



ESTD : 1946

**THE NATIONAL INSTITUTE OF ENGINEERING**

**MYSORE – 8**

**(Autonomous Institution under VTU)**

**B.E - CSE**

**Scheme of III – IV Semester**

**Department of Computer Science and Engineering**

| The National Institute of Engineering, Mysuru |                |             |  |                          |                                    |                               |   |   |   |             |            |            |             |           |
|---|----------------|-------------|--|--------------------------|------------------------------------|-------------------------------|---|---|---|-------------|------------|------------|-------------|-----------|
| Scheme of Teaching and Examination            |                |             |  |                          |                                    |                               |   |   |   |             |            |            |             |           |
| Department: Computer Science and Engineering  |                |             |  |                          |                                    |                               |   |   |   |             |            |            |             |           |
| B.E. in CSE - 2023 Admitted Batch             |                |             |  |                          |                                    |                               |   |   |   |             |            |            |             |           |
| Semester : III                                |                |             |  |                          |                                    |                               |   |   |   |             |            |            |             |           |
| Sl. No.                                       | Type of Course | Course Code | Course Title                                 | Teaching Department (TD) | Question Paper setting Board (PSB) | Teaching Hrs/Week             |   |   |   | Examination |            |            |             | Credits   |
|   |                |             |  |                          |                                    | L                             | T | P | S | Duration in | CIE Marks  | SEE Marks  | Total Marks |           |
| 1   | PCC/BSC        | BCS301      | Mathematics for Computer Science             | TD: Maths                | PSB: Maths                         | 3                             | 2 | 0 | - | 3           | 50         | 50         | 100         | 4         |
| 2   | IPCC           | BCS302      | Digital Design & Computer Organization       | TD: CS                   | PSB: CS                            | 3                             | 0 | 2 | - | 3           | 50         | 50         | 100         | 4         |
| 3   | IPCC           | BCS303      | Operating Systems                            | TD: CS                   | PSB: CS                            | 3                             | 0 | 2 | - | 3           | 50         | 50         | 100         | 4         |
| 4   | PCC            | BCS304      | Data Structures and Applications             | TD: CS                   | PSB: CS                            | 3                             | 0 | 0 | - | 3           | 50         | 50         | 100         | 3         |
| 5   | PCCL           | BCSL305     | Data Structures Laboratory                   | TD: CS                   | PSB: CS                            | 0                             | 0 | 2 | - | 3           | 50         | 50         | 100         | 1         |
| 6   | ESC            | BXX306X     | ESC/ETC/PLC                                  | TD: CS                   | PSB: CS                            | 3                             | 0 | 0 | - | 3           | 50         | 50         | 100         | 3         |
|   |                |             |  |                          |                                    | OR                            |   |   |   |             |            |            |             |           |
|   |                |             |  |                          |                                    | 2                             | 0 | 2 | - |             |            |            |             |           |
| 7   | UHV            | BCK307      | Social Connect and Responsibility            | TD: CS                   | PSB: CS                            | 0                             | 0 | 2 | - | 1           | 100        | -          | 100         | 1         |
| 8   | AEC/SEC        | BXX358X     | Ability Enhancement Course (AEC)             | TD: CS                   | PSB: CS                            | If the course is a Theory     |   |   |   |             | 50         | -          | 50          | 1         |
|   |                |             |  |                          |                                    | 1                             | 0 | 0 | - | 1           |            |            |             |           |
|   |                |             |  |                          |                                    | If the course is a Laboratory |   |   |   |             |            |            |             |           |
|   |                |             |  |                          |                                    | 0                             | 0 | 2 | - | 2           |            |            |             |           |
| 9   | MC             | BNSK359     | National Service Scheme (NSS)                | NSS Coordinator          |                                    | 0                             | 0 | 2 | - | -           | 100        | -          | 100         | 0         |
|   |                | BPEK359     | Physical Education (PE) (Sports & Athletics) | PED                      |                                    |                               |   |   |   |             |            |            |             |           |
|   |                | BYOK359     | Yoga   | Yoga Teacher             |                                    |                               |   |   |   |             |            |            |             |           |
| <b>Total</b>                                  |                |             |  |                          |                                    |                               |   |   |   |             | <b>550</b> | <b>350</b> | <b>850</b>  | <b>21</b> |

PCC: Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, MC: Mandatory Course (Non-credit), AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, L: Lecture, T: Tutorial, P: Practical: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation. K :This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

| <b>Engineering Science Course (ESC/ETC/PLC)</b> |                                       |         |                                |
|---|---------------------------------------|---------|--------------------------------|
| BCS306A   | Object Oriented Programming with Java | BCS306D | JavaScript                     |
| BCS306B   | Object Oriented Programming with C++  |         |                                |
| <b>Ability Enhancement Course -IV</b>           |                                       |         |                                |
| BCS358A   | Data Analytics with Excel             | BCS358D | Data Visualization with Python |
| BCS358B   | Data Analytics with R                 |         |                                |
| BCS358C   | Project Management with Git           |         |                                |

**Code: BCS302****Course: Digital Design & Computer Organization****Credits: 4****CIE: 50 Marks****L:T:P - 3:0:2****SEE: 50 Marks****SEE Hours: 3****Total Marks:100**

| Prerequisites if any       | Fundamentals of Logic   |
|----------------------------|---|
| <b>Learning objectives</b> | 1. To provide the knowledge to explain the fundamentals of Logic circuits and combinational circuits.<br>2. To introduce the design of sequential circuit systems, Registers and Counters<br>3. The basics involved in number representation and arithmetic operations in the computer system.<br>4. Basic processor concept, instruction execution and Bus architecture, Memory architecture and mapping techniques. |

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

| COs | Course Outcomes   | Bloom's level |
|-----|---|---------------|
| CO1 | Use logic minimization techniques and design combinational circuits using logic gates.                      | Apply         |
| CO2 | Design and analyze sequential circuits systems, Registers and Counters.                                     | Apply         |
| CO3 | Use algorithms to perform fast multiplication, division and to represent floating point numbers in binary.  | Apply         |
| CO4 | Explain the basic processing unit and design of control system, memory architecture and mapping techniques. | Understanding |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | 2   | -    | -    | 2    | 2    | -    |
| CO2 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | 2   | -    | -    | 2    | 2    | -    |
| CO3 | 3   | 2   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 2    | -    |
| CO4 | 3   | 2   | -   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 2    | -    |

**Mapping Strength:**      **Strong – 3**      **Medium – 2**      **Low – 1**

**Course Structure**

| Module 1-Combinational Circuits |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|---------------------------------|---|----------------------|-----------------------|------------------------|
| 1.1                             | <b>Simplification of Boolean Functions:</b> The Map Method, Two-and Three Variable Maps, Four- Variable Maps, Don't-Care Conditions | 2                    | -                     | -                      |
| 1.2                             | <b>Combinational Logic:</b> Adders, Subtractors   | 2                    | -                     | 2                      |
| 1.3                             | Binary Parallel Adder, Decimal Adder  | 3                    | -                     | 1                      |
| 1.4                             | Decoders, Multiplexer   | 3                    | -                     | 2                      |
| Module 2-Sequential Circuits    |   |                      |                       |                        |
| 2.1                             | <b>Sequential Logic:</b> Introduction, Flip-Flops, Triggering of Flip Flops   | 2                    | -                     | 2                      |
| 2.2                             | Analysis of Clocked Sequential Circuits   | 2                    | -                     | -                      |
| 2.3                             | State Reduction and Assignment  | 1                    | -                     | -                      |
| 2.4                             | Design Procedure  | 1                    | -                     | -                      |
| 2.5                             | <b>Registers, Counters:</b> Introduction, Registers   | 1                    | -                     | -                      |
| 2.6                             | Shift Registers, Ripple-Counters  | 2                    | -                     | 1                      |
| 2.7                             | Synchronous & Asynchronous Counters.  | 1                    | -                     | 2                      |
| Module 3-Arithmetic Unit        |   |                      |                       |                        |

|                                     |  |           |           |           |
|-------------------------------------|--|-----------|-----------|-----------|
| 3.1                                 | <b>Arithmetic unit:</b> Multiplication of Positive numbers A signed operand multiplication             | 2         | -         | -         |
| 3.2                                 | Bit pair recoding of multipliers, carry save addition of summands                                      | 3         | -         | -         |
| 3.3                                 | Integer division, IEEE standard for floating point numbers.  | 2         | -         | -         |
| <b>Module 4-The Processor</b>       |  |           |           |           |
| 4.1                                 | <b>The Processor:</b> Fundamental concepts   | 1         | -         | -         |
| 4.2                                 | Execution of complete instruction, Multiple-Bus organization   | 2         | -         | -         |
| 4.3                                 | Hardwired control unit and Micro programmed control unit   | 4         | -         | -         |
| <b>Module 5-Memory Unit</b>         |  |           |           |           |
| 5.1                                 | <b>Memory Unit:</b> Basic concepts, Internal organization of memory chips, Structure of larger memory. | 3         | -         | -         |
| 5.2                                 | Cache memories, Mapping functions  | 3         | -         | -         |
| <i>Total No. of Lecture Hours</i>   |  | <b>40</b> | -         | -         |
| <i>Total No. of Tutorial Hours</i>  |  |           | <b>00</b> | -         |
| <i>Total No. of Practical Hours</i> |  |           |           | <b>10</b> |

| <b>Integrated Lab Component: Digital Design &amp; Computer Organization</b> |     |  |
|---|-----|--|
| Sl. No  | COs | <b>PART-A HARDWARE</b>   |
| 1.  | CO1 | a) Implement Half adder and Full adder using logic gates.<br>b) Implement Half subtractor, Full subtractor using logic gates |
| 2.  | CO1 | Implement BCD to Excess-3 using basic gates.   |
| 3.  | CO1 | Realize a full adder or any Boolean function using 3:8 decoder IC.   |
| 4.  | CO1 | Given any four variable logic expression, realize the simplified logic expression using 8:1 multiplexer                      |
| 5.  | CO2 | Realize a JK master Slave Flip- Flop and verify its truth table.   |
| 6.  | CO2 | Realize mod-4 and mod-6 counter using Synchronous counter design   |
| <b>PART-B SIMULATION USING XILINX</b>                                       |     |  |
| 1.  | CO1 | Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioural model.                           |
| 2.  | CO1 | Write a VHDL code for a Half –Adder , Simulate and verify its working.   |
| 3.  | CO1 | Write a VHDL code for a Full –Adder , Simulate and verify its working.   |
| 4.  | CO1 | Write a VHDL code for a Full –Subtractor, Simulate and verify its working.   |
| 5.  | CO1 | Write a VHDL code for 2:1, 4:1 and 8:1 multiplexer , Simulate and verify its working.  |
| 6.  | CO2 | Write a VHDL code for a SR, D and JK flip flop , Simulate and verify its working.  |

**Text Books:**

1. Digital Design: M. Morris Mano, Pearson (2013), new print 5<sup>th</sup> edition
2. Computer Organization: C Hamacher, Z Vranesic, S Zaky:, Tata McGraw Hill, 5<sup>th</sup> Edition, 2011.

**Reference Books:**

1. Computer Architecture A Quantitative Approach: John L Hennessy, David A Patterson, Elsevier, 5th Edition 2012.
2. The Elements of Computing System – Building the Modern Computer from First Principles: Noam Nisan, Shimon Schocken, The MIT Press (2005).
3. Digital Principles and Applications: Donald P Leach, Albert Paul Malvino & Goutham Saha, TMH, 6th Edition, 2006.

**Online Resources:**

1. NPTEL: Switching Circuits and logic design  
[https://onlinecourses.nptel.ac.in/noc21\\_cs64](https://onlinecourses.nptel.ac.in/noc21_cs64)
2. NPTEL: Computer Organization and Architecture A Pedagogical Aspect  
[https://onlinecourses.nptel.ac.in/noc19\\_cs04/preview](https://onlinecourses.nptel.ac.in/noc19_cs04/preview)
3. Edx: Computation Structures 3: Computer Organization  
<https://www.edx.org/course/computation-structures-3-computer-mitx-6-004-3x0>
4. Coursera: Digital Systems - <https://www.coursera.org/learn/digital-systems>

**Code: BCS303**  
**Credits: 4**  
**L:T:P - 3:0:2**  
**SEE Hours: 3**

**Course: Operating Systems**  
**CIE: 50 Marks**  
**SEE: 50 Marks**  
**Total Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | None   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To understand the roles and responsibilities of Operating Systems.</li> <li>Familiarize about Process, Scheduling, Memory management and Deadlocks concepts.</li> </ol> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Understand the process management policies and process scheduling. Illustrate the concepts of Process synchronization. | Understanding |
| CO2 | Illustrate various strategies of Memory Management especially Paging and Segmentation.                                 | Apply         |
| CO3 | Apply various Page Replacement Techniques and identify the design issues in paging.                                    | Apply         |
| CO4 | Illustrate the various mechanisms to handle deadlock.  | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO2 | 2   | 2   | 2   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO3 | 3   | 2   | 2   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO4 | 2   | 3   | 2   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

**Course Structure**

|   |  | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|---|--|----------------------|-----------------------|------------------------|
| <b>Module – 1: Processes , Threads and Scheduling</b> |  |                      |                       |                        |
| 1.1   | <b>Introduction:</b> what is an operating system?  | 1                    | -                     | -                      |
| 1.2   | The Operating system as an extended machine, The Operating system as a resource manager.                               | 1                    | -                     | -                      |
| 1.3   | <b>Processes:</b> The process model, Process creation, Process termination   | 1                    | -                     | -                      |
| 1.4   | Process hierarchies, Process states.   | 1                    | -                     | -                      |
| 1.5   | <b>Threads:</b> The Thread model, Thread usage, implementing threads in user space, implementing threads in the kernel | 1                    | -                     | -                      |

|  |   |   |   |   |
|--|---|---|---|---|
| 1.6  | Hybrid implementations, Scheduler activations   | 1 | - | - |
| 1.7  | <b>Scheduling:</b> Introduction to scheduling, Scheduling in batch Systems                                | 1 | - | 2 |
| 1.8  | Scheduling in interactive systems.: Priority based Scheduling   | 1 | - | 1 |
| 1.9  | Round robin Scheduling  | 1 | - | 1 |
| <b>Module – 2: Interprocess Communication</b>                    |   |   |   |   |
| 2.1  | <b>Interprocess communication:</b> Race conditions, Critical regions                                      | 1 | - | - |
| 2.2  | Mutual exclusion with busy waiting  | 1 | - | - |
| 2.3  | Sleep and wakeup  | 1 | - | - |
| 2.4  | Semaphores  | 1 | - | 1 |
| 2.5  | Message Passing   | 1 | - | - |
| 2.6  | Barriers  | 1 | - | - |
| 2.7  | <b>Classical IPC problems:</b> The dining philosophers' problem   | 1 | - | - |
| 2.8  | The Readers and Writers problem.  | 1 | - | - |
| <b>Module – 3: Memory Management</b>                             |   |   |   |   |
| 3.1  | <b>Memory Management:</b> Background  | 1 | - | - |
| 3.2  | Swapping  | 1 | - | - |
| 3.3  | Contiguous Memory Allocation  | 1 | - | - |
| 3.4  | Paging  | 1 | - | 1 |
| 3.5  | Structure of the page Table   | 1 | - | - |
| 3.6  | Segmentation.   | 1 | - | - |
| 3.7  | <b>Virtual Memory:</b> Background; Demand Paging  | 1 | - | - |
| 3.8  | Thrashing   | 1 | - | - |
| <b>Module – 4: Page replacement Algorithms and Design Issues</b> |   |   |   |   |
| 4.1  | <b>Page Replacement Algorithms:</b> The optimal page replacement algorithm                                | 1 | - | - |
| 4.2  | The not recently used page replacement algorithm  | 1 | - | - |
| 4.3  | The first-in first-out  | 1 | - | 1 |
| 4.4  | The second chance page replacement algorithm  | 1 | - | - |
| 4.5  | The clock page replacement algorithm  | 1 | - | - |
| 4.6  | The least recently used.  | 1 | - | 1 |
| 4.7  | <b>Design issues for paging systems:</b> Local versus Global allocation policies, Load control, Page size | 1 | - | - |
| 4.8  | Separate instruction and data spaces, Shared pages, Shared libraries.                                     | 1 | - | - |
| <b>Module – 5: Deadlocks</b>                                     |   |   |   |   |
| 5.1  | <b>Deadlocks: Introduction to deadlocks:</b> Conditions for deadlock,                                     | 1 | - | - |

|                                     |   |           |           |           |
|-------------------------------------|---|-----------|-----------|-----------|
| 5.2                                 | Deadlock modeling.  | 1         | -         | -         |
| 5.3                                 | <b>Deadlock detection and recovery:</b> Deadlock detection with one resource of each type,  | 1         | -         | 1         |
| 5.4                                 | Deadlock detection with multiple resource of each type, Recovery from deadlock.   | 1         | -         | -         |
| 5.5                                 | <b>Deadlock avoidance:</b> Resource trajectories, Safe and Unsafe States, The Banker's algorithm for a single resource, The banker's algorithm for multiple resources.                    | 1         | -         | 1         |
| 5.6                                 | <b>Deadlock prevention:</b> Attacking the Mutual-Exclusion Condition, Attacking the Hold-and-Wait condition, Attacking the No-Preemption condition, Attacking the Circular Wait condition | 1         | -         | -         |
| 5.7                                 | Two-Phase Locking, Starvation.  | 1         |           |           |
| <b>Total No. of Lecture Hours</b>   |   | <b>40</b> | -         | -         |
| <b>Total No. of Tutorial Hours</b>  |   |           | <b>00</b> | -         |
| <b>Total No. of Practical Hours</b> |   |           |           | <b>10</b> |

