॥ विद्या विनयेन शोभते ॥

Janardan Bhagat Shikshan Prasarak Sanstha's
CHANGU KANA THAKUR
ARTS, COMMERCE AND SCIENCE
COLLEGE, NEW PANVEL (AUTONOMOUS)

Re-accredited 'A+' Grade by NAAC

'College with Potential for Excellence' Status Awarded by UGC 'Best College Award' by University of Mumbai

Program: B.Sc.

Revised Syllabus of F.Y.B.Sc. Computer Science

Choice Based Credit & Grading System (60:40)

w.e.f. Academic Year 2023-24 as per NEP 2020

Programme Outcomes

After completion of B.Sc. Programme students will acquire

Sr. No.	**After completion of B.Sc. program students will acquire**	**Graduate Attribute**
PO1	The knowledge of the disciplines and in-depth and extensive Disciplinary knowledge, understanding and skills in a specific field of interest.	Disciplinary knowledge
PO2	An ability to develop and conduct experiments, analyze, and Interpret data and use scientific judgment to draw conclusions.	Scientific reasoning
PO3	An ability to use current technology, and modern tools necessary for creation, analysis, dissemination of information.	Digital literacy
PO4	Innovative, professional, and entrepreneurial skills needed in various disciplines of science.	Life-long learning
PO5	An ability to achieve high order communication skills.	Communication skills
PO6	An ability to collect, analyze and evaluate information and ideas and apply them in problem solving using conventional as well as modern approaches.	Problem solving
PO7	A sense of social responsibility; intellectual and practical skills and demonstration of ability to apply it in real-world settings.	Reflective thinking
PO8	An ability to engage in independent and life-long learning through openness, curiosity, and a desire to meet new challenges.	Life-long learning
PO9	A capacity to relate, collaborate, and lead others, and to exchange views and ideas to work in a team to achieve desired outcomes.	Teamwork
PO10	An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	Leadership
PO11	An ability to understanding values, ethics, and morality in a multidisciplinary context.	Moral and ethical awareness.

Preamble

Educators and stakeholders of higher education in Computer Science, inspired by the principles of the National Education Policy (NEP), strive to create a comprehensive and transformative learning experience for Bachelor of Science students in Computer Science. This program is designed to nurture critical thinking, creativity, problem-solving skills, and a deep understanding of computer science concepts and applications.

Recognizing the dynamic nature of the field of computer science and its immense impact on society, our goal is to equip students with a strong foundation in theoretical concepts, practical skills, and ethical considerations. We aim to empower them to contribute meaningfully to the rapidly evolving digital landscape and address the challenges and opportunities presented by emerging technologies.

The F.Y.B.Sc. Computer Science program is built on the principles of interdisciplinary learning, fostering connections between computer science and other domains such as mathematics, statistics, engineering, and social sciences. This interdisciplinary approach enables students to explore the breadth and depth of computer science while developing a holistic understanding of its relationship with various fields of knowledge.

Our curriculum emphasizes hands-on experiential learning, where students engage in project-based assignments, laboratory work, and real-world applications. They are encouraged to collaborate, innovate, and think critically, preparing them for the ever-changing demands of the industry and research in computer science. Additionally, the program promotes a culture of lifelong learning, encouraging students to stay abreast of the latest developments and contribute to the advancement of the field.

We recognize the importance of ethical considerations in computer science and emphasize the ethical use of technology, data privacy, and security. Our curriculum integrates ethics and social responsibility, enabling students to understand the impact of their work on individuals, communities, and society at large. We believe that responsible computing is an integral part of a computer scientist's role and encourage our students to be ethical leaders in their respective fields.

We strive to create an inclusive and diverse learning environment, fostering a sense of belonging for all students. The program encourages equal participation and representation, irrespective of gender, caste, creed, or socio-economic background. We are committed to promoting diversity, equity, and inclusion, recognizing that diverse perspectives enhance creativity, innovation, and problem-solving.

By aligning with the National Education Policy, we envision that the F.Y.B.Sc. Computer Science program will empower students to become competent professionals, entrepreneurs, researchers, and leaders in the field of computer science. We aim to cultivate a passion for

Lifelong learning, instill ethical values, and foster a deep appreciation for the transformative power of technology in addressing the challenges of the present and the future.

With this vision, we embark on this journey, empowering our students to shape the world through their knowledge, skills, and commitment to the highest standards of excellence and integrity in the field of computer science.

Additionally, we extend our gratitude to the members of the Board of Studies (BoS) for their valuable contributions in shaping the curriculum of the B.Sc. Computer Science program. Their expertise and insights have played a crucial role in ensuring that the syllabi align with the principles of the National Education Policy and meet the evolving needs of the industry and academia. We appreciate their dedication and commitment to maintaining academic excellence and fostering innovation in computer science education.

Scheme of Examination

- **Revised Scheme of Examination**

The performance of the learners shall be evaluated into two components. The learner's Performance shall be assessed by Internal Assessment with 40% marks in the first component by conducting the Semester End Examinations with 60% marks in the second component. The allocation of marks for the Internal Assessment and Semester End Examinations are as shown below:-

A) Internal Assessment: 40 **%** **40 Marks**

Sr. No.	Particular	Marks
01	One periodical class test / online examination to be conducted in the given semester	20 Marks
02	Test on Practical Skills/ Case studies /Group/ Individual Survey Project/Presentation and write up on the selected topics of the subjects/ Test based on tutorials /Book Review / Open Book Test	15 Marks
03	Active participation in routine class instructional deliveries and overall conduct as a responsible learner, mannerism and articulation and exhibit of leadership qualities in organizing related academic activities	05 Marks

Question Paper Pattern

(Periodical Class Test)

Maximum Marks: 20

Duration: 40 Minutes

Questions to be set: 02

All Questions are Compulsory

Question No.	Particular	Marks
Q-	Multiple Choice Questions and Match the following	20 Marks

B) Semester End Examination: 60 % 60 Marks

- **Undergraduate Programmes of F. Y. B.Sc. (Sem. I & II)**
  - Duration: The examination shall be of 2 hours duration.

