | **Group - II** | DSC1-1 | Go6-0101 | OOP'S with C++-I | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0101-P | Practical based on DSC1-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC2-1 | Go6-0102 | Python Programming-I | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0102-P | Practical based on DSC2-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC4-1 | Go6-0104 | Numerical Methods | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0104-P | Practical based on DSC4-1 | - | - | 4 | 50 | 30 | 20 | 2 |
| Generic/ Open Elective Courses | | GE1/OE1 | G06-GE-OE-101 | Office Automation | 2 | - | - | 50 | 30 | 20 | 2 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SEC/ VSC | | SEC-1 | G06-SEC-101 | Introduction to Web Design | - | - | 2 | 50 | 30 | 20 | 2 |
| AES, IKS, VEC | | L1-1 | ENG-101 | English | 2 | - | - | 50 | 30 | 20 | 2 |
| | | IKS | G06-IKS-101 | To be selected from the Basket of IKS | 2 | - | - | 50 | 30 | 20 | 2 |
| | | VEC-1 | ICD-101 | Constitution of India | 2 | - | - | 50 | 30 | 20 | 2 |
| **Total** | | | | | **14** | | **14** | **550** | **330** | **220** | **22** |
| **Sem-II** | | | | | | | | | | | |
| Major **(SELECT ANY ONE GROUP)** | **GROUP -I** | DSC1-2 | Go6-0201 | OOP'S with C++- II | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0201-P | Practical based on DSC1-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC2-2 | Go6-0202 | Python Programming-II | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0202-P | Practical based on DSC2-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC3-2 | Go6-0203 | Introduction to Microcontroller and Embedded Systems | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0203-P | Practical based on DSC3-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| | **GROUP -II** | DSC1-2 | Go6-0201 | OOP'S with C++- II | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0201-P | Practical based on DSC1-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC2-2 | Go6-0202 | Python Programming-II | 2 | - | - | 50 | 30 | 20 | 2 |
| | | Practical | Go6-0202-P | Practical based on DSC2-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| | | DSC4-2 | Go6-0204 | Discrete Mathematics | 2 | - | - | 50 | 30 | 20 | 2 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Practical | Go6-0204-P | Practical based on DSC4-2 | - | - | 4 | 50 | 30 | 20 | 2 |
| Generic/ Open Elective Courses | | GE2/ OE2 | G06-GE-OE-201 | Software Engineering | 2 | - | - | 50 | 30 | 20 | 2 |
| SEC/ VSC | | SEC2 | G06-SEC-201 | Advanced Web Designing | - | - | 2 | 50 | 30 | 20 | 2 |
| AES, IKS,<br>VEC | | L1-2 | ENG-201 | English | 2 | - | - | 50 | 30 | 20 | 2 |
| | | VEC-2 | ENS24 | Environmental Studies | 2 | - | - | 50 | 30 | 20 | 2 |
| CC1 | | CC1 | CES-201 | Community Engagement and Services | 2 | - | - | 50 | 30 | 20 | 2 |
| **Total** | | | | | **14** | **-** | **14** | **550** | **330** | **220** | **22** |
| **Grand Total** | | | | | **28** | **-** | **28** | **1100** | **660** | **440** | **44** |

## Abbreviations:

| | | | | | |
|---|---|---|---|---|---|
| **L**: Lectures | **T**: Tutorials | **P**: Practical | | **UA** : University Assessment | **CA** : College Assessment |

| | | |
|---|---|---|
| Generic/ Open Electives: **GE/OE** | | Skill Enhancement Courses: **SEC** |
| Indian Knowledge System: **IKS** | | Ability Enhancement Courses: **AES** |
| Value Education Courses: **VEC** | | Vocational Skill and Skill Enhancement Courses: **VSEC** |

Co-curricular Courses: **CC**

**Student contact hours per week:** 24 Hours (Min.)

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

**Medium of instruction:** English

I. Practical Examination is the Semester after the 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 ComputerProgramming

| | |
|---|---|
| if he/she complete the courses equivalent to a minimum of 44 credits and an additional. 4 credits core NSQF course/Internship. | |
| **Course Structure:**<br>Lectures and Practicals should be conducted as per the scheme of lectures and practicals indicated inthe course structure. | |
| **Teaching and Practical Scheme**<br>I. Contact session for teaching 60 minutes each.<br>II. One Practical Batch should be of 20 students. | III. |
| **Assessment**<br>I. The final practical examination will be conducted by the University appointed examiners internalas well as external at the end of the semester for each lab course and marks will be submitted to the university by the panel.<br>II. The practical examination will be conducted semester-wise to maintain the relevance of the respective theory course with the laboratory course.<br>III. The final examinations shall be conducted at the end of the semester. | IV. |
| **Practical Examination:**<br>I. Each paper carries 30 Marks.<br>II. ***Duration of Practical Examination:*** 2 Hrs.<br>III. ***Nature of Question Paper:*** There will be four questions of 10 marks each. Students will attemptany two out of four questions.<br>IV. Certified Journal carries 5 Marks and Viva voce carries 5 Marks. | |
| **Standard of Passing:**<br>I. Minimum 12 marks in each subject. There shall be separate passing for theory (semester endexam and Internal) and practical also.<br>II. Admission to B.Sc. (Entire Computer Science) Part III is allowed only if a student has passed all the subjects of B.Sc. (Entire Computer Science) Part I. | |
| **Board of Paper Setters /Examiners:**<br>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