**Lab Programs:**

| Sl. No. | Programs   | COs |
|---------|--|-----|
| 1       | Implement FCFS (First Come First Serve) scheduling algorithm                                       | CO1 |
| 2       | Implement SJF (Shortest Job First) scheduling algorithm.   | CO1 |
| 3       | Implement Priority Scheduling algorithm.   | CO1 |
| 4       | Implement Round Robin Scheduling algorithm.  | CO1 |
| 5       | Write a program to implement Producer-Consumer Problem using semaphores.                           | CO1 |
| 6       | Implement Memory Allocation methods for Fixed partition using<br>i. First Fit, Worst Fit, Best Fit | CO2 |
| 7       | Write a program to implement FIFO page replacement algorithm.                                      | CO3 |
| 8       | Write a program to implement LRU page replacement algorithm.                                       | CO3 |
| 9       | Write a program to detect Deadlock.  | CO4 |
| 10      | Write a program to implement Banker's algorithm for Deadlock avoidance.                            | CO4 |

**Text Books:**

1. **Modern Operating systems**, 4<sup>th</sup> Edition, Andrew S.Tanenbaum, Herbert Bos, Pearson Education Limited; Global Edition. (MODULE1, MODULE 2, MODULE 4, MODULE 5)
2. **Operating System Principles** – Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Wiley- India, 8th edition (MODULE 3)

**Reference Books:**

1. Operating Systems, William Stallings, PHI, Fourth Edition
2. Operating Systems, Milan Kovic, Tata McGraw Hill, 2001.
3. Operating System Design: v. 1: The Xinu Approach (Prentice-hall Software) Douglas Comer.
4. TUNIX Operating Systems: A Practical Approach, Robert Switzer, Prentice-Hall, 1993.

**Online Resources:**

1. <http://www.nptel.ac.in/courses/106108101/>

**Course Code: BCS304****Credits: 3****L:T:P: 3:0:0****SEE Hours: 3****Course: Data Structures and Applications****CIE: 50 Marks****SEE: 50 Marks****Total Marks: 100**

| Prerequisites if any       | Problem Solving through Programming  |
|----------------------------|--|
| <b>Learning objectives</b> | <ul style="list-style-type: none"> <li>Understand the fundamentals of data structures for problem solving, including linear structures like linked lists, stacks, and queues.</li> <li>Apply non-linear data structures such as trees, sorting algorithms, and hash functions to solve real-world problems.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Apply the concepts of pointers and structures in problem solving.                                | Apply         |
| CO2 | Use different types of linked lists to solve problems.   | Apply         |
| CO3 | Demonstrate stack and queue data structures to solve problems.                                   | Apply         |
| CO4 | Illustrate the operations performed on tree data structures, hash functions for problem solving. | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 2   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO2 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |
| CO3 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |
| CO4 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |

**Mapping Strength: 3 – Strong 2 – Medium 1 – Low**

**Course Structure**

| Nos   | Modules   | No. of Lecture Hours | No. of Tutorial Hours |
|---|---|----------------------|-----------------------|
| <b>Module – 1 Pointers and Data Structures Fundamentals</b> |   |                      |                       |
| 1.1   | <b>Pointers:</b> Introduction   | 1                    | -                     |
| 1.2   | Structures, Nested Structures, Arrays of Structures   | 2                    | -                     |
| 1.3   | Structures and Functions, Self-Referential Structures   | 2                    | -                     |
| 1.4   | <b>Introduction to Data Structures</b> , Classification of data structures, Operations on data structures, Abstract data type | 1                    | -                     |
| <b>Module – 2 Linked Lists</b>                              |   |                      |                       |
| 2.1   | <b>Linked Lists - Singly linked lists</b>   | 3                    | -                     |
| 2.2   | Circular linked lists   | 2                    | -                     |
| 2.3   | Doubly linked lists   | 2                    | -                     |
| 2.4   | Circular doubly linked lists  | 2                    | -                     |

| <b>Module – 3 Stacks and Queues</b>   |  |    |   |
|---------------------------------------|--|----|---|
| 3.1                                   | <b>Stacks:</b> Introduction, Array Representation of Stacks  | 1  | - |
| 3.2                                   | Operations on a Stack, Linked Representation of Stacks   | 1  | - |
| 3.3                                   | <b>Applications of Stacks:</b> Conversion of an infix expression into a postfix expression, Evaluation of a postfix expression, Recursion.       | 3  | - |
| 3.4                                   | <b>Queues:</b> Introduction, Array Representation of Queues, Linked Representation of Queues   | 2  | - |
| 3.5                                   | <b>Types Of Queues:</b> Circular Queue, Priority Queues  | 2  | - |
| <b>Module – 4 Trees</b>               |  |    |   |
| 4.1                                   | <b>Trees:</b> Introduction, Types of Trees   | 1  | - |
| 4.2                                   | <b>Traversing A Binary Tree,</b>   | 1  | - |
| 4.3                                   | Binary Search Trees, Operations on Binary Search Trees   | 3  | - |
| 4.4                                   | <b>Threaded Binary Trees:</b> One-way Threading,   | 1  | - |
| 4.5                                   | AVL Tree, Binary Heaps operations  | 3  | - |
| <b>Module – 5 Sorting and Hashing</b> |  |    |   |
| 5.1                                   | <b>Sorting:</b> Introduction to Sorting, Radix Sort, Heap Sort   | 2  | - |
| 5.2                                   | <b>Hashing and Collision:</b> Introduction, Hash Tables, Different Hash Functions, Collisions, Pros and Cons of Hashing, Applications of Hashing | 5  | - |
| <b>Total No. of Lecture Hours</b>     |  | 40 | - |
| <b>Total No. of Tutorial Hours</b>    |  |    | - |

**Text Books:**

1. Data Structures using C++, Reema Thareja, 2<sup>nd</sup> Edition, 2018, Oxford University Press.
2. Data Structures using C++, Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

**Reference Books:**

1. Richar F Gilberg and Behronz A Forouzan, –Data Structures, A Pseudocode Approach with C”, 2<sup>nd</sup> Edition, 2012, Thomson.
2. Horowitz, Sahni, Anderson-Freed,–Fundamentals of Data Structures in C++, 2<sup>nd</sup> Edition, 2011, Universities Press.

**Online Resources:**

1. NPTEL: Programming and Data structures- <https://nptel.ac.in/courses/106/105/106105085/>
2. Coursera: Data Structures - <https://www.coursera.org/learn/data-structures>
3. Programming & Data structures: <http://nptel.ac.in/courses/106106130/>
4. Programming, Data structures and Algorithms: <http://nptel.ac.in/courses/106106133/>

**Course Code: BCSL305****Credits: 1****L:T:P: 0:0:2****SEE Hours: 2****Course: Data Structures Laboratory****CIE: 50 Marks****SEE: 50 Marks****Total Marks: 100**

| Prerequisites if any       | Fundamentals of C programming   |
|----------------------------|---|
| <b>Learning objectives</b> | <ul style="list-style-type: none"> <li>To gain proficiency in applying advanced C concepts, such as pointers and dynamic memory allocation, for analyzing and solving complex problems.</li> <li>To gain a comprehensive understanding of different data structures, including linked lists, stack, queue, binary trees, sorting algorithms, and hashing, enabling students to analyze, compare, and apply them effectively in various problem-solving contexts.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes   | Bloom's level |
|-----|---|---------------|
| CO1 | Apply concepts, such as pointers and dynamic memory allocation, to analyze and solve complex problems.  | Apply         |
| CO2 | Create and evaluate different types of linked lists to develop robust and efficient applications.   | Apply         |
| CO3 | Demonstrate the ability to analyze, compare, and apply stack, queue, binary tree, sorting, and hashing data structures for problem-solving in various contexts. | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 2   | 2   | -   | -   | -   | -   | -   | 2   | 2    | -    | 3    | 2    | 2    |
| CO2 | 3   | 2   | 3   | -   | -   | -   | -   | -   | 2   | 2    | -    | 3    | 3    | 3    |
| CO3 | 3   | 2   | 3   | -   | -   | -   | -   | -   | 2   | 2    | -    | 3    | 3    | 3    |

**Mapping Strength : 3 – Strong 2 – Medium 1 – Low**

| SL.No | CO  | Experiment   |
|-------|-----|--|
| 1.    | CO1 | (a) Write a program to accept 3 integers and find the maximum among 3 numbers using functions and pointers.<br>(b) Write a C program using pointer for searching the desired element from the array using pointers.<br>(c) Write a program to find the maximum element in each row of the matrix using pointers. |

|    |     |   |
|----|-----|---|
| 2. | CO1 | <p>(a) Write a C program to read and display the Time in specified format. Create a structure called TIME with hour (int), minute(int), second(int) and next(self-referencing pointer ) as its members. Dynamically create two variables of structure TIME and link the first variable to the second one and display it. Write a Display function that takes address of first TIME variable and displays both times in the format h:m:s.</p> <p>(b) Write a C program to read and display the student details. Define a structure 'Student' with fields name(string) , usn(int), marks of 3 subjects (int) and average(float) in it. Store the details of n students in an array of structure 'Student'. Display the details of all students in the descending order their total marks.</p> |
| 3. | CO2 | <p>(a) Write a C program using dynamic variables and pointers, to construct a singly linked list. The operations to be supported are:</p> <ol style="list-style-type: none"> <li>Insert at the front of a list.</li> <li>Insert at any position in the list.</li> <li>Deleting a node based on specified value.</li> <li>Searching a node based on specified value.</li> <li>Displaying all the nodes in the list.</li> </ol> <p>(b) Write a C program to reverse a linked list elements.</p>   |
| 4. | CO2 | <p>Write a C program to support the following operations on a doubly linked list where each node consists of integers.</p> <ol style="list-style-type: none"> <li>Create a doubly linked list by adding each node at the front.</li> <li>Insert a new node to the left of the node whose key value is read as an input.</li> <li>Delete the node of a given data, if it is found, otherwise display appropriate message.</li> <li>Display the contents of the list.</li> </ol>  |
| 5. | CO3 | <p>Write a program to design, Develop and Implement a menu- driven program in C for the following operations on STACK of integers (Array implementation of the stack with maximum size MAX = 4).</p> <ol style="list-style-type: none"> <li>Push an element on to stack.</li> <li>Pop an element from the stack.</li> <li>Check Overflow and Underflow situations on the stack.</li> <li>Display the contents of stack.</li> <li>Exit.</li> </ol> <p>Support the program with appropriate functions for each of the above operations.</p>   |
| 6. | CO3 | <p>a) Write a C program to convert an expression given in -infix   form to -postfix   form using stack concept.</p> <p>b) Check whether a given string is a palindrome or not by using a stack.</p> <p>c) Write a program to find the nth term in the Fibonacci series using recursion.</p>   |
| 7. | CO3 | <p>a) Write a C program to simulate the working of a queues using an array provide and implement the following operations:</p> <ol style="list-style-type: none"> <li>Insert</li> <li>Delete</li> <li>Display</li> </ol> <p>Assume that the size of the queue is 5.</p> <p>b) Write a C program to implement a circular queue using linked lists.</p>   |
| 8. | CO3 | <p>a) Write a C program to construct a binary search tree of integers and also display the elements in the tree using Inorder, Preorder and Postorder traversals.</p> <p>b) Write a C program to find the number of leaf nodes in a BST</p> <p>c) Write a C program to print all root to leaf paths of a BST.</p>   |
| 9. | CO3 | <p>a) Write a C program to sort set of integers using radix sorting technique.</p> <p>b) Write a C program to search using closed hashing.</p>  |

**Text Books:**

1. “Data Structures using C”, Reema Thareja, 2<sup>nd</sup> Edition, 2018, Oxford University Press.
2. Data Structures using C++, Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, 2014, low price edition, Pearson education.

**Reference Books:**

1. Aaron M Tenenbaum, YedidyahLangsam and Moshe J Augenstein, “Data Structures using C”, 2014, low price edition,Pearson education,.
2. Richar F Gilberg and Behronz A Forouzan, “Data Structures, A Pseudocode Approach withC”, 2<sup>nd</sup> Edition,2012, Thomson.
3. Horowitz, Sahni, Anderson-Freed, “Fundamentals of Data Structures in C”, 2<sup>nd</sup> Edition 2011,Universities Press.