Question Paper Pattern

Theory question paper pattern
1. There shall be four questions of 15 marks each (30 marks with internal options). 2. On each unit there will be one question and fourth question will be based on entire syllabus. 3. All questions shall be compulsory with internal options. 4. Question may be subdivided into sub-questions a, b, c… and the allocation of marks depends on the weightage of the unit.

- **Passing Standard**

The learners to pass a course shall have to obtain a minimum of 40% marks in aggregate for each course where the course consists of Internal Assessment and Semester End Examination. The learners shall obtain minimum of 40% marks (i.e. 16 out of 40) in the Internal Assessment and 40% marks in Semester End Examination (i.e. 24 Out of 60) separately, to pass the course and minimum of Grade D, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Internal Assessment and Semester End Examination together.

I. **Practical Examination :** – 150 (50 marks x 3 papers)

II. **Each core subject carries:** - 50 Marks

Sr. No.	**Particulars of External**	**Marks**
1	Laboratory Work	40
2	Journal	05
3	Viva	05
	TOTAL	50

Minimum 75 % practical from each core subject are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Exam) ---------

Semester - I *[Under CBCS Scheme]*

Course Type	Course	Course Code	Credits	Hrs ./week	Internal assessment and Semester - end examination	Total
DSC	Computer Organization Design and Architecture	UCS1COD	2	2	40+60	100
	Programming with Python I	UCS1IPP	2	2	40+60	100
	Practical of Computer Organization Design and Architecture + Programming with Python I	UCS1PR1	1+1	2+2	50+50	100
VSC	Statistics for Data Science	UCS1SDS	1	1	40+60	100
	Practical of Statistics for Data Science	UCS1PR2	1	2	50	50
SEC (Anyone from the List)	Web Designing	UCS1WEB	1	1	20+30	50
	Programming with C	UCS1PWC				
SEC	Practical of SEC	UCS1PR3	1	2	50	50
AEC (Anyone from	Communication Skill in English Hindi/Marathi		2	2	40+60	100

the List)						
VEC	● Environmental Science ● Understanding India ● Digital and Technological Solution		2	2	40+60	100
IKS	History and Evolution of Computing	UCS1IKS	2	2	40+60	100
CCC	Foundation Course in NSS/NCC/PE/PA		2	2	40+60	100
OE	● Introduction to Accountancy-II (BAF) ● Introduction to Taxation-II (BAF) ● Personality Development-II (BMS) ● Business Administration-II (BMS) ● Organic Framing (BIOT) ● Bio-Entrepreneurship (BIOT)		4	4	50	100
Total Credits			22			

Semester - II *[Under CBCS Scheme]*

Course Type	Course	Course Code	Credits	Hrs./week	Internal assessment nt and Semester - end examina ion	Total
DSC	Relational Database System	UCS2RDS	2	2	40+60	100
	Programming with Python II	UCS2APP	2	2	40+60	100
	Practical of Database Management System + Programming with Python II	UCS2PR1	1+1	2+2	50+50	100
Minor	Artificial Intelligence	UCS2ARI	2	2	40+60	100
VSC	Linear Algebra: Applications in Computer Science	UCS2LAR	1	1	40+60	100
	Practical’s of Linear Algebra: Applications in Computer Science	UCS2PR2	1	2	50	50
SEC (Anyone from the List)	PHP Programming	UCS2PHP	1	1	20+30	50
	Programming with C++	UCS2CPP	1		20+30	50

SEC	Practical of SEC	UCS2PR2	1	2	50	50
AEC (Anyone from the List)	● Communication Skill in English ● Hindi/Marathi		2	2	40+60	100
VEC (Anyone from the List)	● Environmental Science ● Understanding India ● Digital and Technological Solution		2	2	40+60	100
CCC	Foundation Course in NSS/NCC/PE/PA		2	2	40+60	100
OE	● Introduction to Accountancy-II (BAF) ● Introduction to Taxation-II (BAF) ● Personality Development-II (BMS) ● Business Administration-II (BMS) ● Organic Framing (BIOT) ● Bio-Entrepreneurship (BIOT)		4	4	50	100
Total Credits			22			

SEMESTER I

Paper I COMPUTER ORGANIZATION DESIGN AND ARCHITECTURE

Course Description	
Semester	I
Course Name	Computer Organization Design and Architecture
Course Code	UCS1COD
Credit	2
Theory	2 hrs. per week
Practical	2 hrs. per week
Hours	30

Course Objectives

1. To understand the basic structure and organization of computers
2. To analyze the structure and working of digital circuits
3. To understand the memory organization in computer architecture

Course Outcomes

Learners will be able to:

1. Explain the underlying principles of computers.
2. Analyze the Instruction set architecture.
3. Analyze the role of various hardware components of processor.