**Award of Class:**

***Grading:*** PAHS University has introduced a ten-point grading system as follows:

| **Sr. No.** | **Grade Abbreviation** | **From( Marks)** | **To (Marks)** | **Status** | **Grade Point** | **Description** |
|---|---|---|---|---|---|---|
| 1. | O | 80 | 100.00 | Pass | 10.00 | Excellent / Outstanding |
| 2. | $A^{+}$ | 70 | 79.99 | Pass | 9.00 | Very Good |
| 3. | A | 60 | 69.99 | Pass | 8.00 | Good |
| 4. | B+ | 55 | 59.99 | Pass | 7.00 | Fair |
| 5. | B | 50 | 54.99 | Pass | 6.00 | Above Average |
| 6. | C+ | 45 | 49.99 | Pass | 5.00 | Average |
| 7. | C | 40 | 44.99 | Pass | 4.00 | Below Average |
| 8. | F | 0 | 39.99 | Fail | 0.00 | Fail |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** |  |  |  |  |  |  |
| **Sem:** | I |  |  |  |  |  |
| **Paper Category:** | DSC1-1 **(Major)** |  |  |  |  |  |
| **Paper Name:** | OOPs with C++-I **Paper Code :** G06-0101 |  |  |  |  |  |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week |  |  |  |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Course Objective:**

Students will 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: Introduction to (Object Oriented Programming) OOP:** | **No. of Lectures:15** | **Weightage:**<br>**8-12 Marks** |
|---|---|---|

**Introduction:** Introduction to algorithm and flowchart, Introduction to OOP, Featuresof OOP’s- Class, Object, Data Abstraction and encapsulation, Data hiding, Message passing, polymorphism, inheritance, delegation, 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: Classes and Objects:** | **No. of Lectures:15** | **Weightage: 18-22 Marks** |
|---|---|---|

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 an object, Returning objects from functions, Passing object as a parameter by value, by pointer, by reference, Dynamic memory allocation (new, delete), Friend function and friend class, nesting of classes, Constructors, characteristics of a 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. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | Practical **(Major)** | | | | | |
| **Paper Name:** | Practical based on DSC1-1 | | | **Paper Code :** G06-0101-P | | |
| **Credit:** | 02 | | | **Practical:** | 4 Hrs./Week | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

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 parameter passing techniques in C++
3. Write a program that shows defining member function inside and outside of the class body and demonstrates the use of inline function
4. Write a program to implement the function overloading concept
5. Write a program to implement all types of constructors and destructor also demonstrate constructor overloading
6. Write a program that shows the use of static data member and static member function.
7. Write a program that shows the use of nesting classes.
8. Write a program that shows passing and returning an object from a function.
9. Write a program that shows explicit type conversion
10. Write a program to overload different unary and binary operators by using friend and member function.
11. Generate the result for 5 students with the following data - Name, exam no, Theory marks in5 subjects, grade(Use array of object concepts).
12. Write a program to demonstrate the friend function, friend class, member function of a classis friend to another class.
13. Write a program to count no. of objects created by using static data member & member function.
14. Write a program to overload unary operators (++, - -, -).
15. Write a program to overload the binary operator.(+, -, *, /, %) by using the member function andfriend function.

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | DSC2-1 **(Major)** | | | | | |
| **Paper Name:** | Python Programming-I | **Paper Code :** G06-0102 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**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: Basics of Python:** | **No. of Lectures:12** | **Weightage: 9-12 Marks** |

**Introduction:** Features of Python, Python Virtual Machine, Memory management, Garbage Collection, Installation of Python, setting the path to operating system environment, writing the first Python program, executing a Python program.

**Datatypes in Python**: Datatypes-Numeric, Sequence Type-String, List, Tuple, Boolean, Set, Dictionary, Binary Types, Type conversion- implicit and explicit, Python 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: Data Structure and Function:** | **No. of Lectures:18** | **Weightage: 18-21 Marks** |
|---|---|---|

**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.

**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:** Introduction, advantages of array, creating an array, types of arrays, importing array module, indexing and slicing on arrays, methods of array module.

**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. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | Practical **(Major)** | | | | | |
| **Paper Name:** | Practical based on DSC2-1 | **Paper Code :** G06-0102-P | | | | |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

1. Write a simple Python script that prints "Hello, World!" to the console.
2. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
3. Concatenate strings and convert between data types as needed.
4. Implement conditional statements (if-else, elif) to control program flow based on different conditions.
5. Use loops (for, while) to iterate over lists, tuples, dictionaries, and ranges.
6. Create nested loops and conditional statements for more complex control flow logic.
7. Create and manipulate lists, tuples, dictionaries, and sets.
8. Access elements in data structures using indexing and slicing operations.
9. Use list comprehensions and generator expressions for concise data manipulation.
10. Write a Python program to find the sum of a list of numbers using a for loop.
11. Write a Python program to display stars in right angled triangular form using nestedfor loops.
12. Write a Python program to display a multiplication table from 1 to 10 using nested for loops.
13. Write a Python program to display elements in a list in reverse order.
14. Write a Python program to accept elements in the form of a tuple and display theirsum and average.
15. Write a Python program to create a dictionary with employee details and retrieve the values upon giving keys.