**Online Resources:**

1. NPTEL: Programming and Data structures- <https://nptel.ac.in/courses/106/105/106105085/>
2. Coursera: Data Structures - <https://www.coursera.org/learn/data-structures>
3. Programming & Data structures: <http://nptel.ac.in/courses/106106130/>  
Programming, Data structures and Algorithms: <http://nptel.ac.in/courses/106106133/>

**ENGINEERING SCIENCE COURSE (ESC/ ETC/ PLC)****Code: BCS306A****Course: Object Oriented Programming with Java****Credits: 3****CIE: 50 Marks****L:T:P - 2:0:2****SEE: 50 Marks****SEE Hours: 3****Total Marks: 100**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | <b>C programming</b>  |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>1. Distinguish Object-Oriented programming paradigm from Procedure-Oriented Programming</li> <li>2. Use the Java programming language for various programming technologies.</li> </ol> |

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

| <b>COs</b> | <b>Course Outcomes</b>   | <b>Bloom's level</b> |
|------------|--|----------------------|
| CO1        | Demonstrate proficiency in writing simple programs involving branching and looping structures. | Understanding        |
| CO2        | Design a class involving data members and methods for the given scenario.                      | Apply                |
| CO3        | Apply the concepts of inheritance and interfaces in solving real world problems.               | Apply                |
| CO4        | Use the concept of packages and exception handling in solving complex problem                  | Apply                |
| CO5        | Apply the concepts of multithreading in java.  | Apply                |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | -    |
| CO2 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO3 | 3   | 3   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | 2    |
| CO4 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |
| CO5 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1****Course Structure**

| <b>Module – 1 Overview of Java</b> |  | <b>No. of Lecture Hours</b> | <b>No. of Tutorial Hours</b> | <b>No. of Practical Hours</b> |
|------------------------------------|--|-----------------------------|------------------------------|-------------------------------|
| 1.1                                | <b>An Overview of Java:</b> Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords). | 1                           | -                            | -                             |
| 1.2                                | <b>Data Types, Variables, and Arrays:</b> The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays                      | 1                           | -                            | -                             |
| 1.3                                | <b>Operators:</b> Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ?  | 1                           | -                            | -                             |

|   |  |    |   |    |
|---|--|----|---|----|
|   | Operator, Operator Precedence, Using Parentheses.  |    |   |    |
| 1.4   | <b>Control Statements:</b> Java's Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop), Jump Statements (Using break, Using continue, return). | 1  | - | 2  |
| <b>Module – 2: Classes and Objects</b>          |  |    |   |    |
| 2.1   | <b>Introducing Classes:</b> Class Fundamentals, Declaring Objects, Assigning Object Reference Variables  | 1  | - | 2  |
| 2.2   | Introducing Methods, Constructors, This Keyword, Garbage Collection.   | 1  | - | 2  |
| 2.3   | <b>Methods and Classes:</b> Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Access Control  | 1  | - | -  |
| 2.4   | Understanding static keyword, introducing final, Introducing super keyword.  | 2  | - | -  |
| <b>Module – 3: Inheritance &amp; Interfaces</b> |  |    |   |    |
| 3.1   | <b>Inheritance:</b> Inheritance Basics, Creating a Multilevel Hierarchy.   | 1  | - | -  |
| 3.2   | When Constructors Are Executed, Method Overriding  | 1  |   |    |
| 3.3   | Using Abstract Classes   | 1  | - | -  |
| 3.4   | <b>Interfaces:</b> Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.   | 1  | - | 2  |
| <b>Module – 4: Packages &amp; Exceptions</b>    |  |    |   |    |
| 4.1   | <b>Packages:</b> Packages, Packages and Member Access, Importing Packages.   | 1  | - | 2  |
| 4.2   | <b>Exceptions:</b> Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions   | 1  | - | 2  |
| 4.3   | Using try and catch, Multiple catch Clauses, Nested try Statements,  | 1  | - | -  |
| 4.4   | Throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses.  | 1  | - | -  |
| <b>Module – 5: Threads</b>                      |  |    |   |    |
| 5.1   | <b>Multithreaded Programming:</b> The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads,  | 1  | - | 2  |
| 5.2   | Using isAlive() and join(), Thread Priorities  | 1  | - | 1  |
| 5.3   | Synchronization  | 1  | - | -  |
| <b>Total No. of Lecture Hours</b>               |  | 25 |   |    |
| <b>Total No. of Tutorial Hours</b>              |  |    | - | -  |
| <b>Total No. of Practical Hours</b>             |  |    |   | 15 |

**Integrated lab program List**

| Sl. No. | Programs  | COs |
|---------|---|-----|
| 1.      | For Practice only – not added for SET<br>( Input -using Command Line Arguments)<br>a) Develop a JAVA program to display the message.<br>b) Develop a JAVA program to implement basic arithmetic operations.<br>c) Develop a JAVA program to count and display the number of arguments.  | CO1 |
| 2.      | Develop a JAVA program to add TWO matrices of suitable order N (The value of N should be read from command line arguments).   | CO1 |
| 3.      | A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary (percent) increases the salary by the given percentage. Develop the Employee class and suitable main method for demonstration.   | CO2 |
| 4.      | A class called MyPoint, which models a 2D point with x and y coordinates, is designed as follows: <ul style="list-style-type: none"> <li>• Two instance variables x (int) and y (int).</li> <li>• A default (or "no-arg") constructor that construct a point at the default location of (0, 0).</li> <li>• A overloaded constructor that constructs a point with the given x and y coordinates.</li> <li>• A method setXY() to set both x and y.</li> <li>• A method getXY() which returns the x and y in a 2-element int array.</li> <li>• A toString() method that returns a string description of the instance in the format "(x, y)".</li> <li>• A method called distance(int x, int y) that returns the distance from this point to another point at the given (x, y) coordinates</li> <li>• An overloaded distance(MyPoint another) that returns the distance from this point to the givenMyPoint instance (called another)</li> <li>• Another overloaded distance() method that returns the distance from this point to the origin (0,0)</li> </ul> Develop the code for the class MyPoint. Also develop a JAVA program (called TestMyPoint) to test all the methods defined in the class. | CO2 |
| 5.      | Develop a JAVA program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate polymorphism concepts by developing suitable methods, defining member data and main program.  | CO3 |
| 6.      | Develop a JAVA program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.  | CO3 |
| 7.      | Develop a JAVA program to raise a custom exception (user defined exception) for DivisionByZero using try, catch, throw and finally.   | CO4 |
| 8.      | Develop a JAVA program to create a package named mypack and import & implement it in a suitable class.  | CO4 |
| 9.      | Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500milliseconds).   | CO5 |
| 10.     | Develop a program to create a class MyThread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.  | CO5 |

**Text book:**

1. Herbert Schildt, *Java The Complete Reference-Eleventh Edition*, McGraw Hill; 12<sup>th</sup> Edition.

**Reference Book:**

1. Dr. R. Nageswara Rao, Core Java, An Integrated Approach, Dream tech Press, 2016.
2. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008
3. Rajkumar Buyya, S Thamarai selvi, xing chenchu, Object oriented Programming withjava, Tata McGraw Hill Education Private Limited, 2009
4. Richard A Johnson, An Introduction to Java Programming and Object-OrientedApplication Development, Delmar Cengage Learning, 2007.
5. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML,2<sup>nd</sup> Edition, Pearson Education/ PHI, 2007.
6. E Balagurusamy , Programming with Java, 6th Edition, by, Mar-2019, McGraw Hill Education, ISBN: 9789353162337.
7. Bruce Eckel Thinking in Java, Fourth Edition, Prentice Hall, 2006([https://sd.blackball.lv/library/thinking\\_in\\_java\\_4th\\_edition.pdf](https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf))

**Online Resources:**

1. Java Tutorial: <https://www.geeksforgeeks.org/java/>
2. Introduction To Programming In Java (by Evan Jones, Adam Marcus and Eugene Wu):
3. <https://ocw.mit.edu/courses/6-092-introduction-to-programming-in-java-january-iap-2010/>
4. Java Tutorial: <https://www.w3schools.com/java/>
5. Java Tutorial: <https://www.javatpoint.com/java-tutorial>
6. <https://www.udacity.com/course/java-programming-basics--ud282>

**Course Code: BCS306B****Credits: 3****L:T:P - 2:0:2****SEE Hours: 3****Course: Object Oriented Programming with C++****CIE: 50 Marks****SEE: 50 Marks****Total Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | <b>C Programming</b>   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To provide the knowledge of fundamental principles of Object-Oriented Programming.</li> <li>To introduce the concepts of C++ language such as class, object, inheritance, overloading, virtual functions, STL etc.</li> </ol> |

**Course Outcomes:**

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

| COs        | Course Outcomes  | Bloom's level |
|------------|--|---------------|
| <b>CO1</b> | Explain principles of Object-Oriented Programming using class and objects.   | Understanding |
| <b>CO2</b> | Illustrate the concepts of function and operator overloading.  | Apply         |
| <b>CO3</b> | Demonstrate reusability using inheritance and virtual functions.   | Apply         |
| <b>CO4</b> | Apply function template and class template, STL and Exception Handling using real world problems for C++ programs. | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | 2   | -   | 2   | -   | -   | -   | 2   | 2    | -    | 2    | -    | -    |
| CO2 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | 2   | 2    | -    | 2    | 3    | 2    |
| CO3 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | 2   | 2    | -    | 2    | 3    | 2    |
| CO4 | 3   | 3   | 2   | -   | 2   | -   | -   | -   | 2   | 2    | -    | 2    | 3    | 2    |

Mapping strength:                      3 – Strong      2 – Medium      1 – Low

**Course Structure**

| Nos                                  | Modules  | No of Lecture Hours | No of Tutorial Hours | No of Practical Hours |
|--------------------------------------|--|---------------------|----------------------|-----------------------|
| <b>Module-1: Classes and Objects</b> |  |                     |                      |                       |
| 1.1                                  | <b>Classes and Objects:</b> An overview of C++, classes and objects, Relationship of Structure, Union and Class in C++   | 1                   | -                    | -                     |
| 1.2                                  | Friend Functions, Friend Classes, Inline Functions- Defining Inline Functions Within a Class   | 1                   | -                    | -                     |
| 1.3                                  | Constructors and Destructors, Parameterized Constructors- Constructors with One Parameter: A Special Case, copy constructor.   | 1                   | -                    | -                     |
| 1.4                                  | Static Class Members- Static Data Members and Static Member Functions, When Constructors and Destructors Are Executed, Passing Objects to Functions, Returning Objects.                            | 1                   | -                    | 1                     |
| 1.5                                  | <b>Arrays, pointers, References, and the Dynamic Allocation Operators:</b> <code>_this</code> Pointer, References – Reference Parameters, Passing References to Objects and Returning References., | 1                   | -                    | 1                     |

|                                       |   |           |   |           |
|---------------------------------------|---|-----------|---|-----------|
| 1.6                                   | C++_s Dynamic Allocation Operators- Initializing Allocated Memory, Allocating Arrays and Allocating Objects   | 1         | - | 1         |
| <b>Module-2: Overloading</b>          |   |           |   |           |
| 2.1                                   | <b>Function Overloading, Copy Constructor, and Default Arguments:</b> Function Overloading  | 1         | - | -         |
| 2.2                                   | Copy Constructor, Default Function Arguments, Default Argument vs. Overloading.   | 1         | - | 1         |
| 2.3                                   | <b>Operator Overloading:</b> Creating a Member Operator Function- Creating Prefix and Postfix Forms of the Increment and Decrement Operators  | 1         | - | 1         |
| 2.4                                   | Operator Overloading Using a Friend Function – Using a Friend to Overload ++ or --, Friend operator Functions Add Flexibility, Overloading << and >>.   | 2         | - | 1         |
| <b>Module – 3: Inheritance</b>        |   |           |   |           |
| 3.1                                   | <b>Inheritance:</b> Base-Class Access Control, Inheritance and protected Members- Protected Base-Class Inheritance, Inheritance Multiple Base Classes   | 2         | - | 1         |
| 3.2                                   | Constructors, Destructors, and Inheritance- When Constructors and Destructors Are Executed, Passing Parameters to Base-Class Constructors. Granting Access, Virtual Base Classes.             | 1         | - | 1         |
| 3.3                                   | <b>Virtual Functions and Polymorphism:</b> Virtual Functions- Calling a Virtual Function Through a Base- Class Reference, The Virtual Attribute vs. Inherited,                                | 1         | - | 1         |
| 3.4                                   | Virtual Functions Are Hierarchical, Pure Virtual Functions- Abstract Classes, Using Virtual Functions, Early vs. Late Binding   | 1         | - | 1         |
| <b>Module – 4: Generic functions</b>  |   |           |   |           |
| 4.1                                   | <b>Templates:</b> Generic Functions- A Function with Two Generic Types, Explicitly Overloading a Generic Function,  | 1         | - | -         |
| 4.2                                   | Overloading a Function Template, Using Standard Parameters with Template Functions, Generic Function Restrictions.  | 1         | - | 1         |
| 4.3                                   | <b>Applying Generic Function:</b> A generic Sort.   | 2         | - | -         |
| 4.4                                   | <b>Applying Generic Classes:</b> An Example with Two Generic Data Types, A Generic Array Class, Using Non-Type Arguments with Generic classes, Using Default Arguments with Template Classes. | 1         | - | 1         |
| <b>Module – 5: Exception handling</b> |   |           |   |           |
| 5.1                                   | <b>Exception Handling:</b> Fundamentals, Handling derived class Exception   | 1         | - | 1         |
| 5.2                                   | Exception Handling options.   | 1         | - | 1         |
| 5.3                                   | <b>Introducing the Standard Template Library:</b> An Overview of the STL, Container Classes,  | 1         | - | -         |
| 5.4                                   | General Theory of Operation, Vector container class, Algorithms.  | 1         | - | 1         |
| <b>Total No. of Lecture Hours</b>     |   | <b>25</b> | - | -         |
| <b>Total No. of Tutorial Hours</b>    |   |           | - | -         |
| <b>Total No. of Practical Hours</b>   |   |           |   | <b>15</b> |

| Integrated Lab Programs |   |     |
|-------------------------|---|-----|
| Sl. No.                 | Programs  | COs |
| 1.                      | <p>a. Write a C++ program to read three numbers from the keyboard and display the largest value on the screen.</p> <p>b. Write a C++ program to check whether the given number is an Armstrong number or not. (Hint: Armstrong number <math>153=1^3 + 5^3 + 3^3</math> ).</p> <p>c. Write a C++ program to find and display Factorial of a number. (Using an iterative method or recursion).</p> <p>d. Write a C++ program to print the ASCII value of a character.</p> <p>Write a C++ program that will accept an array of numbers and display the number of times the given number occurred in the array.</p>   | CO1 |
| 2.                      | <p>a) Convert time from HH:MM:SS format to seconds using class in C++.</p> <p>b) A phone number, such as (044) 234-8900, can be thought of as having three parts: the area code (044), the exchange (234) and the number (8900). Write a program that uses a class to store these three parts of a phone number separately. Call the class phone. Create two class objects of type phone. Initialize one, and have the user input a number for the other one. Display both the numbers.</p>   | CO1 |
| 3.                      | <p>Create two classes DM and DB which store the value of distances. DM stores distance in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.</p>  | CO1 |
| 4.                      | <p>Write a C++ program to create a class Rectangle with data members: length, breadth, area and member functions as:</p> <p>Rectangle &amp;setDimensions(const int&amp;, const int&amp;) – sets the length and breadth of arguments using constant references. Use this pointer to return the resized Rectangle by reference.</p> <p>i. int computeArea() – to compute and returns area of a rectangle.</p> <p>ii. int computePerimeter() – to compute and return perimeter of a rectangle.</p> <p>iii. Two constructors, default constructor to initialize data members to zero and an overloaded constructor as arguments with breadth having a default value.</p> <p>Define all the member functions outside the class. Create objects of Rectangle type and test all the functions. Delete the objects before the program terminates.</p> | CO1 |
| 5.                      | <p>Write a C++ program to create a class called Complex and implement the following overloading member functions that return a Complex number after performing addition of input arguments.</p> <ul style="list-style-type: none"> <li>● Complex ADD (int a, Complex s) – where a is an integer (real part) and s is a complex number.</li> <li>● Complex ADD (Complex &amp;s1, Complex &amp;s2) Create objects of Rectangle type on heap utilizing both constructors. Use the member functions to modify the rectangle size, compute area, perimeter. Display the area &amp; perimeter of each rectangle. Delete the objects before program termination.</li> </ul>  | CO2 |

|     |  |     |
|-----|--|-----|
| 6.  | <p>Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators <code>+=</code> and <code>-=</code>.<br/> <code>S1 = S1 + element</code>; where S1 is the object of class STACK and element is an integer to be pushed on the top of stack.<br/> <code>int element = S1--</code>; where S1 is the object of class STACK. <code>--</code> operator pops the top element.<br/> Handle the STACK empty and full conditions and also display the contents after every operation by overloading <code>&lt;&lt;</code> operator.</p>   | CO2 |
| 7.  | <p>Write a C++ program to read and print Employee information (name, empID, gender) with Department (deptName, workAssigned) and with Loan information (loanDetails, loanAmt) using hierarchical inheritance.</p>  | CO3 |
| 8.  | <p>Write a C++ program to design a Student class representing USN and a Test class representing the scores of the student in various subjects and a Sports class representing the score in sports. The Sports and Test classes is inherited by Result class having the functionality to add the scores and display the final result of a student.</p>  | CO3 |
| 9.  | <p>Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students from UG and PG. Find the average age for all UG and PG students separately.</p>   | CO3 |
| 10. | <p>Implement class Shape with the following specification:<br/> <pre>class Shape{ protected: float area, perimeter; public: Shape(); virtual void initialize()=0; virtual float computeArea()=0; virtual float computePerimeter()=0; virtual ~Shape(); };</pre> <p>Implement 2 classes Triangle and Rectangle publicly derived from class Shape, with suitable data members. Implement all the functions derived from class shape in each of the derived classes. Write a C++ program to create objects of each of the derived class and assign to the base class (Shape) type pointer/reference. Demonstrate runtime polymorphism by calling the functions of the derived class objects by using the base class pointer/reference.</p> </p> | CO3 |
| 11. | <p>Write two function templates in C++ to</p> <ol style="list-style-type: none"> <li>Sort the numbers.</li> <li>To search a given number</li> </ol> <p>Demonstrate the above functions on an array of integers and double.</p> <ol style="list-style-type: none"> <li>A point on the 2D can be represented by two numbers: an x co-ordinate and a y co-ordinate. The sum of two points can be defined as a new point whose x co-ordinate is the sum of x co-ordinates of both points and same for y co-ordinates. Using function template, find the third point in C++.</li> </ol>   | CO4 |
| 12. | <ol style="list-style-type: none"> <li>Write a simple calculator using class template in C++</li> <li>Write a program implementing stack and its operations using template class.</li> </ol>   | CO4 |

|     |  |     |
|-----|--|-----|
| 13. | <p>Write a C++ program with the following:</p> <ul style="list-style-type: none"> <li>. A function to read two double type numbers from the keyboard.</li> <li>a. A function to calculate the division of these two numbers.</li> <li>b. A try block to throw an exception when a wrong type of data is keyed in.</li> <li>c. A try block to detect and throw an exception if the condition —divide-by-zero occurs.</li> </ul> <p>Appropriate catch block to handle the exceptions thrown.</p>   | CO4 |
| 14. | <ul style="list-style-type: none"> <li>a. Perform these basic vector operation using Standard Template Library: <ul style="list-style-type: none"> <li>. Find the number of elements in the vector.</li> <li>i. Check whether the vector is empty or not.</li> <li>ii. Insert some elements into the vector.</li> <li>iii. Remove the element at a particular position.</li> <li>iv. Find the index of a particular element in a vector.</li> </ul> </li> <li>b. Make a vector of random numbers and sort it in descending order using STL and also find its sum.</li> </ul> | CO4 |