Syllabus		**30 L**
Unit I	**Fundamentals of Computers:** Basic structure and operation of computer, functional units and their interaction, High level ,Machine level & Assembly language	15 L

	Data representation and Computers Arithmetic: Decimal, Binary, Octal, Hexadecimal Number system and their interconversion, BCD code, Gray code, Excess-3 code, ASCII , EBCDIC, Unicode, Concept of parity code. Signed and Unsigned numbers, 1’s, 2’s, 9‘s and 15’s complement of binary numbers, Binary arithmetic (Addition, subtraction and subtraction using1’s complement and 2’s complement), Boolean Algebra, K-MAP. **Memory Organization:** Memory System Overview, Memory Design, Cache Memory, Internal Memory, External Memory, Virtual Memory	
Unit II	**Digital Circuits :** **Combinational Circuit:** Logic Gates, Half & Full adder, Half & Full subtractor, Multiplexer (up to 4 to 1 MUX), and Demultiplexer (up to 1 to 4 DEMUX), Encoder(Decimal to BCD encoder and 3 bit priority encoder), Decoder(3 to 8 line decoder using gates only) , Magnitude comparator **Sequential Circuit:** Concept of sequential circuits; Latch, Flip-flops: RS, clocked RS, JK, T, D, Counter –(types: synchronous, asynchronous), shift register **CPU and Processor Organization:** Block diagram of CPU, functions of CPU, general register organization, stack organization (operation of stack, types of stack, register stack and memory stack), block diagram of ALU, Processor bus organization, arithmetic logic unit (ALU) instruction formats, addressing modes, data transfer and manipulation, program control, microprocessor organization, RISC & CISC Architectures, Introduction to microprocessor and microcontroller	15 L
Text Books	1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012 2.Microprocessors and Microcontrollers: Architecture, Kant Krishna 3. Computer Architecture, Berhooz Parhami	
Reference Books	1. Computer Organization and Design, 4th Ed, D. A. Patterson and J. L. Hennessy 2. Microprocessor Architecture, Jean Loup Baer	

Sr. No	Practicals of Computer Organization Design & Architecture
1	Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).
2	Simplify given Boolean expression and realize it.
3	Design and verify a half/full adder.
4	Design and verify half/full subtractor.
5	Design a 2-bit magnitude comparator using combinational circuits.
6	Design and verify the operation of flip-flops using logic gates.
7	Verify the operation of a counter.
8	Verify the operation of a 4-bit shift register.
9	Using SPIM, write the program to do the addition, subtraction of two numbers.
10	Using an 8085 simulator, perform the sum of two 8 bit numbers.
11	Using an 8085 simulator, perform the sum of two 16 bit numbers.
12	Using an 8085 simulator, perform the multiplication of two 8 bit numbers.
13	Using an 8085 simulator, perform subtraction of two 8-bit numbers with or without borrow.
14	Using an 8085 simulator, find the square of an 8 bit number.
15	Using an 8085 simulator, find out the factorial of a number.

Module/ Unit	Course Description	Hrs	CO No.	PSO No.	PO No.
1	Fundamentals of Computers	15h	1	1	1
2	Digital Circuits & CPU, Processor Organization	15h	2	1	1

Paper II Programming with Python I

Course Description	
Semester	I
Course Name	Programming with Python I
Course Code	UCS1IPP
Credit	3
Theory	2 hrs. per week
Practical	2 hrs. per week
Hours	30

Course Objectives

1. To master the fundamentals of python
2. To make the learners understand the concepts of basic and advanced python
3. To discover how to work with lists, tuples, dictionary

Course Outcomes

Learners will be able to:

1. Understand the basics of python
2. Apply functions, loops, conditional statements
3. Create the lists, tuples and dictionaries

Syllabus		Total 30 L
Unit I	**Overview of Python:** History, Features of Python, Python Interpreter,	15 L

	IDLE, Execution of Python Program, Python syntax, Comments. **Data Types, Variables and Operators: Data types**- Numeric, String, Boolean, Dictionary, Sets, Lists, Tuple, Data type conversion. **Variables**- Variable names, Creating and printing variables, local and global variables, Assigning values. **Operators**- Arithmetic operators, Assignment operators, Comparison operators, Logical operators, Identity operators, Membership operators, Bitwise operators. Operator precedence. **Input and output operations:** Input() function, Output statements, print() function, formatted output, range() function. **Control Statements: Conditional Statements**- if statement, if…elif Statement, if…elif … else Statement, And, Or and Not keyword, Loops-while loop, for loop, Infinite loop, Nested loop, break statement, continue statement, pass statement, assert statement, return statement	
Unit II	**Functions**: Function definition and call, Built-in and User defined Functions, Parameters and Arguments, Formal and Actual Arguments, keyword arguments, default arguments, List as an argument, Returning values, Pass statement, Recursion, Anonymous Function, Using Lambda with filter(), map() and reduce(). **Strings:** Creating Strings, looping through strings, Slicing, Modifying, and Concatenation, formatting. Escape characters, string methods. **List and Tuples: Lists:** List operations, list functions, List Slices, Nested Lists. **Tuples:** accessing, updating, Tuple operations, tuple functions in Tuple. **Dictionaries:** Creating a Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries, Passing Dictionaries to Functions.	15 L
Text Books	1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries, Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014 2. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016.	