<table>
<tr><th colspan="6">B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level</th></tr>
<tr><td colspan="2"><b>Sem:</b></td><td colspan="4">I</td></tr>
<tr><td colspan="2"><b>Paper Category:</b></td><td colspan="4">DSC3-1 <b>(Major) (Group - I)</b></td></tr>
<tr><td colspan="2"><b>Paper Name:</b></td><td colspan="4">Digital Electronics and Microprocessors <b>Paper Code :</b> G06-0103</td></tr>
<tr><td colspan="2"><b>Credit:</b></td><td>02</td><td colspan="2"><b>Theory:</b></td><td>2 Hrs./Week</td></tr>
<tr><td><b>Marks:</b></td><td><b>UA:</b> 30</td><td><b>CA:</b></td><td>20</td><td><b>Total:</b></td><td>50</td></tr>
</table>

**Course Objective:**

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:**

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

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:Logic Gates, Combinational and Sequential circuits:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

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: Introduction to Microprocessor:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

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

<table>
<tr><td colspan="7"><b>B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level</b></td></tr>
<tr><td colspan="2"><b>Sem:</b></td><td colspan="5">I</td></tr>
<tr><td colspan="2"><b>Paper Category:</b></td><td colspan="5">DSC3-1 <b>(Major) (Group - I)</b></td></tr>
<tr><td colspan="2"><b>Paper Name:</b></td><td colspan="5">Practical based on DSC3-1 <b>Paper Code :</b> G06-0103-P</td></tr>
<tr><td colspan="2"><b>Credit:</b></td><td colspan="2">02</td><td colspan="2"><b>Practical:</b></td><td>4 Hrs./Week</td></tr>
<tr><td><b>Marks:</b></td><td><b>UA:</b></td><td>30</td><td><b>CA:</b></td><td>20</td><td><b>Total:</b></td><td>50</td></tr>
</table>

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

1. Testing and implementation of Basic gates OR 7432, AND 7408, NOT 7404
2. Testing and implementation of universal gates NAND 7400, NOR 7402
3. Testing and implementation of EX-OR 7486 and EX-NOR 7486
4. Implementation of the basic gate using universal gate 7400 NAND gate
5. Implementation of the basic gate using universal gate 7402 NOR
6. Implementation of half adder using the basic gate
7. Implementation of half subtractor using the basic gate
8. Implementation of full adder using basic gate
9. Implementation of RS flip flop using NAND gate
10. Implementation of RS flip flop using NOR
11. Implementation of D-Flip Flop using IC 7474.
12. To write and execute ALP for the Addition of two, three, or multiple-byte numbers.
13. To write and execute ALP for Subtraction of two, three, or multiple-byte numbers.
14. To write and execute ALP for Multiplication of two, three, or multiple byte numbers.
15. To write and execute ALP for Division of 8 bit, 16 bit.
16. To write and execute assembly language program to arrange numbers in ascending order
17. To write and execute assembly language program to arrange numbers in

descending order

18. To write and execute assembly language program to transfer blocks of multiple byte number
19. To write and execute assembly language program for sorting of numbers.
20. To write and execute assembly language program for square of numbers.

| | | | | | |
|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | |
| **Sem:** | I | | | | |
| **Paper Category:** | DSC4-1 **(Major)** | | | | |
| **Paper Name:** | Numerical Methods **(Group - II)** **Paper Code :** G06-0104 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | |
| **Marks:** | **UA:** 30 | | **CA:** 20 | | **Total:** 50 |

**Course Objectives:**

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

**Course Outcomes:**

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

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

| **Unit-I: Polynomial Interpolation Approximation and Errors:** | **No. of Lectures:18** | **Weightage: 10-16 Marks** |
|---|---|---|

Polynomial Interpolation Approximation : 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, Divided Difference Operator , Divided difference table, 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: Matrices, Solution of System of linear equations And Solution of Non Linear Equations: | No. of Lectures:12 | Weightage:<br>14-20 Marks |
|---|---|---|

Matrices, Solution Of System of linear equations : Introduction, definition of a matrix and elementary results, types of matrices, operations on matrix, linear equations, system of linear equations, consistent system of linear equations, in consistent system of linear equations, ill and well-conditioned system of linear equations, solution of system of linear equations by using Jacobi's Iterative method and by using Gauss-Seidel iterative method, examples, characteristic polynomial, characteristic equation and characteristic roots of a matrix, Eigen values of a matrix( 2x2 and 3x3 ), examples

Solution of non-linear equations: Introduction, definition of polynomial equation of degree n, linear equations, transcendental equations, non- linear equations. Location of roots, Algorithms to find root of given non-linear equations by using Bisection method, Regula Falsi Method, Newton Raphson Method. Solution of nonlinear equations by using bisection, regula-falsi and Newton Raphson method, examples.

**Reference Books:**

1. Introductory Methods of Numerical Analysis -- S.S. Sastry(Prentice Hall)
2. Computer Oriented Numerical Methods. -- Rajaraman
3. Introduction to applied Numerical Analysis. -- C. Richard, W. Hamming.
4. Numerical Methods for Science and Engineering -- R. G. Stanton (Prentice Hall)
5. Computers and Numerical Methods by E. Balguruswamy, (TMH).
6. Numerical Methods for Scientific and Engineering Computation --- M. K. Jain, S. R. K. Iyengar and R. K. Jain, New Age International Publisher.