**Text Books:**

1. C++ The Complete Reference, Herbert Schildt, TMH, McGraw-Hill, 4th Edition, 2017.

**Reference Books:**

2. The C++ programming language, Bjarne Stroustrup, Pearson Education, 3<sup>rd</sup> Edition, 2013.
3. C++ Primer, Stanley B. Lippman and Josee Lajore, Addison Wesley, 3<sup>rd</sup> Edition, 2014.
4. WEBLINK: <http://www.cplusplus.com/>

**Online Resources:**

1. NPTEL: Programming in modern C++ [https://nptel.ac.in/courses/noc22\\_cs43](https://nptel.ac.in/courses/noc22_cs43)  
Coursera: Programming in C++- <https://www.coursera.org/specializations/hands-on-cpp>

**Code: BCS306D**  
**Credits: 3**  
**L:T:P - 2:0:2**  
**SEE Hours: 3**

**Course: JavaScript**  
**CIE: 50 Marks**  
**SEE: 50 Marks**  
**Total Marks:100**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | Programming Fundamentals  |
| <b>Learning objectives</b>  | <p>Learners will be able to:</p> <ul style="list-style-type: none"> <li>• Compose syntactically correct JavaScript programs, including a functional "Hello World" example. They will confidently apply essential lexical elements such as comments, literals, and identifiers.</li> <li>• Proficiently manipulate JavaScript's core data structures. They will aptly create, modify, and query object properties, employ functions to solve tasks, comprehend closures for encapsulation, and effectively utilize the JavaScript Standard Library to solve challenges involving sets, maps, regular expressions, and error handling.</li> </ul> |

### Course Outcomes:

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Apply the knowledge of JavaScript lexical elements, data types, values, and variables to write programs.   | Apply         |
| CO2 | Design and develop interactive web applications using object-oriented concepts, various control structures for statements and loops, and effectively manage data using arrays and objects. | Analyze       |
| CO3 | Create programs which uses Document Object Model, manipulate web documents and styles dynamically to enhance user experiences.   | Apply         |

### Mapping with POs and PSOs:

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 2   | -   | -   | 3   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |
| CO2 | 3   | 2   | -   | -   | 3   | -   | -   | -   | -   | -    | -    | 2    | 3    | 3    |
| CO3 | 3   | 2   | -   | -   | 3   | -   | -   | -   | -   | -    | -    | 3    | 3    | 3    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

### Course Structure

|  | Modules  | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|--|--|----------------------|-----------------------|------------------------|
| <b>Module – 1: Introduction to JavaScript</b>            |  |                      |                       |                        |
| 1.1  | Introduction to JavaScript: Exploring JavaScript, Hello World, A Tour of JavaScript  | 1                    | -                     | 1                      |
| 1.2  | Lexical Structure: The Text of a JavaScript Program, Comments, Literals, Identifiers and Reserved Words, Unicode, Optional Semicolons  | 2                    | -                     | 1                      |
| 1.3  | Types, Values, and Variables: Numbers, Text, Boolean Values, null and undefined, Symbols, The Global Object, Immutable Primitive Values and MutableObject References, Type Conversions | 2                    | -                     | 1                      |
| <b>Module – 2: Expressions, Operators and Statements</b> |  |                      |                       |                        |

|  |   |           |          |           |
|--|---|-----------|----------|-----------|
| 2.1  | Expressions and Operators: Primary Expressions, Object and Array Initializers, Function Definition Expressions, Property Access Expressions, Invocation Expressions, Arithmetic Expressions, Relational Expressions, Logical Expressions, Assignment Expressions, Evaluation Expressions, | 3         | -        | 2         |
| 2.2  | Statements: Expression Statements, Compound and Empty Statements, Conditionals, Loops, Jumps, Declarations  | 2         | -        | 1         |
| <b>Module – 3: Objects, Arrays and functions</b> |   |           |          |           |
| 3.1  | Objects: Introduction to Objects, Creating Objects, Querying and Setting Properties, Deleting Properties, Testing Properties, Enumerating Properties, Extending Objects, Serializing Objects, Object Methods  | 2         | -        | 1         |
| 3.2  | Arrays: Creating Arrays, Reading and Writing Array Elements, Sparse Arrays, Array Length, Adding and Deleting Array Elements, Iterating Arrays, Multidimensional Arrays, Array Methods, Array-Like Objects, Strings as Arrays   | 2         | -        | 1         |
| 3.3  | Functions: Defining Functions, Invoking Functions, Function Arguments and Parameters, Functions as Values, Functions as Namespaces, Closures, Function Properties, Methods and Constructor  | 2         | -        | 1         |
| <b>Module – 4: Classes</b>                       |   |           |          |           |
| 4.1  | Classes: Classes and Prototypes, Classes and Constructors, Classes with the class Keyword, Adding Methods to Existing Classes, Subclasses   | 2         | -        | 2         |
| 4.2  | The JavaScript Standard Library: Sets and Maps, Typed Arrays and Binary Data, Pattern Matching with Regular Expressions, Dates and Times, Error Classes   | 3         | -        | 1         |
| <b>Module – 5: Iterators and Generators</b>      |   |           |          |           |
| 5.1  | Iterators and Generators: How Iterators Work, Implementing Iterable Objects, Generators   | 2         | -        | 1         |
| 5.2  | JavaScript in Web Browsers: Web Programming Basics, Events, Scripting Documents, Scripting CSS, Document Geometry and Scrolling, Web Components   | 2         | -        | 2         |
| <i>Total No. of Lecture Hours</i>                |   | <b>25</b> | -        | -         |
| <i>Total No. of Tutorial Hours</i>               |   |           | <b>0</b> | -         |
| <i>Total No. of Practical Hours</i>              |   |           |          | <b>15</b> |

| SL.N<br>O | COs | List of Experiments  |
|-----------|-----|--|
| 1.        | CO1 | <b>Type Conversions</b><br>a. Create a program that demonstrates different types of type conversions in JavaScript.<br>b. Convert a number to a string, a string to a number, and a boolean to a number.                   |
| 2.        | CO2 | <b>Object Properties</b><br>Create an object representing a person with properties like name, age, and occupation. Access and modify these properties using both dot notation and bracket notation.                        |
| 3.        | CO2 | <b>Array Manipulation</b><br>Write a program that demonstrates various array operations. Create an array of numbers, add new elements to it, remove elements, and iterate through the array using loops and array methods. |
| 4.        | CO2 | <b>Function and Closure</b><br>Define a function that takes two parameters and returns their sum. Create a closure that uses a variable from its containing function's scope.  |
| 5.        | CO2 | <b>Classes and Inheritance</b><br>Create a class representing a basic shape. Extend the class to create subclasses like Circle and Square. Add methods to calculate their areas.   |
| 6.        | CO2 | <b>Error Handling</b><br>Write a program that uses a try-catch block to handle an error. Attempt to access a property of an undefined object and handle the resulting exception.   |
| 7.        | CO1 | <b>String Manipulation</b><br>Write a program that takes a sentence as input and reverses the order of words using string manipulation techniques.   |
| 8.        | CO1 | <b>Working with Dates</b><br>Create a program that displays the current date and time. Format it in a user-friendly way, such as "August 10, 2023, 3:30 PM".   |
| 9.        | CO3 | <b>Event Handling and DOM Manipulation</b><br>Create a simple to-do list. Allow users to add items, mark items as completed, and remove items using event listeners and DOM manipulation.                                  |
| 10.       | CO2 | <b>Generators</b><br>Implement a simple generator that yields a sequence of even numbers up to a specified limit.  |
| 11.       | CO2 | <b>Regular Expressions for Data Validation</b><br>Create a form with input fields for email, password, and phone number. Use regular expressions to validate the user's input in real-time as they type.                   |
| 12.       | CO3 | <b>Creating a Basic Web Component</b><br>Explore creating a basic web component using the `CustomElementRegistry`. Build a custom element for a simple counter that users can interact with.                               |

|     |                     |   |
|-----|---------------------|---|
| 13. | CO1,<br>CO2,<br>CO3 | <p><b>Project-based experiments (Students can choose any one of the following) (CO1, CO2 and CO3)</b></p> <p><b>a. Task Manager Web App</b> - Build a task manager web application using HTML, CSS, and JavaScript. The app should have the following features:</p> <ul style="list-style-type: none"> <li>● Task List: Display a list of tasks with their titles and due dates.</li> <li>● Add Task: Allow users to add new tasks with titles and due dates.</li> <li>● Mark Complete: Users can mark tasks as completed, which will visually differentiate them from incomplete tasks.</li> <li>● Delete Task: Implement a way to delete tasks from the list.</li> <li>● Local Storage: Store tasks in the browser's local storage so that they persist even after the user refreshes the page.</li> </ul> <p>This project will require you to work with HTML for creating the structure, CSS for styling, and JavaScript for handling user interactions, managing tasks, and storing data.</p> <p><b>OR</b></p> <p><b>b. Interactive Quiz Game</b> - Create an interactive quiz game using HTML, CSS, and JavaScript. The app should have the following features:</p> <ul style="list-style-type: none"> <li>● Quiz Questions: Prepare a set of multiple-choice questions along with their correct answers.</li> <li>● Display Questions: Display one question at a time with its options.</li> <li>● Answer Selection: Allow users to select an answer from the provided options.</li> <li>● Score Tracking: Keep track of the user's score based on the number of correct answers.</li> <li>● Results: After all questions are answered, display the user's total score and a congratulatory message.</li> </ul> <p>To make the quiz more engaging, you could also implement a countdown timer for each question. Additionally, consider using animations to transition between questions.</p> |
|-----|---------------------|---|

**Textbook:**

1. JavaScript: The Definitive Guide, David Flanagan, 7th Edition (Released May 2020), O'Reilly Media,

**Reference Book:**

1. JavaScript from Beginner to Professional, Laurence Lars Svekis, Maaike van Putten, Rob Percival, PacktPublishing Limited, 2021.

**Online Resources:**

1. <https://ocw.mit.edu/courses/6-170-software-studio-spring-2013/pages/lecture-notes/>
2. <https://web.stanford.edu/class/cs98si/courseinfo.html>

**Course Code: BSCK307****Course: Social Connect and Responsibility****Credits: 1****CIE: 100 Marks****L:T:P - 0:0:2****Total Marks: 100**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | NIL   |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>To make students understand and appreciate the important societal/ environmental issues like conservation/ sustainability/ waste management and the like and inculcate ethical responsibility towards the same.</li> <li>Provide a formal platform for students to communicate and connect to their surroundings and enable them to have a responsible connection with society.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes                                | Bloom's level |
|-----|--|---------------|
| CO1 | Understand social responsibility               | Understand    |
| CO2 | Practice sustainability and creativity         | Apply         |
| CO3 | Demonstrate planning and organizational skills | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | -   | -   | -   | -   | -   |     | 2   | 1   | 3   | 1    | 1    | -    | 1    | 1    |
| CO2 | -   | -   | -   | -   | -   |     | 2   | 1   | 3   | 2    | 2    | -    | 1    | 1    |
| CO3 | -   | -   | -   | -   | -   |     | 2   | 1   | 3   | 3    | 3    | -    | 2    | 1    |

S – Strong (3) M – Medium (2) L – Low (1)

**Course Structure**

| Nos.   | Module Name   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|--|---|----------------------|-----------------------|------------------------|
| <b>Module – 1: Plantation /Heritage Walk</b> |   |                      |                       |                        |
| 1.1  | Plantation and adoption of a tree: Plantation of a tree that will be adopted for four years by a group of B.E. students. They will also make an excerpt either as a documentary or a photo blog describing the plant's origin, its usage in daily life, and its appearance in folklore and literature ( <b>After Plantation of specific tree like sandalwood or any medicinal plants mention the significance of the plants one to three plants</b> ) | 01                   | 0                     | 04                     |
| <b>OR</b>                                    |   |                      |                       |                        |
| 1.1  | Heritage walk and crafts corner: Heritage tour, <b>knowing the history and culture of the city</b> , connecting to people around through their history, knowing the city and its craftsmen, photo blog and documentary on evolution and practice of various craft forms.  | 01                   | 0                     | 04                     |
| <b>Module – 2: Organic farming/Food walk</b> |   |                      |                       |                        |
| 2.1  | Organic farming and waste management: usefulness of organic farming, wet <b>waste management in</b> neighboring villages, and implementation <b>in the campus</b> .   | 01                   | 0                     | 04                     |
| <b>OR</b>                                    |   |                      |                       |                        |

|                                       |  |           |   |    |
|---------------------------------------|--|-----------|---|----|
| 2.2                                   | Food Walk; City's culinary practices, food lore, and indigenous materials of the region used in cooking.   | 01        | 0 | 04 |
| <b>Module – 3: Water Conservation</b> |  |           |   |    |
| 3.1                                   | <b>Water Conservation:</b> knowing the present practices in the surrounding villages and implementation in the campus, <b>documentary or photo blog presenting the current practices</b> | 01        | 0 | 04 |
| <b>Total No. of Hours</b>             |  | <b>15</b> |   |    |

**Activities:**

Jamming session, open mic, and poetry: Platform to connect to others. Share the stories with others. Share the experience of Social Connect: Exhibit the talent like playing instruments, singing, one-act play, art-painting, and fine art.

**Pedagogy:**

The pedagogy will include interactive lectures, inspiring guest talks, field visits, social immersion and a course project.

Applying and synthesizing information from these sources to define the social problem to address and try to arrive at the solution through the course project, of your group.

Social immersion with NGOs/social sections will be a key part of the course.