Reference Books	1. Programming through Python, M. T Savaliya, R. K. Maurya & G M Magar, Sybgen Learning India, 2020Paul Gries , Jennifer Campbell, Jason Montojo, 2. Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014 3. James Payne , Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010

Sr. No.	**Practicals of programming with Python I**
1	Execution of basic operations in python.
2	Programs based on input () and formatted output.
3	Programs based on demonstration of variables, operators and data types.
4	Programs based on conditional statements.
5	Programs based on looping statements.
6	Programs based on break and continue statements.
7	Programs based on pass and return statements.
8	Programs based on user defined functions.
9	Programs based on anonymous functions.
10	Programs based on recursion.
11	Programs based on filter(), map() and reduce() functions.
12	Programs based on strings.
13	Programs based on lists.
14	Programs based on tuples.
15	Programs based on dictionaries.

Module/ Unit	Course Description	Hrs	CO No.	PSO No.	PO No.
1	Python Basics, Conditional and Looping Statements	15h	1,2	2	2
2	Functions, List, Dictionaries and Tuples	15h	3	2	2

VSC

Paper III Statistics for Data Science

Course Description	
Semester	I
Course Name	Statistics for Data Science
Course Code	UCS1SDS
Credit	2
Theory	1 hr. per week
Practical	2 hrs. per week
Hours	30

Course Objectives

1. To provide an understanding for the learners on statistical concepts to include measures of dispersion, probability distribution.

Course Outcomes

Learners will be able to:

1. Recall measures of central tendency.

2. Describe the measures of dispersion and Classify discrete, continuous probability distribution.

Syllabus		Total 30 L
Unit I	**Measures of central tendency and dispersion:** Averages, Arithmetic Mean, Median, Mode, Empirical Relation Between the Mean, Median, and Mode, Geometric Mean, Harmonic Mean, Relation Between the Arithmetic, Geometric, and Harmonic Means, Quartiles, Deciles and Percentiles. **Standard Deviation and Other Measures of Dispersion**: Dispersion, or Variation, Range, Mean Deviation, Semi-Interquartile Range, 10–90 Percentile Range, Standard Deviation, Variance. **Elementary Probability Theory:** Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Mathematical Expectation, Discrete and continuous Probability distribution.	15 L
Text Books	1. STATISTICS, Murray R Spiegel, Larry J. Stephens, mcgraw –HILL INTERNATIONAL, Fourth edition.	
Reference Books	1. A Practical Approach using R, R.B. Patil, H.J. Dand and R. Bhavsar, SPD publication, First edition. 2. Fundamental of mathematical statistics s.c. gupta and v.k. kapoor, sultan chand and sons, eleventh edition.	

Sr. No.	Practical's of Statistics for Data Science
1	Using R execute the basic commands, array, list and frames, sequences and repetition.
2	Create a Matrix using R and Perform the operations: addition, multiplication
3	Create a Matrix using R and Perform the operations inverse
4	Create a Matrix using R and Perform the operation: transpose
5	Using R Execute the statistical function: mean
6	Using R Execute the statistical function: median
7	Using R Execute the statistical function: mode
8	Using R Execute the statistical functions: quartiles
9	Using R Execute the statistical function: range
10	Using R Execute the statistical function: inter quartile range histogram.
11	Using R import the data from Excel / .CSV file and perform the above functions

12	Using R import the data from Excel / .CSV file and calculate the standard deviation
13	Using R import the data from Excel / .CSV file and calculate the variance, co-variance.
14	Using R compute the probability.
15	Using R compute the conditional probability.

Module/ Unit	**Course Description**	**Hrs**	**CO No.**	**PSO No.**	**PO No.**
1	Measures of central tendency and dispersion, Elementary Probability Theory	15h	1,2	2	4

SEC Web Designing

Course Description	
Semester	I
Course Name	Web Designing
Course Code	UCS1WEB
Credit	2
Theory	1 hrs per week
Practical	1 hrs per week
Hours	30

Course Objectives

1. To understand Internet and WWW

2. To learn the language of the web: HTML5 and CSS

Course Outcomes

Learners will be able to:

1. Design Table, Form using HTML5.
2. Understand the concept of CSS.

Syllabus		**30 L**
Unit I	**Internet and the World Wide Web:** Introduction to the internet and its applications, E-mail, Telnet, FTP, E-Commerce, Video Conferencing, e-business. Internet service providers, domain name server, internet address **World Wide Web (WWW):** World Wide Web and its evolution, uniform resource locator (URL), browsers – internet explorer, Netscape navigator, opera, Firefox, chrome, Mozilla. search engine, web saver – apache, IIS, proxy server, HTTP protocol **HTML5:** Introduction, Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets.	15 L
Text Books	1.Web Design The Complete Reference, Thomas Powell ,Tata McGraw Hill 2. HTML5 Step by Step ,Faithe Wempen, Microsoft Press 2011	
Reference Books	1. CSS - HTML5 - JavaScript: HTML , Jennifer Kyrnin, Third Edition 2. Learning Web Designing, Jennifer Niederest Robbins, Fourth Edition 3. https://html.com/	

Sr. No.	**Practical of Web Designing**
1.	Create an html page with 7 separate lines in different colors. State color of each line in its text

2.	Write html code to generate the following output. • Coffee • Tea o Black Tea o Green Tea • Milk
3.	Write a program using CSS.
4.	Create a user form using HTML5.
5.	Write a program for creating hyperlinks and anchors.
6.	Implement a program using navigational aids.
7.	Write a program to create division based layout.
8.	Write a program to create a timetable.
9.	Create a web page using <a> </a>.
10.	Write a code using HTML media.