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | DSC4-1 **(Major)** | | | | | |
| **Paper Name:** | Practical based on DSC4-1 **(Group - II)** **Paper Code :** G06-0104-P | | | | | |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Choose one of the following languages to solve the assignment: Python / C++**

1. Write a Python program to prepare equally spaced data for the given arguments.
2. Write a Python program to check whether the given data is equally spaced or not.
3. Write a Python program to prepare Newton's forward difference table. (Both static and Dynamic programmers)
4. Write a Python program to prepare Newton's backward difference table. (Both static and Dynamic programmers)
5. Write a Python program that implements Newton's forward difference interpolation formula to interpolate a value at a given point using equally spaced data points.
6. Write a Python program that implements Newton's backward difference interpolation formula to interpolate a value at a given point using equally spaced data points.
7. Write a Python programme to prepare divided difference table for a set of given data points.
8. Write a Python programme that calculates the absolute error, relative error and percentage error between a true value and an approximate value.
9. Write a Python program that performs addition, subtraction, multiplication, and division of two floating-point numbers in normalized form.
10. Write a Python program to find addition and subtraction of two matrices.
11. Write a Python program to find multiplication of two matrices.
12. Write a Python program to solve system of linear equations by using Jacobi’s iterative method.
13. Write a Python program to solve system of linear equations by using Gauss Seidel iterative method.
14. Write a Python program to find the interval in which root of the equation lies.
15. Write a Python program to find one of the roots of given non linear equation by using Bisection method.

16. Write a Python program to find one of the roots of given non linear equation by using Regula Falsi or False Position method.
17. Write a Python program to find one of the roots of given non linear equation by using Newton Raphson Method..

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | GE1 / OE1 | | | | | |
| **Paper Name:** | Office Automation **Paper Code :** G06-GE-OE-101 | | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Course Objectives:**

1. To provide an in-depth training in use of office automation, internet and internet tools. The course also helps the candidates to get acquainted with IT.
2. To help the students to understand how to format, edit, and print text documents and prepare for desktop publishing.
3. To create various documents newsletters, brochures, making document using photographs, charts, presentation, documents, drawings and other graphic images.
4. To work with the worksheet and presentation software..

**Course Outcomes:**

1. At the end of this course, the student should be able to
2. Integrate both graphs and tables created in Microsoft Excel into a laboratory report in Microsoft Word.
3. Generate equations, sample calculations, and basic diagrams in Microsoft Word.
4. Input experimental data into Microsoft Excel.
5. Perform calculations in Microsoft Excel using both manually inputting formulas and built-in Functions.
6. Generate simple and effective tables and graphs to describe experimental data in Microsoft Excel.
7. Properly format and organize a formal laboratory report in Microsoft Word.

| **Unit-I: Introduction to Computer and MS-Word :** | **No. of Lectures:15** | **Weightage: 8-12 Marks** |
|---|---|---|

**Introduction to Computer:** Applications of Computer, Advantages of Computer, Characteristics of Computer, Hardware and Software, Block diagram of computer

**MS Word:** Working with Documents -Opening and Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page and setting Margins, Converting files to different formats, Importing and Exporting documents, Sending files to others, Using Tool bars, Ruler, Using Icons, using help.,

**Formatting Documents:** Setting Font styles, Font selection- style, size, color etc, Type face - Bold, Italic, Underline, Case settings, Highlighting, Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets and Numbering.

**Setting Page style:** Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border and Shading, Columns, Header and footer, Setting Footnotes and end notes – Shortcut Keys; Inserting manual page break, Column break and line break, Creating sections and frames, Anchoring and Wrapping, Setting Document styles, Table of Contents, Index, Page Numbering, date and Time, Author etc., Creating Master Documents, Web page. Creating Tables: Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and formula. , Drawing: Inserting Clip Arts, Pictures/Files etc.,

**Tools:** Word Completion, Spell Checks, Mail merge, Templates, Creating contents for books, Creating Letter/Faxes, Creating Web pages, Using Wizards, Tracking Changes, Security, Digital Signature. Printing Documents – Shortcut keys.

| **Unit-II: MS-Excel and MS Power point:** | **No. of Lectures:15** | **Weightage: 18-22 Marks** |
|---|---|---|

**MS Excel:** Spread Sheet and its Applications, Opening Spreadsheet, Menus - main menu, Formula, Editing, Formatting, Toolbars, Using Icons, Using help, Shortcuts, Spreadsheet types. Working with Spreadsheets- opening, Saving files, setting Margins, Converting files to different formats (importing, exporting, sending files to others), Spreadsheet addressing - Rows, Columns and Cells, Referring Cells and Selecting Cells – Shortcut Keys.

**Entering and Deleting Data:** Entering data, Cut, Copy, Paste, Undo, Redo, Filling Continuous rows, columns, Highlighting values, Find, Search and replace, Inserting Data, Insert Cells, Column, rows and sheets, Symbols, Data from external files, Frames, Clipart, Pictures, Files etc, Inserting Functions, Manual breaks.