**Guideline for Assessment Process: Continuous Internal Evaluation (CIE)**

After completion of the social connect, the student shall prepare, with daily diary as reference, a comprehensive report in consultation with the mentor/s to indicate what he has observed and learnt in the social connect period. The report should be signed by the mentor. The course shall be evaluated on the basis of the following criteria and/or other relevant criteria pertaining to the activity completed / based on the rubrics approved by the DC.

| Components   | Marks |
|--|-------|
| Marks allotted for the diary                         | 20    |
| Planning and scheduling the social connect           | 10    |
| Information/Data collected during the social connect | 20    |
| Analysis of the information/data and report writing  | 30    |
| Final presentation                                   | 20    |
| Total  | 100   |

**ABILITY ENHANCEMENT COURSE****Code: BCS358A****Course: Data Analytics with Excel****Credits: 1****CIE: 50 Marks****L:T:P - 1:0:0****SEE: NA****SEE Hours: NA****Total Marks:50**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Basic Computer Skills, Fundamental Excel Skills  |
| <b>Learning objectives</b>  | The learning objectives for a data analytics course with Excel typically revolve around equipping students with the knowledge and skills to effectively analyze and interpret data using Excel as a tool |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | To familiarize oneself with Excel's Basic features                     | Understanding |
| CO2 | To gain skills on data visualization for data analysis using MS Excel. | Analyze       |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | -   | -   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | -    | -    |
| CO2 | 3   | 3   | 3   | 2   | 3   | -   | -   | -   | -   | -    | -    | 2    | -    | -    |

**Mapping Strength: Strong– 3      Medium – 2      Low – 1**

**Course Structure**

| Sl. No  | Module   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|---|--|----------------------|-----------------------|------------------------|
| <b>Module – 1: I Basics of MS Excel</b>               |  |                      |                       |                        |
| 1.1   | Features of MS   | 2                    | -                     | -                      |
| 1.2   | Excel Worksheets and Workbooks: Labeling and Naming Worksheets and Workbooks, Adding, Deleting and Saving Worksheets and Workbooks, Reposition Worksheets, Inserting, Deleting, and Renaming Worksheets, Copy Worksheets, Printing a Workbook, Formatting a Worksheet, Adding Elements to a Workbook, Protecting Worksheet and Workbook. | 2                    | -                     | -                      |
| <b>Module – 2: Data Representation using MS Excel</b> |  |                      |                       |                        |
| 2.1   | Import external data, Creating a Table, Sorting Data into a Table, Data Validation, Consolidation  | 1                    | -                     | -                      |
| 2.2   | Defining Names in MS Excel, Macros: View Macros, Record Macros   | 2                    | -                     | -                      |
| 2.3   | Formulas and Functions: Creating a Formula, Formula Auditing, Meaning and Advantages of functions, Insert function, Use relative References, Mathematical Functions, Statistical Functions, Date & Time Functions  | 2                    | -                     | -                      |
| <b>Module – 3: Data Visualization and Analysis</b>    |  |                      |                       |                        |
| 3.1   | Charts: Chart elements: Titles, legend, data labels, creating a New Chart, Formatting the Chart, Types of charts, Using Chart Templates  | 2                    | -                     | -                      |
| 3.2   | PivotTables: Creating a PivotTable, Filtering and Sorting a PivotTable, Using Slicers to manipulate PivotTables, Creating a PivotChart   | 2                    | -                     | -                      |
| 3.3   | Filtering Data: Creating a Custom AutoFilter, Using an Advanced Filter. Data Group, Ungroup and Subtotals  | 2                    | -                     | -                      |
| <b>Total No. of Lecture Hours</b>                     |  | <b>15</b>            | -                     | -                      |
| <b>Total No. of Tutorial Hours</b>                    |  |                      | <b>00</b>             | -                      |
| <b>Total No. of Practical Hours</b>                   |  |                      |                       | <b>00</b>              |

**Textbook:**

1. "Microsoft Excel 2019 Step by Step" by Curtis Frye
2. "Excel 2019 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach

**Reference Book:**

1. "Microsoft Excel Data Analysis and Business Modeling" by Wayne L. Winston:
2. "Excel 2019 All-in-One For Dummies" by Greg Harvey:

**Online Resources:**

1. [https://www.academia.edu/42074058/Excel\\_2019\\_BIBLE](https://www.academia.edu/42074058/Excel_2019_BIBLE)
2. <https://ptgmedia.pearsoncmg.com/images/9780735681019/samplepages/9780735681019.pdf>

**Code: BCS358B****Credits: 1****L:T:P - 1:0:0****SEE Hours: NA****Course: Data Analytics with R****CIE: 50 Marks****SEE: NA****Total Marks:50**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Basics of any programming language and knowledge of statistics and mathematics.  |
| <b>Learning objectives</b>  | 1. Understand the basics of data analytics and its importance in decision-making processes.<br>2. Utilize the R programming language for data manipulation and analysis tasks. |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Apply the R programming language to perform data analysis tasks. | Analysis      |
| CO2 | Apply statistical concepts and techniques to analyze data.       | Analysis      |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | 2   | 2   | 3   | 3   | -   | -   | -   | -   | -    | -    | 3    | 2    | 2    |
| CO2 | 2   | 2   | 2   | 3   | 3   | -   | -   | -   | -   | -    | -    | 3    | 2    | 2    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

**Course Structure**

| Sl. No  | Module   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|---|--|----------------------|-----------------------|------------------------|
| <b>Module – 1: Introduction to R</b>              |  |                      |                       |                        |
| 1.1   | Introduction, Downloading and Installing Data Types in R.  | 1                    | -                     | -                      |
| 1.2   | RIDEs and Text Editors, Handling Packages in R, Working with Directory.                              | 1                    | -                     | -                      |
| 1.3   | Few Commands for Data Exploration.   | 2                    | -                     | -                      |
| <b>Module – 2: Loading and Handling Data in R</b> |  |                      |                       |                        |
| 2.1   | Introduction, Challenges of Analytical Data Processing, Expression, Variables and Functions.         | 2                    | -                     | -                      |
| 2.2   | Missing Values Treatment in R, Using the <code>_as_</code> Operator to Change the Structure of Data. | 2                    | -                     | -                      |
| 2.3   | Vectors, Few Common Analytical Tasks, Methods for Reading Data.                                      | 1                    | -                     | -                      |
| <b>Module-3:Exploring Data in R</b>               |  |                      |                       |                        |
| 3.1   | Introduction, Data Frames, R Functions for Understanding Data in Data Frames.                        | 2                    | -                     | -                      |
| 3.2   | Load Data Frames, Exploring Data.  | 2                    | -                     | -                      |
| 3.3   | Descriptive Statistics, Spotting Problems in Data with Visualization.                                | 2                    | -                     | -                      |
| <b>Total No. of Lecture Hours</b>                 |  | <b>15</b>            | -                     | -                      |
| <b>Total No. of Tutorial Hours</b>                |  |                      | <b>00</b>             | -                      |
| <b>Total No. of Practical Hours</b>               |  |                      |                       | <b>00</b>              |

**Textbook:**

1. Data Analytics Using R by Seema Acharya, MC GRAW HILL INDIA, © Published: April 20, 2018.

**Reference Book:**

1. Jared P Lander, R for everyone: advanced analytics and graphics, Pearson Education, 2013

**Online Resources:**

1. <https://dokumen.pub/data-analytics-using-r-paperback-jan-01-2018-seema-acharya-9352605241-9789352605248.html>

**Code: BCS358C****Course: Project Management with Git****Credits: 1****CIE: 50 Marks****L:T:P -1:0:0****SEE: NA****SEE Hours: NA****Total Marks:50**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | NIL   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To make the students understand the basic concepts related to Version controller with Git.</li> <li>To provide knowledge of Installation procedure and basic commands in Git.</li> </ol> |

**Course Outcomes:**

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

| COs | Course Outcomes   | Bloom's level |
|-----|---|---------------|
| CO1 | Describe basic concepts related to Version controller with GiT. | Understanding |
| CO2 | Explain Installation procedure and basic commands in GiT.       | Understanding |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 2   | -   | -   | 2   | -   | -   | -   | -   | -    | 2    | 2    | -    | -    |
| CO2 | 3   | 2   | -   | -   | 2   | -   | -   | -   | -   | -    | 2    | 2    | -    | -    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

**Course Structure**

| Sl. No.           | Module   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|-------------------|--|----------------------|-----------------------|------------------------|
| <b>Module - 1</b> |  |                      |                       |                        |
| 1.1               | Setting Up and Basic Commands<br>Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message. | 1                    | -                     | -                      |
| 1.2               | Creating and Managing Branches<br>Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."                                       | 1                    | -                     | -                      |
| 1.3               | Creating and Managing Branches<br>Write the commands to stash your changes, switch branches, and then apply the stashed changes.   | 1                    | -                     | -                      |
| 1.4               | Collaboration and Remote Repositories<br>Clone a remote Git repository to your local machine.  | 1                    | -                     | -                      |
| <b>Module 2</b>   |  |                      |                       |                        |
| 2.1               | Collaboration and Remote Repositories<br>Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.                                      | 1                    | -                     | -                      |
| 1.6               | Collaboration and Remote Repositories<br>Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge                                     | 2                    | -                     | -                      |
| 2.1               | Git Tags and Releases<br>Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.   | 1                    | -                     | -                      |
| 2.2               | Advanced Git Operations<br>Write the command to cherry-pick a range of commits from "source-branch" to the current branch.   | 1                    | -                     | -                      |
| <b>Module 3</b>   |  |                      |                       |                        |

|                                     |   |           |           |           |
|-------------------------------------|---|-----------|-----------|-----------|
| 2.3                                 | Analysing and Changing Git History<br>Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message? | 1         | -         | -         |
| 2.4                                 | Analysing and Changing Git History<br>Write the command to list all commits made by the author "JohnDoe" between "2023-01-01" and "2023-12-31."                             | 2         | -         | -         |
| 2.5                                 | Analysing and Changing Git History<br>Write the command to display the last five commits in the repository's history.   | 2         |           |           |
| 2.6                                 | Analysing and Changing Git History<br>Write the command to undo the changes introduced by the commit with the ID "abc123".  | 1         | -         | -         |
|                                     |   |           | -         | -         |
| <b>Total No. of Lecture Hours</b>   |   | <b>15</b> | -         | -         |
| <b>Total No. of Tutorial Hours</b>  |   |           | <b>00</b> | -         |
| <b>Total No. of Practical Hours</b> |   |           |           | <b>00</b> |

**Textbook:**

1. Git: Version Control for Everyone Beginner's Guide Copyright © 2013 Packt, Publishing Ravishankar Somasundaram.

**Reference Book:**

1. Version Control with Git by Jon Loeliger, Copyright © 2009 Jon Loeliger. All rights reserved. Printed in the United States of America. ISBN: 978-0-596-52012-0 [M]1242320486

**Online Resources:** <https://www.udemy.com/course/introduction-to-version-control-with-git-and-github/>

**Code: BCS358D****Credits: 1****L:T:P - 0:0:2****SEE Hours: NA****Course: Data Visualization with Python****CIE: 50 Marks****SEE: NA****Total Marks:50**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | Introduction to python programming  |
| <b>Learning objectives</b>  | To understand the basics of data visualizations techniques<br>To understand the different plotting libraries and apply in real world scenarios. |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Understand the basics of data visualization techniques and libraries | Understanding |
| CO2 | Apply the data visualization techniques in the real world problems   | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 3   | -   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |
| CO2 | 3   | 3   | -   | -   | 2   | -   | -   | -   | -   | -    | -    | 2    | 2    | 2    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

| Sl.No  | Experiments  | No. of Practical Hours |
|--|--|------------------------|
| <b>List of problems for which student should develop program and execute in the Laboratory</b> |  |                        |
| 1  | a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user.<br>b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.<br>Datatypes: <a href="https://www.youtube.com/watch?v=gCCVsvgR2KU">https://www.youtube.com/watch?v=gCCVsvgR2KU</a> Operators:<br><a href="https://www.youtube.com/watch?v=v5MR5JnKcZI">https://www.youtube.com/watch?v=v5MR5JnKcZI</a> Flow Control:<br><a href="https://www.youtube.com/watch?v=PqFKRqpHrjw">https://www.youtube.com/watch?v=PqFKRqpHrjw</a> For loop:<br><a href="https://www.youtube.com/watch?v=0ZvaDa8eT5s">https://www.youtube.com/watch?v=0ZvaDa8eT5s</a> While loop:<br><a href="https://www.youtube.com/watch?v=HZARImviDxg">https://www.youtube.com/watch?v=HZARImviDxg</a> Exceptions:<br><a href="https://www.youtube.com/watch?v=6SPDvPK38tw">https://www.youtube.com/watch?v=6SPDvPK38tw</a> | 2 Hour                 |
| 2  | a) Defined as a function F as $F_n = F_{n-1} + F_{n-2}$ . Write a Python program which accepts a value for N (where $N > 0$ ) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.<br>b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.<br>c) Write a Python programs for Cleaning Data, Checking for missing values and Handling the missing values.<br>Functions: <a href="https://www.youtube.com/watchca9nw">https://www.youtube.com/watchca9nw</a><br>Arguments: <a href="https://www.youtube.com/watch?v=ijXMGpoMkhQ">https://www.youtube.com/watch?v=ijXMGpoMkhQ</a> Return value:<br><a href="https://www.youtube.com/watch?v=nuNXiEDn44">https://www.youtube.com/watch?v=nuNXiEDn44</a><br>Data_Analysis_and_Visualization_Using_Python<br>by Dr. Ossama Embarak 2018   | 2 Hours                |

|                                     |  |           |
|-------------------------------------|--|-----------|
| 3                                   | <p>a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.</p> <p>b) Write a Python program to find the string similarity between two given strings</p> <p>c) Write a Python program for Reading and Cleaning CSV data.</p> <p><b>Sample Output:</b><br/>Original string:<br/>Python Exercises<br/>Python Exercises<br/>Similarity between two said strings:</p> <p><b>Sample Output:</b><br/>Original string:<br/>Python Exercises<br/>Python Exercise<br/>Similarity between two said strings:1.0<br/>0.967741935483871</p> <p>Strings: <a href="https://www.youtube.com/watch?v=ISItwlnF0eU">https://www.youtube.com/watch?v=ISItwlnF0eU</a><br/>String functions: <a href="https://www.youtube.com/watch?v=9a3CxJyTq00">https://www.youtube.com/watch?v=9a3CxJyTq00</a><br/>Data_Analysis_and_Visualization_Using_Python by Dr. Ossama Embarak 2018</p> | 2 Hours   |
| 4                                   | <p>a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.</p> <p>b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.</p> <p><a href="https://www.youtube.com/watch?v=RRHQ6Fs1b8w&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=3">https://www.youtube.com/watch?v=RRHQ6Fs1b8w&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=3</a><br/><a href="https://www.youtube.com/watch?v=7ABCuhWO9II&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=4">https://www.youtube.com/watch?v=7ABCuhWO9II&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=4</a></p>   | 1 Hour    |
| 5                                   | <p>a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.</p> <p>b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.</p> <p><a href="https://www.youtube.com/watch?v=Qk7caotaQUQ&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=6">https://www.youtube.com/watch?v=Qk7caotaQUQ&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=6</a><br/><a href="https://www.youtube.com/watch?v=PSji21jUNO0&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=7">https://www.youtube.com/watch?v=PSji21jUNO0&amp;list=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&amp;index=7</a></p>  | 1 Hour    |
| 6                                   | <p>a) Write a Python program to illustrate Linear Plotting using Matplotlib.</p> <p>b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib.</p> <p><a href="https://www.youtube.com/watch?v=UO98IJQ3QGI&amp;list=PL-osiE80TeTvipOqomVEeZ1HRrcEvtZB">https://www.youtube.com/watch?v=UO98IJQ3QGI&amp;list=PL-osiE80TeTvipOqomVEeZ1HRrcEvtZB</a></p>   | 1 Hour    |
| 7                                   | Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions. <a href="https://www.youtube.com/watch?v=6GUZXDef2U0">https://www.youtube.com/watch?v=6GUZXDef2U0</a>  | 1 Hours   |
| 8                                   | <p>a) Write a Python program to explain working with bokeh line graph using Annotations and Legends.</p> <p>b) Write a Python program for plotting different types of plots using Bokeh.</p> <p><a href="https://www.youtube.com/watch?v=HDvxYoRadcA">https://www.youtube.com/watch?v=HDvxYoRadcA</a></p>  | 2 Hours   |
| 9                                   | Write a Python program to draw 3D Plots using Plotly Libraries.<br><a href="https://www.youtube.com/watch?v=cCck7hCanpw&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=4">https://www.youtube.com/watch?v=cCck7hCanpw&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=4</a>   | 1 hour    |
| 10                                  | <p>a) Write a Python program to draw Time Series using Plotly Libraries.</p> <p>b) Write a Python program for creating Maps using Plotly Libraries.</p> <p><a href="https://www.youtube.com/watch?v=xnJ2TNRGYik&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=5">https://www.youtube.com/watch?v=xnJ2TNRGYik&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=5</a><br/><a href="https://www.youtube.com/watch?v=D35m2CdMhVs&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=6">https://www.youtube.com/watch?v=D35m2CdMhVs&amp;list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&amp;index=6</a></p>   | 2 Hours   |
| <b>Total No. of Lecture Hours</b>   |  | <b>00</b> |
| <b>Total No. of Practical Hours</b> |  | <b>15</b> |