Module/ Unit	**Course Description**	**Hrs**	**CO No.**	**PSO No.**	**PO No.**
1	Internet and the World Wide Web, HTML5, CSS	15h	1,2	1	2

SEC: Programming with C

Course Description	
Semester	I
Course Name	Programming with C
Course Code	UCS1PWC
Credit	2
Theory	1 hrs. per week
Practical	1 hrs. per week
Hours	15

Course Objectives

1. To provide basic concepts of C programming language.
2. To describe implementation of operators, data types and loops.

Course Outcomes

Learners will be able to:

1. Implement basic concepts of C.
2. Develop code based on decision making.

Syllabus		15 L
Unit I	**Programming Paradigms:** Use of Algorithms/Flow Charts for problem solving **Structure of C program**: Header and body, Use of comments. Interpreters' vs compilers, Python vs C. Compilation of a program. **Formatted I/O:** print(), scan (). **Data**: Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers. Compare with data types in Python. Compare static typing in C vs dynamic typing in Python **Variables**: Declaring variables, scope of the variables according to block, hierarchy of data types. Compare explicit declarations in C with implicit declarations in Python. **Types of operators**: Arithmetic, relational, logical, compound assignment, increment and decrement, conditional or ternary, bitwise and comma operators. Precedence and order of evaluation, statements and Expressions. Automatic and explicit type conversion. **Iterations**: Control statements for decision making: (i) Branching: if statement, else. if statement, (does the writer mean if-else or nested ifs) switch statement	15 L

Text Books	Programming in ANSI C (Third Edition): E Balagurusamy, TMHrosoft Press 2011
Reference Books	1. Pradip Dey, Manas Ghosh, “Programming in C”, second edition, Oxford University Press

Sr. No.	**Practical’s of Programming with C**
1.	Write a program to understand the basic data types and I/O.
2.	Write a program on Operators.
3.	Write a program on Expressions.
4.	Write a program on decision statements
5.	Write a program on for loop.
6.	Write a program on while loop.
7.	Write a program on 1-D arrays.
8.	Write a program on multidimensional array.
9.	Write a program on functions.
10.	Write a program on structures.
11.	Write a program on unions.
12.	Write a program on Recursion.
13.	Write a program on Data Input and Output Functions.
14.	Write a program on nested for loop.
15.	Write a program on Jump statements.

Module/ Unit	**Course Description**	**Hrs**	**CO No.**	**PSO No.**	**PO No.**
1	Introduction to basic concepts of C	15h	1,2	1	1

Indian Knowledge System: History and Evolution of Computing

Course Description	
Semester	I
Course Name	History and Evolution of Computing
Course Code	UCS1IKS
Credit	2
Theory	1 hr. per week
Practical	1 hr. per week
Hours	30

Course Objectives

1. To understand the Indian Knowledge System in terms of computing.

2. To interpret the difference between traditional and modern knowledge system.

Course Outcomes:

Learners will be able to

1. Understand the Indian Knowledge System in terms of computing.

2. Apply concepts of modern computing.

Syllabus		**30 L**
Unit I	Introduction to Computers: What is Computer? Basic Computer Terminologies, Analog, Digital and Hybrid Computers, Characteristics of Computer, History and Evolution of Computer, Early Computing Devices , Generation of Computer, Applications Computer Architecture: Computer System Hardware, Computer Memory, Processor, Secondary Storage Devices, Input Output Devices Types of Translator Software, Application Software, Software Acquisition, Generation of Programming Languages: Machine Language, Assembly Language and High Level Language, Computing with Green Energy and E-Waste	15 L
Unit II	Introduction to Operating System, Objectives, Types of OS, Functions of Operating System, Process Management: Process States, CPU Scheduling, Process Synchronization, Deadlock Data Communication and Computer Networks, The Internet, Introduction to Computer Programming Fundamentals: Program Development Life Cycle, Algorithm, Control Structures, Flow Chart , Pseudo Code, Programming Paradigms: Structured Programming, Object Oriented Programming	15 L
Text Books	1. Computer Fundamentals, Goel Anita ,Pearson	
Reference Books	1. Operating System Concepts by Silberschatz, Galvin , Gagne, John Wiley and Sons 2. Data Communication by Behrouz A. Forouzan, 5th Edition, Mcgraw Hills 3. https://www.javatpoint.com/ 4. https://www.geeksforgeeks.org/	

Module/ Unit	Course Description	Hrs	CO No.	PSO No.	PO No.
1	Introduction to Computer	15h	1	2	1
2	Introduction to Operating System	15h	2	2	2

SEMESTER II

Paper I Relational Database Management System

Course Description	
Semester	II
Course Name	Relational Database System
Course Code	UCS2RDS
Credit	3
Theory	2 hrs. per week
Practical	2 hrs. per week
Hours	30

Course Objectives

1. To understand the basic concepts and the applications of database systems.

2. To Master the basics of SQL and construct queries using SQL.

Course Outcomes

Learners will be able to:

1. Illustrate the basic elements of a relational database management system.

2. Ability to identify the data models for relevant problems.

Syllabus		Total 30 L
Unit I	**Introduction:** Basic concepts, Advantages of DBMS over file processing systems, Data abstraction, Database languages, Data models, Data independence, Components of a DBMS, Multi-user DBMS architecture, Data Modeling: Basic concepts, Entity, attributes, relationships, Relational Constraints and keys. **ER diagrams:** Components of ER model, Conventions, Converting ER diagrams into tables Relational Model, Codd's rules **Relational Algebra**: Selection and projection set operations, renaming, Joins, Division **Query Processing:** Overview, Measures of Query cost, Selection and Join operations, Evaluation of Expressions, Query optimization	15 L
Unit II	**Introduction to SQL:** SQL Data Types, Literals, DDL, DML, DCL SQL Operators, Tables, Views, Backup and Recovery, Indexing **SQL DML Queries**: SELECT query and clauses, Set operations, Tuple, Variables, Set comparison, Ordering of Tuples , Aggregate Functions, Date Functions, String Functions, Math functions, Nested Queries **Programmatic SQL** : Embedded SQL, Dynamic SQL, ODBC **Relational Databases Design:** Normalization, Data Redundancy and Update Anomalies, Functional Dependencies.	15 L
Text Books	1. Silberschatz A., Korth H., Sudarshan S. "Database System Concepts", 6th edition, Tata McGraw Hill Publishers 2. G. K. Gupta "Database Management Systems" , Tata McGraw Hill	

Referenc e Books	1. Rab P., Coronel C. “Database Systems Design, Implementation and Management”, 5th edition, Thomson Course Technology, 2002 2. Elmasri R., Navathe S. “ Fundamentals of Database Systems”, 4th edition, Pearson Education, 2003 3. Date C. “ An Introduction to Database Systems”, 7th edition, Pearson Education, 2002 4. Ramkrishna R., Gehrke J. “ Database Management Systems”, 3rd edition, McGraw Hill	

Sr. No.	Practical’s of Database Management System
1	For given scenario Draw an E-R diagram and convert entities and relationships to tables.
2	Write relational algebra queries on any five tables
3	Manipulating Data • Using INSERT statement • Using DELETE statement • Using UPDATE statement
4	Perform the following: • Viewing all databases • Creating a Database • Viewing all Tables in a Database • Creating Tables
5	Create table by using different constraints in SQL
6	Perform the following: • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database
7	Perform the following: • Granting and revoking permissions • Saving (Commit) and Undoing (rollback)
8	Queries involving • Date Functions • String Functions • Math Functions

9	• Simple Queries with Aggregate functions • Queries with Aggregate functions (group by and having clause)
10	Subqueries • With IN clause • With EXISTS clause
11	Join Queries • Inner Join • Outer Join
12	Views • Creating Views (with and without check option) • Dropping views • Selecting from a view
13	Create, update, alter, and drop sequence.
14	Create Stored Procedures
15	Perform the normalization by removing anomalies from tables

Module/ Unit	**Course Description**	**Hrs**	**CO No.**	**PSO No.**	**PO No.**
1	Basic Concepts of DBMS	15h	1	2	1
2	Introduction to SQL, Programmatic SQL	15h	2	3	2

Paper II Python Programming II

Course Description	
Semester	II
Course Name	Python Programming II
Course Code	UCS2APP
Credit	3

Theory	2 hrs. per week
Practical	2 hrs. per week
Hours	30

Course Objectives

1. To understand the concepts of file handling.
2. To understand the concepts of exception handling.
3. To design the graphical user interfaces using tkinter.

Course Outcomes

Learners will be able to:

1. Implement the concepts of file handling.
2. Implement the concepts of exception handling.
3. Develop the Graphical user Interfaces using tkinter.

Syllabus		**Total** **30 L**
Unit I	**Python File Input-Output:** Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. **Iterables:** iterators and their problem-solving applications. **Exception handling:** What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise. **Regular Expressions:** Concept of regular expression, various types of regular expressions, using match function. **GUI Programming in Python** (using Tkinter/wxPython/Qt) What is GUI, Advantages of GUI, Introduction to GUI library. Layout management, events and bindings, fonts, colors, drawing on canvas (line, oval, rectangle, etc.) Widgets such as : frame, label, button, check button, entry, list box, message, radio button, text, spin box etc.	15 L

Unit II	**Database connectivity in Python**: Installing MySQL connector, accessing connector module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity. **Game Design**: Introduction to google collab notebook, Introduction to Jupyter notebook, Pygame: how to create the game window, Creating Basic Movements and key Press, changing title and background color, adding images, Adding Sounds Adding Effects etc	15 L
Text Books	1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014	
Reference Books	1. James Payne , Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010 2. A. Lukaszewski, MySQL for Python: Database Access Made Easy, Pact Publisher, 2010 3. Making Games with Python & Pygame	

Sr. No.	**Practical's of Python Programming II**
1	Write Python Programs to read and write files.
2	Write Python Programs with iterables and iterators.
3	Write a Python Program to demonstrate exception handling using try-catch keywords.
4	Write a Python Program to demonstrate exception handling using finally and raise keywords.
5	Write a Python Program to demonstrate the use of regular expressions using meta characters.
6	Write a Python Program to demonstrate the use of regular expressions using functions.