**Setting Formula:** Finding total in a column or row, Mathematical operations (Addition, Subtraction, Multiplication, Division, Exponentiation), using other Formulae.

**Formatting Spreadsheets:** Labeling columns and rows, Formatting- Cell, row, column and Sheet, Category- Alignment, Font, Border and Shading, Hiding/ Locking Cells, Anchoring objects, Formatting layout for Graphics, Clipart etc., Worksheet Row and Column Headers, Sheet Name, Row height and Column width, Visibility - Row, Column, Sheet, Security, Sheet Formatting and style, Sheet background, Colour etc, Borders and Shading ,Shortcut keys., **Working with sheets:** Sorting, Filtering, Validation, Consolidation, and Subtotal.

**Creating Charts:** Drawing. Printing. Using Tools – Error checking, Formula Auditing, Creating and Using Templates, Pivot Tables, Tracking Changes, Security, Customization.

**MS Power point:** Presentation – Opening new presentation, Different presentation templates, setting backgrounds, selecting presentation layouts.

**Creating a presentation:** Setting Presentation style, Adding text to the Presentation. Formatting a Presentation: Adding style, Colour, gradient fills, Arranging objects, Adding Header and Footer, Slide Background, Slide layout. Adding Graphics to the Presentation-Inserting pictures, movies, tables etc into presentation, Drawing Pictures using draw.

**Adding Effects to the Presentation:** Setting Animation and transition effect. Printing Handouts, Generating Standalone Presentation viewer.

**Reference Books:**

1. Information Technology in Business: Principles, Practices, and Opportunities by James A Senn, Prentice Hall.
2. Technology and Procedures for Administrative Professionals by Patsy Fulton-Calkins, Thomson Learning.
3. Computer Fundamental MS Office-Including Internet and Web Technology: Anupama Jain, Avneet Mehra
4. The Complete Reference: Virginia Andersen, McGraw Hill
5. MS Office 2007 in a Nutshell: S. Saxena, Vikas Publications
6. MS-Office 2007 Training Guide: S. Jain, BPB Publications
7. Learning Computer Fundamentals, MS Office and Internet and Web Technology: D. Mai dasani. Reading, Vols. 1 and 2. Macmillan, 1975, Bhasker, W. W. S. and Prabhu, N. S.

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | I | | | | | |
| **Paper Category:** | SEC1 | | | | | |
| **Paper Name:** | Introduction to Web Design **Paper Code :** G06-SEC-101 | | | | | |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**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 the Cascading Style sheet.
4. Utilize the concepts of JavaScript.

**List of Assignments:**

| | |
|---|---|
| 1. | Write Html code to display “Welcome in Web Technology” message. |
| 2. | Write Html code to display different heading levels. |
| 3. | Write Html code to subscript and superscript. |
| 4. | Write Html code to implement all physical tags. |
| 5. | Write Html code to implement all logical tags. |
| 6. | Write Html code to implement <pre> tag. |

| | |
|---|---|
| 7. | Write Html code to implement <font> tags. |
| 8. | Write Html code to implement <address> tags. |
| 9. | Write Html code to implement <meta> tags. |
| 10. | Write Html code to implement lists. |
| 11. | Write Html code to implement hyperlink tag with target attribute. |
| 12. | Write Html code to implement link, alink and vlink attributes. |
| 13. | Write Html code to implement image as hyperlink. |
| 14. | Write Html code to implement image map. |
| 15. | Write Html code for display University examination time table. |
| 16. | Write Html code for display Railway time table. |
| 17. | Write Html code to collect student information. |
| 18. | Write CSS code implement ISS. |
| 19. | Write CSS code implement ESS. |
| 20. | Write CSS code implement lists. |
| 21. | Write CSS code implement class and ids. |
| 22. | Write CSS code implement links. |
| 23. | Write CSS code implement padding. |
| 24. | Write CSS code implement background and border. |
| 25. | Write CSS code implement 2D. |
| 26. | Write CSS code implement 3D. |
| 27. | Write CSS code implement animation. |
| 28. | Write JavaScript code for addition of any two numbers. |
| 29. | Write JavaScript code for subtraction of any two numbers. |
| 30. | Write JavaScript code for multiplication of any two numbers. |
| 31. | Write JavaScript code for division of any two numbers. |
| 32. | Write JavaScript code to calculate simple interest. |
| 33. | Write JavaScript code to calculate area and perimeter of circle. |
| 34. | Write JavaScript code to calculate area and perimeter of rectangle. |
| 35. | Write JavaScript code to calculate area and perimeter of triangle. |
| 36. | Write JavaScript code to calculate area and perimeter of square. |
| 37. | Write JavaScript code to exchange value of any two numbers. |