**Textbook:**

1. Data Analysis and Visualization Using Python, Dr. Ossama Embarak, 2018
2. Python for Data Analysis by Wes McKinney, 2nd Edition, October 2017, O'Reilly Media Publisher

**Online resources**

1. <https://doi.org/10.1007/978-1-4842-4109-7>
2. <https://www.oreilly.com/library/view/python-for-data/9781491957653/>

| <b>The National Institute of Engineering, Mysuru</b> |                |             |  |                          |                                    |                               |   |   |   |             |            |            |            |             |
|--|----------------|-------------|--|--------------------------|------------------------------------|-------------------------------|---|---|---|-------------|------------|------------|------------|-------------|
| <b>Scheme of Teaching &amp; Examination</b>          |                |             |  |                          |                                    |                               |   |   |   |             |            |            |            |             |
| <b>Department: Computer Science and Engineering</b>  |                |             |  |                          |                                    |                               |   |   |   |             |            |            |            |             |
| <b>B.E. 2023 Admitted Batch</b>                      |                |             |  |                          |                                    |                               |   |   |   |             |            |            |            |             |
| <b>Semester: IV</b>                                  |                |             |  |                          |                                    |                               |   |   |   |             |            |            |            |             |
| Sl. No.  | Type of Course | Course Code | Course Title                                   | Teaching Department (TD) | Question Paper setting Board (PSB) | Teaching Hrs/Week             |   |   |   | Examination |            |            | Credits    |             |
|  |                |             |  |                          |                                    | L                             | T | P | S | Duration in | CIE Marks  | SEE Marks  |            | Total Marks |
| 1  | PCC/BSC        | BCS401      | Analysis & Design of Algorithms                | TD: CS                   | PSB: CS                            | 3                             | 0 | 0 | 0 | 3           | 50         | 50         | 100        | 3           |
| 2  | IPCC           | BCS402      | Microcontrollers                               | TD: CS                   | PSB: CS                            | 3                             | 0 | 2 | 0 | 3           | 50         | 50         | 100        | 4           |
| 3  | IPCC           | BCS403      | Database Management Systems                    | TD: CS                   | PSB: CS                            | 3                             | 0 | 2 | 0 | 3           | 50         | 50         | 100        | 4           |
| 4  | PCCL           | BCSL404     | Analysis & Design of Algorithms Lab            | TD: CS                   | PSB: CS                            | 0                             | 0 | 2 | 0 | 2           | 50         | 50         | 100        | 1           |
| 5  | ESC            | BCS405x     | ESC/ETC/PLC                                    | TD: CS                   | PSB: CS                            | 2                             | 2 | 0 | 0 | 3           | 50         | 50         | 100        | 3           |
| 6  | AEC/SEC        | BCS456x     | Ability Enhancement Course (AEC) - IV          | TD: CS                   | PSB: CS                            | If the course is a Theory     |   |   |   |             | 50         | -          | 50         | 1           |
|  |                |             |  |                          |                                    | 1                             | 0 | 0 | 0 | 1           |            |            |            |             |
|  |                |             |  |                          |                                    | If the course is a Laboratory |   |   |   |             |            |            |            |             |
|  |                |             |  |                          |                                    | 0                             | 0 | 2 | 0 | 2           |            |            |            |             |
| 7  | BSC            | BBOK407     | Biology for Engineers                          | TD: Chem                 | PSB: Chem                          | 2                             | 0 | 0 | 0 | 2           | 50         | 50         | 100        | 2           |
| 8  | UHV            | BUHK408     | Universal Human Values and Professional Ethics | TD: CS                   | PSB: CS                            | 1                             | 0 | 0 | 0 | -           | 50         | 50         | 100        | 1           |
| 9  | MC             | BNSK459     | National Service Scheme (NSS)                  | NSS Coordinator          |                                    | 0                             | 0 | 2 | 0 | -           | 100        | -          | 100        | 0           |
|  |                | BPEK459     | Physical Education (PE) (Sports & Athletics)   | PED                      |                                    |                               |   |   |   |             |            |            |            |             |
|  |                | BYOK459     | Yoga   | Yoga Teacher             |                                    |                               |   |   |   |             |            |            |            |             |
| <b>Total</b>   |                |             |  |                          |                                    |                               |   |   |   |             | <b>550</b> | <b>300</b> | <b>850</b> | <b>19</b>   |

PCC: Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, MC: Mandatory Course (Non-credit), AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, L: Lecture, T: Tutorial, P: Practical: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation. K :This letter in the course code indicates common to all the stream of engineering. ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course

| <b>Engineering Science Course (ESC/ETC/PLC)</b> |   |         |                                     |
|---|---|---------|-------------------------------------|
| BCS405A   | Discrete Mathematical Structures                        | BCS405C | Optimization Technique              |
| BCS405B   | Graph Theory  | BCS405D | Linear Algebra                      |
| <b>Ability Enhancement Course - IV</b>          |   |         |                                     |
| BCS456A   | Green IT and Sustainability                             | BCS456C | UI/UX (Lab)                         |
| BCS456B   | Capacity Planning for IT Values and Professional Ethics | BCS456D | Technical writing using LATEX (Lab) |
| BCS456E   | UNIX and Shell Programming                              |         |                                     |

**Code: BCS401****Course: Analysis and Design of Algorithms****Credits: 3****L:T:P - 3:0:0****SEE: 50 Marks****CIE: 50 Marks****SEE Hours: 03****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Recurrence Relations, Data Structures  |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>To learn the methods for analyzing algorithms and evaluating their performance.</li> <li>To demonstrate the efficiency of algorithms using asymptotic notations.</li> <li>To solve problems using various algorithm design methods, including brute force, greedy, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking, and branch and bound.</li> <li>To learn the concepts of P and NP complexity classes</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes   | Bloom's level |
|-----|---|---------------|
| CO1 | Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.                                | Analyze       |
| CO2 | Demonstrate brute force approach, divide & conquer approaches and decrease & conquer approaches to solve computational problems.            | Apply         |
| CO3 | Make use of transform & conquer, dynamic programming and greedy approaches to solve the given real world or complex computational problems. | Apply         |
| CO4 | Illustrate backtracking, branch & bound and P, NP and NP Complete problems  | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | 2   | 1   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO2 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO3 | 2   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO4 | 2   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

**Course Structure**

|                   |  | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|-------------------|--|----------------------|-----------------------|------------------------|
| <b>Module - 1</b> |  |                      |                       |                        |
| 1                 | <b>INTRODUCTION:</b>   |                      |                       |                        |
| 1.1               | What is an Algorithm?  | 1                    | -                     | -                      |
| 1.2               | Fundamentals of Algorithmic Problem Solving.                 | 2                    | -                     | -                      |
|                   | <b>FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY:</b> |                      |                       |                        |
| 1.3               | Analysis Framework   | 1                    | -                     | -                      |
| 1.4               | Asymptotic Notations and Basic Efficiency Classes            | 1                    | -                     | -                      |

|                    |  |                                     |           |           |
|--------------------|--|-------------------------------------|-----------|-----------|
| 1.5                | Mathematical Analysis of Non recursive Algorithms    | 1                                   | -         | -         |
| 1.6                | Mathematical Analysis of Recursive Algorithms.       | 1                                   | -         | -         |
|                    | <b>BRUTE FORCE APPROACHES:</b>                       |                                     | -         | -         |
| 1.7                | Selection Sort                                       | 1                                   | -         | -         |
| 1.8                | Sequential Search                                    | 1                                   | -         | -         |
| <b>Module – 2</b>  |  |                                     |           |           |
|                    | <b>BRUTE FORCE APPROACHES (contd.):</b>              |                                     |           |           |
| 2.1                | Brute Force String Matching                          | 1                                   | -         | -         |
| 2.2                | Exhaustive Search (Travelling Salesman problem)      | 1                                   | -         | -         |
| 2.3                | Exhaustive Search (Knapsack Problem).                | 1                                   | -         | -         |
|                    | <b>DECREASE-AND-CONQUER:</b>                         |                                     | -         | -         |
| 2.4                | Topological Sorting.                                 | 1                                   | -         | -         |
|                    | <b>DIVIDE AND CONQUER:</b>                           |                                     | -         | -         |
| 2.5                | Merge Sort   | 1                                   | -         | -         |
| 2.6                | Quick Sort   | 1                                   | -         | -         |
| 2.7                | Strassen’s Matrix Multiplication.                    | 2                                   | -         | -         |
| <b>Module – 3</b>  |  |                                     |           |           |
|                    | <b>TRANSFORM-AND-CONQUER:</b>                        |                                     |           |           |
| 3.1                | Balanced Search Trees                                | 2                                   | -         | -         |
| 3.2                | Heaps and Heapsort.                                  | 3                                   | -         | -         |
|                    | <b>SPACE-TIME TRADEOFFS:</b>                         |                                     | -         | -         |
| 3.3                | Sorting by Counting: Comparison counting sort        | 3                                   | -         | -         |
| <b>Module – 4:</b> |  |                                     |           |           |
|                    | <b>DYNAMIC PROGRAMMING:</b>                          |                                     |           |           |
| 4.1                | The Knapsack Problem and Memory Functions            | 2                                   | -         | -         |
| 4.2                | Warshall’s and Floyd’s Algorithms.                   | 2                                   | -         | -         |
|                    | <b>THE GREEDY METHOD:</b>                            |                                     | -         | -         |
| 4.3                | Prim’s Algorithm                                     | 2                                   | -         | -         |
| 4.4                | Kruskal’s Algorithm                                  | 1                                   | -         | -         |
| 4.5                | Dijkstra’s Algorithm                                 | 1                                   | -         | -         |
| <b>Module – 5</b>  |  |                                     |           |           |
|                    | <b>LIMITATIONS OF ALGORITHMIC POWER:</b>             |                                     |           |           |
| 5.1                | Decision Trees                                       | 2                                   | -         | -         |
|                    | <b>COPING WITH LIMITATIONS OF ALGORITHMIC POWER:</b> |                                     | -         | -         |
| 5.2                | Backtracking (n-Queens problem)                      | 2                                   | -         | -         |
| 5.3                | Backtracking (Subset-sum problem)                    | 2                                   | -         | -         |
| 5.4                | Branch-and-Bound (Travelling Salesman Problem),      | 1                                   | -         | -         |
|                    |  | <b>Total No. of Lecture Hours</b>   | <b>40</b> | -         |
|                    |  | <b>Total No. of Tutorial Hours</b>  | <b>00</b> | -         |
|                    |  | <b>Total No. of Practical Hours</b> | <b>00</b> | <b>00</b> |

**Textbook**

Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian),

**Reference books**

1. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

**Online Resources:**

- Design and Analysis of Algorithms: <https://nptel.ac.in/courses/106/101/106101060/>

**Code: BCS402**  
**Credits: 4**  
**SEE: 50 Marks**  
**SEE Hours: 3**

**Course: Microcontrollers**  
**L:T:P - 3:0:2**  
**CIE: 50 Marks**  
**Max. Marks:100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Digital Design and Computer Organization   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To provide the knowledge of the fundamentals of ARM-based systems and basic architecture of CISC and RISC</li> <li>To familiarize with ARM programming modules along with registers, CPSR and Flags</li> <li>To develop ALP/C Program using various instructions to program the ARM controller</li> <li>To discuss and demonstrate the Exceptions and Interrupt handling mechanism, ARM Firmware packages and Cache memory polices</li> </ol> |

#### Course Outcomes:

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Understand the fundamentals of ARM-based systems and basic architecture of CISC and RISC     | Understanding |
| CO2 | Familiarize with ARM programming modules along with registers, CPSR and Flags                | Apply         |
| CO3 | Develop Assembly Language/C Program using various instructions to program the ARM controller | Apply         |
| CO4 | Demonstrate the Exceptions and Interrupt handling mechanism in Microcontrollers              | Apply         |
| CO5 | Discuss the ARM Firmware packages and Analyze Cache memory polices                           | Analyze       |

#### Mapping with POs and PSOs:

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PS01 | PS02 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | -   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO2 | 3   | -   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO3 | 3   | 3   | 3   | -   | 3   | 2   | 2   | -   | -   | -    | -    | 3    | 3    | -    |
| CO4 | 3   | 2   | 2   | 2   | 2   | -   | -   | -   | -   | -    | -    | 3    | 2    | -    |
| CO5 | 3   | 3   | -   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

#### Course Structure

|  |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|--|---|----------------------|-----------------------|------------------------|
| <b>Module – 1: ARM EMBEDDED SYSTEMS &amp; PROCESSOR FUNDAMENTALS</b> |   |                      |                       |                        |
| 1.1  | ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy | 1                    | -                     | -                      |
| 1.2  | Embedded System Hardware, Embedded System Software                          | 2                    | -                     | -                      |
| 1.3  | ARM Processor Fundamentals: Registers, Current Program Status Register      | 2                    | -                     | 1                      |
| 1.4  | Pipeline  | 1                    | -                     | -                      |
| 1.5  | Exceptions Interrupts, and the Vector Table                                 | 2                    | -                     | -                      |
| <b>Textbook 1: Chapter 1 - 1.1 to 1.4, Chapter 2 - 2.1 to 2.4</b>    |   |                      |                       |                        |
| <b>Module – 2: ARM INSTRUCTION SET</b>                               |   |                      |                       |                        |
| 2.1  | Introduction to the ARM Instruction Set: Data Processing Instructions       | 3                    | -                     | 3                      |
| 2.2  | Branch Instructions, Single Register Transfer, Multiple Register Transfer   | 2                    | -                     | 2                      |
| 2.3  | Software Interrupt Instructions, Program Status Register Instruction        | 2                    | -                     | -                      |
| 2.4  | Loading Constants   | 1                    | -                     | -                      |
| <b>Textbook 1: Chapter 3 - 3.1 to 3.2,3.3.1,3.3.3,3.4,3.5,3.6</b>    |   |                      |                       |                        |

| <b>Module – 3: C COMPILERS AND OPTIMIZATION</b>               |   |           |   |           |
|---|---|-----------|---|-----------|
| 3.1   | C Compilers and Optimization: Basic C Data Types  | 2         | - | -         |
| 3.2   | C Looping Structures  | 2         | - | 1         |
| 3.3   | Register Allocation, Function Calls   | 2         | - | 1         |
| 3.4   | Pointer Aliasing  | 2         | - | -         |
|   | <b>Textbook 1: Chapter 5.1 to 5.6</b>   |           |   |           |
| <b>Module – 4: EXCEPTION, INTERRUPT HANDLING AND FIRMWARE</b> |   |           |   |           |
| 4.1   | Exception and Interrupt Handling: Exception handling  | 1         | - | 2         |
| 4.2   | ARM processor exceptions and modes, vector table, exception priorities                          | 2         | - | -         |
| 4.3   | link register offsets, interrupts, assigning interrupts   | 1         | - | -         |
| 4.4   | interrupt latency, IRQ and FIQ exceptions   | 1         | - | -         |
| 4.5   | Firmware: Firmware and bootloader, ARM firmware suite, Red Hat redboot                          | 3         | - | -         |
|   | <b>Textbook 1: Chapter 9.1 and 9.2, Chapter 10.1</b>  |           |   |           |
| <b>Module – 5: CACHE MEMORY</b>                               |   |           |   |           |
| 5.1   | CACHES: The Memory Hierarchy and Cache Memory   | 2         | - | -         |
| 5.2   | Caches and Memory Management Units, CACHE Architecture: Basic Architecture of a Cache Memory    | 2         | - | -         |
| 5.3   | Basic Operation of a Cache Controller, The Relationship between Cache and Main Memory           | 2         | - | -         |
| 5.4   | Write Buffers, Measuring Cache Efficiency, CACHE POLICY: Write Policy—Writeback or Writethrough | 2         | - | -         |
|   | <b>Textbook 1: Chapter 12.1 to 12.3</b>   |           |   |           |
|   |   |           | - | -         |
| <b>Total No. of Lecture Hours</b>                             |   | <b>40</b> | - | -         |
| <b>Total No. of Tutorial Hours</b>                            |   | <b>00</b> | - | -         |
| <b>Total No. of Practical Hours</b>                           |   |           |   | <b>10</b> |

| <b>Integrated Lab Component: Microcontrollers</b> |             |   |
|---|-------------|---|
| <b>Sl No</b>                                      | <b>CO's</b> | <b>Experiments</b>  |
| 1   | CO2         | Using Keil software, observe the various Registers, Dump, CPSR, with a simple Assembly Language Programs (ALP)                      |
| 2   | CO3         | Develop and simulate ARM ALP for Data Transfer, Arithmetic and Logical operations (Demonstrate with the help of a suitable program) |
| 3   | CO3         | Develop an ALP to multiply two 16-bit binary numbers  |
| 4   | CO3         | Develop an ALP to find the sum of first 10 integer numbers  |
| 5   | CO3         | Develop an ALP to find the largest/smallest number in an array of 32 numbers  |
| 6   | CO3         | Develop an ALP to count the number of ones and zeros in two consecutive memory locations  |
| 7   | CO3         | Simulate a program in C for ARM microcontroller using KEIL to sort the numbers in ascending/descending using bubble sort            |
| 8   | CO3         | Simulate a program in C for ARM microcontroller to find factorial of a number   |
| 9   | CO4         | Demonstrate enabling and disabling of Interrupts in ARM   |
| 10  | CO4         | Demonstrate the handling of divide by zero, Invalid Operation and Overflow exceptions in ARM  |