7	Write a Python Program to draw scenery using Tkinter
8	Program to draw shapes & GUI controls: a. Advance Calculator b. Simple Interest Form
9	Program to draw shapes & GUI controls: a. Pizza Ordering GUI Form b. BMI Calculator GUI Form
10	Write a Python Program on database connectivity to create a table and insert values in it.
11	Write a Python Program on database connectivity to update and delete a table.
12	Create a Python Game in Google-Collaboratory notebook.
13	Create a Python Game in Jupyter Notebook
14	Write a python program to create a game window using pygame
15	Write a python program to demonstrate basic movements and key press using pygame

Module/ Unit	Course Description	Hrs	CO No.	PSO No.	PO No.
1	Python File Input-Output, GUI Programming in Python	15h	1	2	1
2	Database Connectivity in Python	15h	2	2	3

Minor: Artificial Intelligence

Course Description	
Semester	II
Course Name	Artificial Intelligence
Course Code	UCS2ARI
Credit	2
Theory	1 hrs per week

Practical	2 hrs per week
Hours	30

Course Objectives

Learners will be able to:

1. To understand the basic concepts of AI
2. To apply different AI strategies for problem solving

Course Outcomes

Learners will be able to:

1. Explain the concepts, techniques and building blocks of AI
2. Apply different AI strategies for problem-solving, inference, vision, knowledge representation and learning

Syllabus		**Total** **30 L**
Unit I	**What Is AI**: Foundations, History and State of the Art of AI. **Intelligent Agents**: Agents and Environments, Nature of Environments, Structure of Agents. **Problem Solving by searching:** Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.	15 L
Unit II	**Learning from Examples**: Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, Theory of Learning, Regression and Classification with Linear Models, Artificial Neural Networks, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning	15 L
Text Book s	1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig,3rd Edition, Pearson, 2010.	

Reference Books	1. Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017. 2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition,	
	2017 3. The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013	

Sr. No	**Practicals of Artificial Intelligence**
1	Case study on problem formulation for vacuum cleaner.
2	Case study on problem formulation for tic-tac-toe.
3	Case study on problem formulation for eight queen problem.
4	Case study on problem formulation for chess.
5	Case study on problem formulation for missionary and cannibals problem.
6	Case study on problem formulation for 8 puzzle problem.
7	Case study on problem formulation for robot navigation.
8	Implement Breadth first search algorithm for Romanian map problem.
9	Implement Iterative deep depth first search for Romanian map problem.
10	Implement A* search algorithm for Romanian map problem.
11	Implement recursive best-first search algorithm for Romanian map problem.
12	Implement decision tree learning algorithm for the restaurant waiting problem.
13	Implement feed forward back propagation neural network learning algorithm for the restaurant waiting problem.
14	Implement Adaboost ensemble learning algorithm for the restaurant waiting problem.
15	Implement Naive Bayes' learning algorithm for the restaurant waiting problem.

Module/ Unit	Course Description	Hrs	CO No.	PSO No.	PO No.
1	Fundamentals of AI	15h	1,2	1	1
2	Learning from Examples	15h	3	2	3

VSC: Linear Algebra: Applications in Computer Science

Course Description	
Semester	II
Course Name	Linear Algebra: Applications in Computer Science
Course Code	UCS2LAR
Credit	2
Theory	1 hr. per week
Practical	2 hrs. per week
Hours	15

Course Objectives:

1. To introduce the concept of the vectors, matrices, linear transformations

Course Outcomes

Learners will be able to:

1. Define vector spaces and subspaces.
2. Relate matrices and linear transformation.

Syllabus		**Total** **15 L**
Unit I	**Vector Spaces**: Vector Spaces: Definition and examples, Subspaces: Definition and examples. **Matrices and Linear transformations**: Matrices, properties of matrices, Transpose of a matrix and its properties, Type of matrices, determinant, inverse of a matrix, echelon form, rank of a matrix, Linear transformation: definition and examples, kernel and image of a linear transformation: definition and examples, Rank-Nullity Theorem (only statement) and examples, Matrix representation of a linear Transformation	15 L
Text Book s	1. K. Hoffman and R. Kunze: Linear Algebra, Tata McGraw-Hill, New Delhi, 1971. 2. Vikas Bisht and Vivek Sahai, Linear Algebra, Alpha Science International Limited, 2002.	
Referenc e Books	1. Serge Lang, Introduction to Linear Algebra, Second Edition, Springer. 2. S. Kumaresan, Linear Algebra, A Geometric Approach, Prentice Hall of India, Pvt. Ltd, 2000.	

Sr. No	**Practical's of Linear Algebra:** **Applications in Computer Science**
1	Write a program to do the following: ● Enter an r by c matrix M (r and c being positive integers) ● Display M in matrix format
2	Write a program to do the following: ● Display the rows and columns of the matrix M
3	Write a program to do the following: ● Find the scalar multiplication of M for a given scalar.

4	Write a program to do the following: ● Find the transpose of the matrix M.
5	Write a program to do the following: ● Find the vector –matrix multiplication of a r by c matrix M with an c-vector u.
6	Write a program to do the following: ● Find the matrix-matrix product of M with a c by p matrix N.
7	Write a program to do the following: ● Enter a vector u as a n-list ● Enter another vector v as a n-list
8	Enter two distinct functions as vectors u and v.
9	Write a program to do the following: ● Find the vector au+bv for different values of a and b
10	Write a program to do the following: ● Find the dot product of u and v
11	Write a program to do the following: ● find determinant of a matrix
12	Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse.
13	Write a program to convert a matrix into its row echelon form.
14	Write a program to determine linearity.
15	Write a program to find matrix representation of a linear transformation.