| | |
|---|---|
| 38. | Write JavaScript code to exchange value of any two numbers (without using third variable). |
| 39. | Write JavaScript code to implement alert. |
| 40. | Write JavaScript code to implement confirm. |
| 41. | Write JavaScript code to implement eval. |
| 42. | Write JavaScript code to find out given number is even or odd. |
| 43. | Write JavaScript code to find out given number is positive or negative. |
| 44. | Write JavaScript code to find out maximum number between any two numbers. |
| 45. | Write JavaScript code to calculate factorial of any number. |
| 46. | Write JavaScript code to calculate digit sum of any number. |
| 47. | Write JavaScript code to calculate face value of any number. |
| 48. | Write JavaScript code to find out given number is prime or not. |
| 49. | Write JavaScript code to find out given number is Armstrong or not. |
| 50. | Write JavaScript code to find out given number is palindrome or not. |
| 51. | Write JavaScript code to find out given number is perfect or not. |
| 52. | Write JavaScript code to find out given number is strong or not. |
| 53. | Write JavaScript code to find out given number is perfect or not. |
| 54. | Write JavaScript code to find out given number is strong or not. |
| 55. | Write JavaScript code for function without argument and without return value. |
| 56. | Write JavaScript code for function without argument and with return value. |
| 57. | Write JavaScript code for function with argument and without return value. |
| 58. | Write JavaScript code for function with argument and return value. |
| 59. | Write JavaScript code calculate string length. |
| 60. | Write JavaScript code to working form. |
| 61. | Write JavaScript code for validation handling. |
| 62. | Write JavaScript code for working with array. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | DSC1-2 **(Major)** | | | | | |
| **Paper Name:** | OOPS with C++-II | **Paper Code :** G06-0201 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**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 and 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: Inheritance and Runtime Polymorphism:** | **No. of Lectures:12** | **Weightage: 12-18 Marks** |
|---|---|---|

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

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

| Unit II: Stream and Files: | No. of Lectures:18 | Weightage: 12-18 Marks |
|---|---|---|

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 Reference C++- Fourth Edition. Herbert Schildt

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | Practical **(Major)** | | | | | |
| **Paper Name:** | Practical based on DSC1-2 | | | **Paper Code :** | G06-0201-P | |
| **Credit:** | 02 | | | **Practical:** | 4 Hrs./Week | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

1. Write a program to implement all types of inheritance.
2. Write a program that shows use of pointer to base class and derived class
3. Write a program that shows use of virtual function and pure virtual function.
4. Write a program that shows use of abstract class
5. Write a program that shows use of virtual destructor
6. Write a program that shows behavior of constructor and destructor in inheritance.
7. Write a program that shows use of istream and ostream class.
8. Write a program that shows use of different manipulators.
9. Write a program to read, write and append data into file.
10. Write a program that checks two files are identical or not.
11. Write a program that shows use of random access of file.
12. Write a program that shows use of command line argument.
13. Write a program that shows use multiple catch blocks.
14. Write a program that shows use of custom exception.
15. Write a program that shows use of function template and class template

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | DSC2-2 **(Major)** | | | | | |
| **Paper Name:** | Python Programming -II | **Paper Code :** G06-0202 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**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: Modules and packages** | **No. of Lectures:15** | **Weightage: 8-10 Marks** |
|---|---|---|

**Introduction:** 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.

**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, operator overloading, abstract classes, and interfaces.

| Unit II: Exception Handling and Threading: | No. of Lectures:15 | Weightage: 20-22 Marks |
|---|---|---|

**Threading:** Understanding threads, Class and threads, Creating Threads, Thread Synchronization, Treads Life cycle, multi-threading.

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

**File Input Output:** 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. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | Practical **(Major)** | | | | | |
| **Paper Name:** | Practical based on DSC2-2 | **Paper Code :** G06-0202-P | | | | |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

1. Write a Python program to create a module and import it.
2. Write a Python program to create a package and import it.
3. Write a Python program to demonstrate the instance method, class method and static method.
4. Write a Python program to demonstrate inner classes.
5. Write a Python program to demonstrate Constructors in Inheritance.
6. Write a Python program to demonstrate method overriding.
7. Write a Python program to demonstrate unary and binary operator overloading.
8. Write a Python program to read, write and append the data from the text file and display them.
9. Write a Python program to count a number of lines, words and characters in a file.
10. Write a Python program to demonstrate all inbuild exception.
11. Write a Python program to demonstrate user defined exception.
12. Write a python program to illustrate the use of raising an exception
13. Write a program for Thread Synchronization
14. Write a program to demonstrate multi-threading.
15. Write a program to demonstrate thread life cycle.

<table>
<tr><th colspan="7">B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level</th></tr>
<tr><td colspan="2"><b>Sem:</b></td><td colspan="5">II</td></tr>
<tr><td colspan="2"><b>Paper Category:</b></td><td colspan="5">DSC3-2 <b>(Major) (Group-I)</b></td></tr>
<tr><td colspan="2"><b>Paper Name:</b></td><td colspan="5">Introduction to Microcontroller and Embedded System<br><b>Paper Code :</b> G06-0203</td></tr>
<tr><td colspan="2"><b>Credit:</b></td><td colspan="2">02</td><td colspan="2"><b>Theory:</b></td><td>2 Hrs./Week</td></tr>
<tr><td><b>Marks:</b></td><td><b>UA:</b></td><td>30</td><td><b>CA:</b></td><td>20</td><td><b>Total:</b></td><td>50</td></tr>
</table>

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

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: Introduction to Microcontroller:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |

**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: Introduction to Embedded System:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

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

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | Practical **(Major) (Group-I)** | | | | | |
| **Paper Name:** | Practical based on DSC3-2 | **Paper Code :** G06-0203-P | | | | |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Software: Keil uVision IDE, Proteus for simulation, Any text editor**