**Textbook:**

1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan, Kaufman publisher,2008

**Reference Book:**

2. Raghunandan.G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication, 2019.,
3. Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005

**Online Resources:**

1. Online Resource 1: [Top Microcontroller Courses Online - Updated \[April 2024\] \(udemy.com\)](#)
2. Online Resource 2: [Microprocessors And Microcontrollers - Course \(nptel.ac.in\)](#)

**Code: BCS403****Course: Database Management Systems****Credits: 4****L:T:P - 3:0:2****SEE: 50 Marks****CIE: 50 Marks****SEE Hours: 3****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | NIL  |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>Understand the fundamental concepts of databases, including database languages, architectures, and conceptual data modeling using entities and relationships.</li> <li>Gain proficiency in relational database management systems, including the relational model, relational algebra, normalization, SQL, and transaction processing.</li> </ol> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Describe fundamental concepts of database management systems, including architecture, languages, and functionalities.                                      | Understand    |
| CO2 | Design and implement database schemas using entities, relationships, and normalization techniques.   | Apply         |
| CO3 | Demonstrate proficiency in SQL for data manipulation, retrieval, and management tasks.   | Apply         |
| CO4 | Analyze and compare concurrency control mechanisms in relational databases and NoSQL databases, understanding their respective advantages and limitations. | Analyze       |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | -   | -   | -   | 3   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO2 | 3   | 3   | 2   | -   | 3   | -   | -   | -   | -   | -    | -    | 3    | 3    | 2    |
| CO3 | 2   | 2   | 2   | -   | 3   | -   | -   | -   | -   | -    | -    | 3    | 3    | 3    |
| CO4 | 2   | -   | -   | -   | 3   | -   | -   | -   | -   | -    | -    | 3    | 3    | 2    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

**Course Structure**

|  |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|--|---|----------------------|-----------------------|------------------------|
| <b>Module – 1: Introduction to Databases</b> |   |                      |                       |                        |
| 1.1  | <b>Introduction to Databases:</b> Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.  | 2                    | -                     | -                      |
| 1.2  | <b>Overview of Database Languages and Architectures:</b> Data Models, Schemas, and Instances. Three schema architecture and data independence. Database languages, and interfaces, The Database System environment. | 3                    | -                     | -                      |
| 1.3  | <b>Conceptual Data Modelling using Entities and Relationships:</b> Entity types, Entity sets and structural constraints, Weak entity types, ER diagrams, Specialization and Generalization                          | 3                    | -                     | 1                      |
| <b>Module – 2: Relational Databases</b>      |   |                      |                       |                        |
| 2.1  | <b>Relational Model:</b> Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.                          | 3                    | -                     | -                      |
| 2.2  | <b>Relational Algebra:</b> Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra.  | 3                    | -                     | -                      |
| 2.3  | <b>Mapping Conceptual Design into a Logical Design:</b> Relational Database Design using  | 2                    | -                     | -                      |

|  |   |           |           |           |
|--|---|-----------|-----------|-----------|
|  | ER-to-Relational mapping  |           |           |           |
| <b>Module – 3: Normalization and SQL</b>                   |   |           |           |           |
| 3.1  | <b>Normalization:</b> Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form. | 5         | -         | 1         |
| 3.2  | <b>SQL:</b> SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL.   | 3         | -         | 2         |
| <b>Module – 4: SQL and Transactions</b>                    |   |           |           |           |
| 4.1  | <b>SQL:</b> Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.   | 3         | -         | 2         |
| 4.2  | <b>Transaction Processing:</b> Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL.  | 5         | -         | 1         |
| <b>Module – 5: Concurrency control and NoSQL Databases</b> |   |           |           |           |
| 5.1  | <b>Concurrency Control in Databases:</b> Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.   | 4         | -         | 1         |
| 5.2  | <b>NoSQL Databases and Big Data Storage Systems:</b> Introduction to NOSQL Systems, The CAP Theorem, Document-Based NOSQL Systems and MongoDB, NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems.   | 4         | -         | 2         |
| <b>Total No. of Lecture Hours</b>                          |   | <b>40</b> | -         | -         |
| <b>Total No. of Tutorial Hours</b>                         |   |           | <b>00</b> | -         |
| <b>Total No. of Practical Hours</b>                        |   |           |           | <b>10</b> |

| Sl. No. | Experiments  | COs |
|---------|--|-----|
| 1.      | <p>Create a table called Employee &amp; execute the following.<br/>Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION)</p> <ol style="list-style-type: none"> <li>Create a user and grant all permissions to the user.</li> <li>Insert any three records in the employee table containing attributes. EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result.</li> <li>Add primary key constraint and not null constraint to the employee table.</li> <li>Insert NULL values to the employee table and verify the result.</li> </ol> | CO3 |
| 2.      | <p>Create a table called Employee that contains attributes EMPNO, ENAME, JOB, MGR, SAL and execute the following.</p> <ol style="list-style-type: none"> <li>Add a column commission with domain to the Employee table.</li> <li>Insert any five records into the table.</li> <li>Update the column details of job.</li> <li>Rename the column of Employ table using alter command.</li> <li>Delete the employee whose EMPNO is 105.</li> </ol>  | CO3 |
| 3.      | <p>Queries using aggregate functions (COUNT, AVG, MIN, MAX, SUM), Group by, Orderby.<br/>Employee (E_id, E_name, Age, Salary)</p> <ol style="list-style-type: none"> <li>Create Employee table containing all Records E_id, E_name, Age, Salary.</li> <li>Count number of employee names from employee table.</li> <li>Find the Maximum age from the employee table.</li> <li>Find the Minimum age from the employee table.</li> <li>Find salaries of employees in Ascending Order.</li> <li>Find grouped salaries of employees.</li> </ol>                          | CO3 |

|    |  |                              |
|----|--|------------------------------|
| 4. | Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary.<br>CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) | CO4                          |
| 5. | Create cursor for Employee table and extract the values from the table. Declare the variables, Open the cursor, and extract the values from the cursor. Close the cursor.<br>Employee (E_id, E_name, Age, Salary)  | CO4                          |
| 6. | Install an Open-Source NoSQL Data base MongoDB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MongoDB basic Queries using CRUD operations.   | CO5                          |
| 7. | Project-based Experiment:<br>The project should use all the database concepts covered in theory and laboratory sessions. Students can integrate other relevant concepts/technologies as required.  | CO1,<br>CO2,<br>CO3,<br>CO4. |

**Textbook:**

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

**Reference Book:**

1. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

**Online Resources:**

1. MIT OpenCourseWare Course Link: <https://ocw.mit.edu/courses/6-830-database-systems-fall-2010/>
2. IIT Kharagpur Course Link: [https://cse.iitkgp.ac.in/~pabitra/course/dbms/dbms\\_new.html](https://cse.iitkgp.ac.in/~pabitra/course/dbms/dbms_new.html)
3. NPTEL Course Link: [https://onlinecourses.nptel.ac.in/noc22\\_cs91/preview](https://onlinecourses.nptel.ac.in/noc22_cs91/preview)

|  |           |            |    |
|--|-----------|------------|----|
| <b>Analysis &amp; Design of Algorithms Lab</b> |           | Semester   | 4  |
| Course Code                                    |           | CIE Marks  | 50 |
| Teaching Hours/Week (L:T:P)                    | 0:0:2     | SEE Marks  | 50 |
| Credits  | 01        | Exam Hours | 2  |
| Examination type (SEE)                         | Practical |            |    |

**Course outcomes:**

At the end of the course the student will be able to:

1. Develop programs to solve computational problems using suitable algorithm design strategy.
2. Compare algorithm design strategies by developing equivalent programs and observing runningtimes for analysis (Empirical).
3. Make use of suitable integrated development tools to develop programs
4. Choose appropriate algorithm design techniques to develop solution to the computational and complex problems.
5. Demonstrate and present the development of program, its execution and running time(s) and record the results/inferences.

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | 2   | 1   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO2 | 3   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO3 | 2   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO4 | 2   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |
| CO5 | 2   | 2   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 1    | 3    | 2    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

| SL.N<br>o | CO's       | Experiments   |
|-----------|------------|---|
| 1         | CO1 to CO5 | Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n > 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. |
| 2         | CO1 to CO5 | Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph.   |
| 3         | CO1 to CO5 | Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n > 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.    |

|   |                  |   |
|---|------------------|---|
| 4 | CO1<br>to<br>CO5 | Design and implement C/C++ Program to sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of $n > 5000$ and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. |
| 5 | CO1<br>to<br>CO5 | Design and implement C/C++ Program to solve 0/1 Knapsack problem using Dynamic programming method.  |
| 6 | CO1<br>to<br>CO5 | a. Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm.<br>b. Design and implement C/C++ Program to find the transitive closure using Warshal's algorithm.   |

|    |                  |   |
|----|------------------|---|
| 7  | CO1<br>to<br>CO5 | Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.                       |
| 8  | CO1<br>to<br>CO5 | Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.                    |
| 9  | CO1<br>to<br>CO5 | Design and implement C/C++ Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm. |
| 10 | CO1<br>to<br>CO5 | Design and implement C/C++ Program for N Queen's problem using Backtracking.  |

**Textbook**

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.
2. Virtual Labs (CSE): <http://cse01-iiith.vlabs.ac.in/>

**Code: BCS405A****Course: Discrete Mathematical Structures****Credits: 3****L:T:P - 2:2:0****SEE: 50 Marks****CIE: 50 Marks****SEE Hours: 3****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Basics of number system, Mathematics   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To solve problems using concepts of Functions.</li> <li>Solve problems using Relations and its properties.</li> <li>To introduce Generating Functions and Recurrence Relations</li> <li>To introduce concepts and properties of Graphs</li> <li>To introduce the concepts of Trees and its properties.</li> </ol> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Apply the concepts and properties of Functions and Relations in solving problem.   | Apply         |
| CO2 | Solve problems using Recurrence Relations and Study its applications in computers. | Apply         |
| CO3 | Solve problems using concepts of graphs and analyze its real-world applications.   | Apply         |
| CO4 | Synthesis tree structure paradigm.   | Analyze       |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | 3   | -   | 2   | -   | -   | -   | -   | 2   | -    | 2    | 2    | -    | 2    |
| CO2 | 3   | 3   | -   | 2   | -   | -   | -   | -   | 2   | -    | 2    | 2    | -    | 2    |
| CO3 | 3   | 3   | -   | 2   | -   | -   | -   | -   | 2   | -    | 2    | 2    | -    | 2    |
| CO4 | 3   | 3   | -   | 3   | -   | -   | -   | -   | 2   | -    | 2    | 2    | -    | 2    |

**Mapping Strength: Strong- 3 Medium - 2 Low - 1**

**Course Structure**

|   |  | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|---|--|----------------------|-----------------------|------------------------|
| <b>Module – 1: Functions</b>            |  |                      |                       |                        |
| 1.1                                     | <b>Functions:</b> Cartesian Products and Relations                                 | 2                    | -                     | -                      |
| 1.2                                     | Plain and One-to-One, Onto Functions   | 1                    | 1                     | -                      |
| 1.3                                     | The Pigeonhole Principle   | 1                    | 1                     | -                      |
| 1.4                                     | Function Composition and Inverse Functions   | 1                    | 1                     | -                      |
| <b>Module – 2: Relations</b>            |  |                      |                       |                        |
| 2.1                                     | <b>Relations:</b> Properties of Relations,   | 1                    | 1                     | -                      |
| 2.2                                     | Computer Recognition – Zero-One Matrices and Directed Graphs                       | 2                    | 1                     | -                      |
| 2.3                                     | Partial Orders – Hasse Diagrams.   | 2                    | 1                     | -                      |
| <b>Module – 3: Recurrence Relations</b> |  |                      |                       |                        |
| 3.1                                     | <b>Recurrence Relations:</b> First order linear recurrence relations,              | 2                    | -                     | -                      |
| 3.2                                     | The Second order linear homogeneous recurrence relation with constant coefficients | 1                    | -                     | -                      |

|  |  |           |           |           |
|--|--|-----------|-----------|-----------|
| 3.3  | Non Homogeneous recurrence relation  | 2         | 1         | -         |
| <b>Module – 4: Graph Theory and Applications</b> |  |           |           |           |
| 4.1  | <b>Graph Theory and Applications:</b> Definitions and Examples Sub graphs, Complements | 1         | 1         | -         |
| 4.2  | Graph Isomorphism, Vertex Degree, Euler Trails and Circuits                            | 1         | 0         | -         |
| 4.3  | Planar Graphs  | 1         | 1         | -         |
| 4.4  | Hamilton Paths and Cycles  | 1         | 1         | -         |
| 4.5  | Graph Coloring, and Chromatic Polynomials  | 1         | 1         | -         |
| <b>Module – 5: Trees</b>                         |  |           |           |           |
| 5.1  | <b>Trees:</b> Definitions, Properties, and Examples                                    | 2         | 1         | -         |
| 5.2  | Rooted Trees   | 1         | 1         | -         |
| 5.3  | Trees and Sorting  | 1         | 1         | -         |
| 5.4  | Weighted Trees and Prefix Codes  | 1         | -         | -         |
| <i>Total No. of Lecture Hours</i>                |  | <b>25</b> | -         | -         |
| <i>Total No. of Tutorial Hours</i>               |  |           | <b>15</b> | -         |
| <i>Total No. of Practical Hours</i>              |  |           |           | <b>00</b> |

**Textbook:**

1. Discrete and Combinatorial Mathematics, Ralph P. Grimaldi, 5th Edition, PHI/Pearson Education, 2004.

**Reference Book:**

1. Handbook of discrete and combinatorial mathematics, Kenneth H.Rosen, John G.Michels.
2. Mathematics of Computer Science, Prof. Albert R.Meyer, MIT Open Course Ware.
3. Concrete Mathematics: A foundation for computer science, Ronald L.Graham, Donald Ervin Knuth, Oren Patashnik
4. Graph Theory with Applications to Engineering and Computer Science by
5. NarsinghDeo, Prentice-Hall, 2004