Module/	Course Description	Hrs	CO	PSO	PO
Unit			No.	No.	No.
1	Vector Spaces, Matrices and Linear transformations	15h	1,2	2	1

SEC: PHP Programming

Course Description	
Semester	II
Course Name	PHP Programming
Course Code	UCS2PHP
Credit	2
Theory	1 hr. per week
Practical	2 hrs. per week
Hours	15

Course Objectives

1. To understand how server-side programming works on the web

Course Outcomes

Learners will be able to:

1. Understand the basic concepts of PHP and its applications.

<table>
<tr><th colspan="2">Syllabus</th><th>Total 15 L</th></tr>
<tr><td><b>Unit I</b></td><td><b>PHP Fundamentals:</b> Introduction to PHP, installation, and configuration, Embedding PHP with HTML, Variable, Data types, Operators, Printing Statements, Comments, Conditions, loops, Array, Multidimensional Array, Associative array, Functions, String functions, Numeric functions, Classes, Regular Expression, Working with Date time, code re-use, require (), include (), and the include- path; file system functions, and file input and output; file uploads; error handling and logging; sending mail, Handling Sessions and Cookies in PHP.</td><td>15 L</td></tr>
<tr><td><b>Text Books</b></td><td>1. Beginning PHP, Apache, MySQL Web Development. Michael K. Glass, Yann Le Scouarnec, Elizabeth Naramore, Gary Mailer, Jeremy Stolz, Jason Gerner
2. Murach's PHP and MySQL, Joel Murach Edition 2, Mike Murach & Associates, Incorporated, 2014</td><td></td></tr>
<tr><td><b>Reference Books</b></td><td>1. Programming PHP, Rasmus Lerdorf, Kevin Tatroe, Bob Kaehms, Ric McGredy
2. PHP & MySQL: Server-side Web Development, Jon Duckett</td><td></td></tr>
</table>

Sr. No.	**Practical’s of PHP Programming**
1	Write a program to give demo of ECHO and PRINT command.
2	Programs on condition Checking.
3	Programs on Looping constructs.
4	Create a database in MySql and connect that database from PHP.
5	Write a program to insert the values of table database MySql.
6	Write a program to Update the values of table database MySql.

7	Write a program to delete the values of table database MySql.
8	Program on File handling operations.
9	Programs on Session handling.
10	Program on the Cookies handling.
11	Program on file uploading and Program on image storage in the database.
12	Program to demonstrate Joomla framework.
13	Program to demonstrate Zend framework.
14	Program to demonstrate Laravel framework.
15	Program for email sending.

Module/Unit	**Course Description**	**Hrs.**	**CO No.**	**PO No.**	**PSO No.**
Unit I	PHP Fundamentals	15L	1	3	2

SEC: C++ Programming

Course Description	
Semester	II
Course Name	C++ Programming
Course Code	UCS2CPP
Credit	2
Theory	1 hr. per week
Practical	2 hrs. per week
Hours	15

Course Objectives

1. To understand object oriented programming and advanced C++ concepts.

Course Outcomes

Learners will be able to:

1. Implement features of object-oriented programming to solve real world problems.

Syllabus		**Total 15 L**
Unit I	**Principles of Object Oriented Programming (OOP):** Evolution of C++ - Programming Paradigms - Key Concepts of OOP - Advantages of OOP -Usage of OOP and C++ .Input and Output in C++- Streams-Stream classes Unformatted console I/O operations- Member functions of istream class-manipulators-manipulators with parameters **Introduction to C++:** Tokens, Keywords, Identifiers, Variables, Operators, Expressions and Control Structures: If, If. Else, Switch – Repetitive Statements- for, while, do…while – User defined data types , System defined data types, Function, Class, Object.	15 L
Text Books	1. “Object Oriented Programming with C++”, E. Balagurusami, Fourth Edition, Tata Mc-Graw Hill 2. “Object Oriented Programming in Turbo C++”, Robert Lafore, Fourth Edition Galgotia Publications.	
Reference Books	1. “The C++ Programming Language”, Bjarne Stroustrup, Third Edition, Addison-Wesley Publishing Company. 2. “Object Oriented Programming Using C++”, Salaria, R. S, Fourth Edition, Khanna Book Publishing 3. “Object Oriented Programming with ANSI & Turbo C++”, Ashok N. Kamthane, Pearson Education, 2006	

Sr. No.	**Practical of C++ Programming**
1.	Write a Program to read a set of numbers in an array and to find the largest of them.
2.	Write a Program to exchange the contents of two variables using call by value and call by reference.
3.	Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used respectively. Where getInfo() will be the private method.
4.	Design the class student, containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively. Where getData() will be private method.
5.	Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check if the given number is palindrome, isArmstrong() which will calculate the given number is armstrong or not.
6.	Write a program to demonstrate function definition outside class and accessing class members in function definition.
7.	Write a friend function for adding the two complex numbers, using a single class.
8.	Write a friend function for adding the two different distances and display its sum, using two classes.
9.	Design a class Geometry containing the methods area() and volume() and also overload the area() function .
10	Design a class Complex for adding the two complex numbers and also show the use of constructor.
11.	Programs on Operator Overloading:
	a. Overload the operator unary(-) for demonstrating operator overloading.
	b. Overload the + for concatenating the two strings. For e.g., “Py” + “thon” = Python.
12.	Programs on Inheritance:

	a. Design a class for single level inheritance.
	b. Design a class for multiple inheritance.
	c. Implement hierarchical inheritance.
13.	Programs on Virtual functions and abstract classes:
	a. Implement the concept of method overriding.
	b. Show the use of virtual functions.
	c. Show the implementation of abstract class.
14.	Programs on Exception handling:
	a. Show the implementation of exception handling.
	b. Show the implementation for exception handling for strings.
15	Program to demonstrate multithreading concept.

Module/Unit	**Course Description**	**Hrs.**	**CO No.**	**PSO No.**	**PO No.**
Unit I	Introduction to C++	15L	1, 2	2	2