1. Create a simple project in Keil MicroVision. Write, compile, and debug a basic program to control an LED. Simulate the program and analyze the results.
2. Download and install Flash Magic from the official website. Launch the Flash Magic application. Select the target microcontroller. Configure the communication settings (COM port, baud rate, etc.). Load the compiled HEX file generated from Keil MicroVision.
3. Write a program to demonstrate the use of internal memory, registers, and Special Function Registers (SFRs) in the 8051 microcontrollers.
4. Write and execute ALP to add, subtract, two 8 bit numbers.
5. Write and execute ALP to multiply, and divide two 8 bit numbers.
6. Write and execute ALP to implement logicaloperations like AND, OR, XOR on data.
7. Write a program to configure and use the ports of the 8051 microcontrollers.
8. Write a program to configure and use the timer/counter functionality of the 8051microcontrollers.
9. Write and test simple programs to use timers for delay generation.
10. Write a program to configure and use the serial port of the 8051 microcontrollers forserial communication.
11. Write simple programs to demonstrate data transfer instructionsusing the 8051-instruction set .
12. Write simple programs to demonstrate arithmetic, and logical instructions using the 8051-instruction set
13. Write a program to turn an LED on and off using a port pin of the 8051 microcontrollers.
14. Write a program to read input from one port and send output to another port using

the8051 microcontrollers.

15. Write a program to blink an LED at regular intervals using the timer functionality ofthe 8051 microcontrollers.
16. Write small assembly programs using different types of instructions.
17. Write a program to send data from the 8051 to a PC via the serial port.
18. Write a program to display numbers 0-9 on a 7-segment display.
19. Implement a programto create a simple counter using the 7-segment display.
20. Write a program to display characters and strings on an LCD. Create a program to scrolltext on the LCD.
21. Write a program to read temperature data from the sensor. Display the temperature onan LCD.
22. Write a program to rotate the stepper motor in different directions. Implement speedcontrol for the stepper motor.

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | DSC4-2 **(Major) (Group-II)** | | | | | |
| **Paper Name:** | Discrete Mathematics | **Paper Code :** G06-0204 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Course Objective:**

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

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

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 effective algebraic techniques to analyze basic discrete structures and algorithms.
4. To understand asymptotic notation, and its significance, and be able to use it to analyze 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. Course Objectives:

| **Unit I: Relations and Functions:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

**Set and Function:** 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: Recurrence Relations and Counting principles:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

**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

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** |  |  |  |  |  |  |
| **Sem:** | II |  |  |  |  |  |
| **Paper Category:** | Practical **(Major) (Group-II)** |  |  |  |  |  |
| **Paper Name:** | Practical based on DSC4-2 | **Paper Code :** G06-0204-P |  |  |  |  |
| **Credit:** | 02 | **Practical:** | 4 Hrs./Week |  |  |  |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Choose one of the following languages to solve the assignment: Python, R, or MATLAB.**

1. Implement functions to check if one set is a subset of another.
2. Write functions to generate the power set of a set using iterative or recursive methods.
3. Create functions to perform union, intersection, complement, set difference and symmetric difference between of two sets.
4. Implement functions to verify Commutative, Distributive, and Associative Laws properties for set operations.
5. Write functions to validate DeMorgan’s laws for sets.
6. Develop algorithms to determine if two sets are disjoint.
7. Implement a function to calculate the cardinality (number of elements) of a set.
8. Write functions to compute the Cartesian product of two sets.
9. Implement functions to classify relations is Void, universal, identity, reflexive, symmetric, transitive, Equivalence, antisymmetric, partial ordering, and asymmetric.
10. Implement functions for matrix operations like addition and multiplication specific to relation matrices.
11. Use Matplotlib or similar libraries to visualize relations as directed graphs (digraphs).
12. Write functions to Calculate indegree and outdegree for vertices in a digraph.
13. Implement Warshall's algorithm to find the transitive closure of a relation.
14. Implement examples to illustrate the pigeonhole principle in counting.
15. Write functions to apply the addition and multiplication principle for counting sets.
16. Develop functions to solve problems using inclusive-exclusive principles for two sets and three sets.
17. Solve advanced problems involving combinations, permutations, and constraints using counting principles.

| | | | | | | |
|---|---|---|---|---|---|---|
| **B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level** | | | | | | |
| **Sem:** | II | | | | | |
| **Paper Category:** | GE2/OE2 | | | | | |
| **Paper Name:** | Software Engineering | **Paper Code :** G06-GE-OE-201 | | | | |
| **Credit:** | 02 | **Theory:** | 2 Hrs./Week | | | |
| **Marks:** | **UA:** | 30 | **CA:** | 20 | **Total:** | 50 |

**Course Objectives –**

The aim of this course is to prepare learners for a foundational understanding of computers, encompassing:

1. To understand the fundamental concepts and characteristics of systems.
2. To categorize different types of systems and understand system analysis.
3. To explore various SDLC models and their applications with different functional and non-functional requirements.
4. To learn various techniques for gathering system requirements.
5. To understand the process of designing and implementing a system.
6. To understand coding standards, size measures, complexity analysis, and verification.
7. To grasp the fundamentals of software testing.
8. To learn about software implementation and the maintenance process.

**Course Outcomes-**

1. Students will be able to define a system, identify its characteristics, and describe various types of systems.
2. Students will be able to describe and compare models such as the Waterfall model, V-shape model, Spiral model, and Prototyping, Incremental, RAD, and Agile methodologies.
3. Students will be able to identify, document, and analyze user and system requirements.
4. Students will be proficient in conducting interviews, questionnaires, record reviews, and observations for requirement gathering.
5. Students will be able to design data flow diagrams, entity-relationship diagrams, structured charts, and create a data dictionary. They will also learn input and output design.

6. Students will learn to write clean, maintainable code, estimate effort and cost, and verify code correctness.
7. Students will be able to perform different types of testing and understand testing methodologies.
8. Students will understand the steps involved in implementing software and the different types of software maintenance.

| **Unit I**: **- Introduction to Software Engineering:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

**System concepts:** Introduction system, characteristics, Elements of system, Types of system, System Analysis, Role of System Analyst, Software Engineering: Definition, Characteristics of software, Qualities of software. System Development life cycle: Waterfall model, V-shape model, Spiral model, Prototyping, incremental, RAD, Agile. Software requirements: Functional, Non-functional requirements, User requirements, System requirements, Fact-finding techniques: Interviews, Questionnaires, Record reviews, Observation Analysis and Design Tools: Flowcharting, Decision tables, Decision Trees, Structured English, Structure charting Techniques.

| **Unit II**: **- System Design and Implementation, Maintenance:** | **No. of Lectures:15** | **Weightage: 12-18 Marks** |
|---|---|---|

Data flow Diagram (Physical, Logical), Entity relation diagram, structured chart, Data Dictionary, Input and output design

Coding: Verification, size measures, complexity analysis, coding standards, Effort Estimation, Cost Estimation, Testing and its types, need

Construction of the system: traditional and incremental approaches, conversion methods, Software Implementation, Overview of the maintenance process, and types of maintenance.

**Reference Books-**

1. Analysis and Design of Information Systems By James Senn.
2. Practical guide to structure System Design By Miller/Page/jones.
3. Software Engineering By Pressman.
4. System Analysis and Design By Parthsarty

<table>
<tr><th colspan="7">B.Sc. (Entire Computer Science)-I, Level - 4.5 UG Certificate Level</th></tr>
<tr><td colspan="2"><b>Sem:</b></td><td colspan="5">II</td></tr>
<tr><td colspan="2"><b>Paper Category:</b></td><td colspan="5">SEC2</td></tr>
<tr><td colspan="2"><b>Paper Name:</b></td><td colspan="5">Advanced Web Designing <b>Paper Code :</b> G06-SEC-201</td></tr>
<tr><td colspan="2"><b>Credit:</b></td><td colspan="2">02</td><td colspan="2"><b>Practical:</b></td><td>4 Hrs./Week</td></tr>
<tr><td><b>Marks:</b></td><td><b>UA:</b></td><td>30</td><td><b>CA:</b></td><td>20</td><td><b>Total:</b></td><td>50</td></tr>
</table>

**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 and Bootstrap.
2. Build interactive web applications using JQuery and bootstrap.
3. Develop solutions to complex problems using appropriate methods, technologies, frameworks, web services and content management
4. Extend this knowledge to .Net Platforms, Java Technologies, and Full Stack Development.

**List of Assignments:**

| | |
|---|---|
| 1. | Hide and Show Elements. |
| 2. | Change CSS Dynamically |
| 3. | Fade In and Fade Out |
| 4. | Slide Up and Slide Down |
| 5. | Animate an Element |
| 6. | Form Validation |
| 7. | Append and Remove Elements |
| 8. | Handle Events |
| 9. | AJAX Request |

| | |
|---|---|
| 10. | Chaining Methods |
| 11. | Basic Grid Layout |
| 12. | Basic Grid Layout |
| 13. | Cards |
| 14. | Forms |
| 15. | Buttons |
| 16. | Modals |
| 17. | Alerts |
| 18. | Carousel |
| 19. | Tooltips |
| 20. | Progress Bar |
| 21. | Tables |
| 22. | List Group |
| 23. | Create a Website with Bootstrap Grid |

### 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. 2023-2024)** |
| 1. | Fundamental of Computer | No Equivalence |
| 2. | Basics of Operating System | No Equivalence |
| 3. | Programming using 'C' | No Equivalence |
| 4. | Python – I | Python Programming – I |
| 5. | Numerical Methods | Numerical Methods |
| 6. | Graph Theory | No Equivalence |
| 7. | Basic Electronics | No Equivalence |
| 8. | 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. 2023-2024)** |
| 1. | Introduction to Web Technology | Introduction to Web Design |
| 2. | Operating System | No Equivalence |
| 3. | Object Oriented Programming using C++ | OOP'S with C++-I (Sem-I) |
| 4. | Python – II | Python Programming – II |
| 5. | Linear Algebra | No Equivalence |
| 6. | Discrete Mathematics | Discrete Mathematics |
| 7. | Digital Electronics and Microprocessor | Digital Electronics and Microprocessor (Sem-I) |
| 8. | Introduction to Microcontroller and Embedded System | Introduction to Microcontroller and Embedded System |