**Code: BCS405B****Credits: 3****SEE: 50 Marks****SEE Hours: 03****Course: GRAPH THEORY****L:T:P - 2:2:0****CIE: 50 Marks****Max. Marks: 100**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | Algorithms and Data Structure   |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>To understand the basic concepts of graphs and their properties, and operations of graphs.</li> <li>To learn Hamiltonian and Euler graphs, trees and matrix representation of the graph.</li> <li>Apply the concepts of a planar graph, matching and colouring in computer science engineering.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes   | Bloom's level |
|-----|---|---------------|
| CO1 | Explain the fundamental concepts of properties and representation of graphs and Solve the problems involving characterization and operations on graphs. | Analyze       |
| CO2 | Apply concepts of trees and graph connectivity to solve real world problems.  | Apply         |
| CO3 | Apply the concepts of planar graph and graph representations to solve the given problem.  | Apply         |
| CO4 | Use the concepts of matching and coloring of graphs to solve the real world problems..  | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | 2   | -   | -   | -   | -   | -   | -   | -   | -    | -    | -    | -    | -    |
| CO2 | 3   | 3   | 3   | -   | 2   | -   | -   | -   | -   | -    | 3    | 3    | 3    | 3    |
| CO3 | 3   | 3   | 3   | -   | 2   | -   | -   | -   | -   | -    | 3    | 3    | 3    | 2    |
| CO4 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | 3    | -    | -    | -    |

**Mapping Strength: Strong– 3 Medium – 2 Low – 1**

**Course Structure**

|                   |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|-------------------|---|----------------------|-----------------------|------------------------|
| <b>Module – 1</b> |   |                      |                       |                        |
| 1.1               | Introduction to Graphs: Introduction- Basic definition – Application of graphs.                               | 1                    | -                     | -                      |
| 1.2               | finite, infinite and bipartite graphs - Incidence and Degree – Isolated vertex, pendant vertex and Null graph | 1                    | 1                     | -                      |
| 1.3               | Paths and circuits – Isomorphism, sub-graphs  | 1                    | 1                     | -                      |
| 1.4               | walks, paths and circuits   | 1                    | -                     | -                      |
| 1.5               | connected graphs, disconnected graphs and components  | 1                    | -                     | -                      |

| <b>Module – 2</b>                   |   |           |           |          |
|-------------------------------------|---|-----------|-----------|----------|
| 2.1                                 | Eulerian and Hamiltonian graphs: Euler graphs, Operations on,   | 1         | 1         | -        |
| 2.2                                 | Hamiltonian paths and circuits,   | 1         | 1         | -        |
| 2.3                                 | Travelling salesman problem   | 1         | 1         | -        |
| 2.4                                 | graphs Directed graphs – types of digraphs  | 1         | 1         | -        |
| 2.5                                 | Digraphs and binary relation.   | 1         | 1         | -        |
| <b>Module – 3</b>                   |   |           |           |          |
| 3.1                                 | <b>Trees</b> – properties, pendant vertex,  | 1         | 1         | -        |
| 3.2                                 | Distance and centres in a tree - Rooted and binary trees, counting trees, spanning trees.                               | 1         | 1         | -        |
| 3.3                                 | <b>Connectivity Graphs:</b> Vertex Connectivity, Edge Connectivity  | 1         | 1         | -        |
| 3.4                                 | Cut set and Cut Vertices, Fundamental circuits.   | 2         | -         | -        |
| <b>Module – 4</b>                   |   |           |           |          |
| 4.1                                 | Planar Graphs: Planar graphs, Kuratowski's theorem (proof not required),  | 2         | 1         | -        |
| 4.2                                 | Different representations of planar graphs, Euler's theorem, Geometric dual.  | 2         | 1         | -        |
| 4.3                                 | Graph Representations: Matrix representation of graphs-Adjacency matrix, Incidence Matrix, Circuit Matrix, Path Matrix. | 1         | 1         | -        |
| <b>Module – 5</b>                   |   |           |           |          |
| 5.1                                 | Graph Colouring: Colouring- Chromatic number,   | 2         | -         | -        |
| 5.2                                 | Chromatic polynomial, Matchings, Coverings,   | 1         | -         | -        |
| 5.3                                 | Four colour problem and Five colour problem.  | 1         | 1         | -        |
| 5.4                                 | Greedy colouring algorithm.   | 1         | 1         | -        |
| <b>Total No. of Lecture Hours</b>   |   | <b>25</b> |           | -        |
| <b>Total No. of Tutorial Hours</b>  |   |           | <b>15</b> | -        |
| <b>Total No. of Practical Hours</b> |   |           |           | <b>0</b> |

**Textbook**

1. Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016
2. J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1<sup>st</sup> edition, 2008.

**Reference books**

1. Garry Chartand and Ping Zhang, Introduction to Graph Theory, Tata McGraw-Hill, 2006.
2. Frank Harary, Graph Theory, Narosa Publishing House, Latest edition.
3. R. Diestel, Graph Theory, free online edition, 2016: diestel-graph-theory.com/basic.html.
4. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd., 2001
5. Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd., 2010

**Online Resources:**

- <http://nptel.ac.in/courses.php?disciplineID=111>
- [http://www.class-central.com/subject/math\(MOOCs\)](http://www.class-central.com/subject/math(MOOCs))

**Code: BCS405C****Course: OPTIMIZATION TECHNIQUE****Credits: 3****L:T:P - 2:2:0****SEE: 50 Marks****CIE: 50 Marks****SEE Hours: 03****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Calculus, probability and statistics   |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>• Appreciate the importance of linear algebra in computer science and allied engineering science.</li> <li>• Gain the knowledge of linear algebra tools and concepts to implement them in their core domain.</li> <li>• Improve their mathematical thinking and acquire skills required for sustained lifelong learning.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Explain the concepts of vector spaces, subspaces, bases, dimension and their properties and Use matrices and linear transformations to solve the given problem | Analyze       |
| CO2 | Compute Eigenvalues and Eigenvectors for the linear transformations  | Apply         |
| CO3 | Determine orthogonality of inner product spaces  | Apply         |
| CO4 | Apply the optimization techniques to solve the problems.   | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | -   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | -    | -    | -    |
| CO2 | 2   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | -    | -    | 3    |
| CO3 | 2   | 3   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | 2    | 3    |
| CO4 | 3   | -   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | 3    | -    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

**Course Structure**

|                                   |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|-----------------------------------|---|----------------------|-----------------------|------------------------|
| <b>Module – 1 VECTOR CALCULUS</b> |   |                      |                       |                        |
| 1.1                               | Functions of several variables  | 1                    | -                     | -                      |
| 1.2                               | Differentiation and partial differentials   | 1                    | 1                     | -                      |
| 1.3                               | Gradients of vector-valued functions, gradients of matrices,                            | 1                    | 1                     | -                      |
| 1.4                               | Useful identities for computing gradients, linearization an multivariate Taylor series. | 2                    | 1                     | -                      |

| <b>Module – 2 APPLICATIONS OF VECTOR CALCULUS</b> |  |           |           |          |
|---|--|-----------|-----------|----------|
| 2.1   | Backpropagation and automatic differentiation                                  | 1         | 1         | -        |
| 2.2   | Gradients in a deep network  | 1         | 1         | -        |
| 2.3   | The Gradient of Quadratic Cost, Descending the Gradient of Cost                | 1         | 1         | -        |
| 2.4   | The Gradient of Mean Squared Error.  | 1         | 1         | -        |
| <b>Module – 3 Convex Optimization-1</b>           |  |           |           |          |
| 3.1   | Local and global optima  | 1         | -         | -        |
| 3.2   | Convex sets and functions separating   | 1         | 1         | -        |
| 3.3   | Hyperplanes application of Hessian matrix in optimization,                     | 1         | 1         | -        |
| 3.4   | Optimization using gradient descent  | 1         | 1         | -        |
| 3.5   | Sequential search 3- point search and Fibonacci search.                        | 1         | -         | -        |
| <b>Module – 4: Convex Optimization-2</b>          |  |           |           |          |
| 4.1   | Unconstrained optimization -Method of steepest ascent/descent                  | 2         | 1         | -        |
| 4.2   | NR method, Gradient descent  | 2         | 1         | -        |
| 4.3   | Mini batch gradient descent  | 1         | 1         | -        |
| 4.4   | Stochastic gradient descent.   | 1         | 1         | -        |
| <b>Module – 5 Advanced Optimization</b>           |  |           |           |          |
| 5.1   | Momentum-based gradient descent methods: Adagrad,                              | 2         | -         | -        |
| 5.2   | RMSprop and Adam.  | 2         | -         | -        |
| 5.3   | Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods. | 1         | 1         | -        |
| <b>Total No. of Lecture Hours</b>                 |  | <b>25</b> |           | -        |
| <b>Total No. of Tutorial Hours</b>                |  |           | <b>15</b> | -        |
| <b>Total No. of Practical Hours</b>               |  |           |           | <b>0</b> |

### Textbook

1. Mathematics for Machine learning, Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, 2020, Cambridge University Press.
2. S. Bubeck, Convex Optimization: Algorithms and Complexity, Foundations and Trends in Optimization, 2015.
3. S. Boyd, N. Parikh, and E. Chu, “Distributed optimization and statistical learning via the alternating direction method of multipliers”, Foundations and Trends in Machine Learning, Now Publishers Inc.

### Reference books

1. Linear Algebra and Optimization for Machine Learning, Charu C. Aggarwal, Springer, 2020.
2. A. Beck, First-Order Methods in Optimization, MOS-SIAM Series on Optimization, 2017.
3. F. Bach, “Learning with Submodular Functions: A Convex Optimization Perspective”, Foundations and Trends in Machine Learning, Now Publishers Inc.

### Online Resources:

- <https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/index.htm>
- <https://www.math.ucdavis.edu/~linear/linear.pdf>
- <https://www.coursera.org/learn/linear-algebra-machine-learning>

**Code: BCS405D****Course: LINEAR ALGEBRA****Credits: 3****L:T:P - 2:2:0****SEE: 50 Marks****CIE: 50 Marks****SEE Hours: 03****Max. Marks: 100**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | <b>Vector spaces, linear transformations</b>  |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>To equip the students with standard concepts and tools in Linear algebra which will find them useful in their disciplines.</li> <li>Gain the knowledge of linear algebra tools and concepts to implement them in their core domain.</li> <li>Improve their mathematical thinking and acquire skills required for sustained lifelong learning.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Explain the concepts of vector spaces, subspaces, bases, dimension and their properties and Use matrices and linear transformations to solve the given problem | Analyze       |
| CO2 | Compute Eigenvalues and Eigenvectors for the linear transformations  | Apply         |
| CO3 | Determine orthogonality of inner product spaces  | Apply         |
| CO4 | Apply the optimization techniques to solve the problems.   | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | 2   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO2 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO3 | 3   | 3   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |
| CO4 | 3   | -   | 3   | -   | -   | -   | -   | -   | -   | -    | -    | 3    | -    | -    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

**Course Structure**

|                                |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|--------------------------------|---|----------------------|-----------------------|------------------------|
| <b>Module-1: VECTOR SPACES</b> |   |                      |                       |                        |
| 1.1                            | Introduction, Vector spaces, Subspaces, Linear Combinations | 2                    | -                     |                        |
| 1.2                            | Linear Spans, row space and column space of a Matrix        | 1                    | 1                     |                        |
| 1.3                            | Linear Dependence and Independence,                         | 1                    | 1                     |                        |
| 1.4                            | Basis and Dimension, Coordinates                            | 1                    | -                     |                        |



**CourseCode:BCS456A****Course Name: Green IT and Sustainability****Credits: 1****L:T:P - 1:0:0****SEE:NA****CIE: 50 Marks**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | Nil   |
| <b>Learning objectives</b>  | This course will enable the students to: <ol style="list-style-type: none"> <li>Understand challenges for Green ICT and the environmental impact.</li> <li>Learnd if ferent aspects of ICT metric sand Sustainable Cloud Computing.</li> <li>Explore effects of software design on the sustainability.</li> </ol> |

**Course Outcomes:**

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

| COs |  | Bloom's level |
|-----|--|---------------|
| CO1 | Classify the challenges for Green ICT                                  | Understand    |
| CO2 | Relate the environmental impact due to emerging technologies.          | Apply         |
| CO3 | Demonstrate different aspects of ICT metrics.                          | Apply         |
| CO4 | Compare the various parameters related to Sustainable Cloud Computing. | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 |  | PO12 |  | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|--|------|--|------|------|
| CO1 | 3   | 2   | -   | -   | -   | -   | 3   | -   | -   | -    | -    |  | -    |  | 2    | -    |
| CO2 | 2   | -   | -   | -   | 3   | -   | 3   | -   | -   | -    | -    |  | -    |  | 3    | -    |
| CO3 | 2   | -   | -   | -   | 2   | -   | 2   | -   | -   | -    | -    |  | -    |  | 3    | -    |
| CO4 | 3   | 3   | 3   | 3   | 3   | -   | 3   | -   | -   | 2    | 3    |  | -    |  | 3    | 3    |

**S–Strong(3) M–Medium(2) L–Low(1)**

**Course Structure**

| <b>Module–1:Green ICT-History, Agenda, and Challenges Ahead</b>              |   | <b>No. of Lecture Hours</b> | <b>No. of Tutorial Hours</b> |
|--|---|-----------------------------|------------------------------|
| 1.1  | Introduction ,Industrial Revolution ,The Emergence ofInformationandCommunicationTechnologies,TheAgendaandChallengesAh ead.  | 1                           | -                            |
| 1.2  | Emerging Technologies and Their Environmental Impact: Introduction, Number of Connected Devices ,Increased,Functionality,IncreasedNumberofSeparateFunctions,IncreasedDe mandforSpeedandReliability , Obsolescence—The Problem of Backward Compatibility | 2                           | Nil                          |
| 1.3  | The Other Side of the Balance Sheet,VideoconferenceasanAlternativetoBusinessTravel,Dematerializationof ProductChain,TravelAdvice/RoadTrafficControl,IntelligentEnergyMetering, BuildingManagementSystems,SavingIT                                       | 2                           | Nil                          |
| <b>Module–2:Measurements and Sustainability, Sustainable Cloud Computing</b> |   |                             |                              |
| 2.1  | Introduction ,ICT Technical Measures, Ecological Measures and Ethical Consideration,SystemsEngineeringforDesigningSustainableICT- BasedArchitectures.   | 1                           | Nil                          |
| 2.2  | Sustainable Cloud Computing: Introduction, Challenges in the Use of Cloud Computing As Green Technology,Cloud Computing and Sustainability, Sustainable Applications of Cloud Computing,  | 2                           | Nil                          |

|  |  |    |     |
|--|--|----|-----|
| 2.3  | Technologies<br>Associated With Sustainable Cloud Computing, Future Prospects of Sustainable Cloud Computing, Reflections on Sustainable Cloud Computing Applications.           | 2  | Nil |
| <b>Module–3: Sustainable Software Design</b> |  |    |     |
| 3.1  | Overview and Scope, Evaluating Sustainability Effects , Sustainability and the Product Life Cycle , Direct Effects: Sustainability During Use, Runtime Energy Consumption Basics | 2  | Nil |
| 3.2  | Analyzing the Energy Consumption of an Application , Energy Consumption Reduction Using Physical Properties of Semiconductors  | 1  | Nil |
| 3.3  | Optimizing the Energy Consumption of an Application: Compiler Techniques, Optimizing the Energy Consumption of an Application: Runtime Approaches                                | 2  | Nil |
| <i>Total No. of Lecture Hours</i>            |  | 15 |     |
| <i>Total No. of Tutorial Hours</i>           |  |    | Nil |

**Textbooks:**

Green Information Technology–A Sustainable Approach , Mohammad Dastbaz Colin Pattinson, Babak Akhgar, Elsevier, 2015 Inc.

San Murugesan; G.R. Gangadharan, Harnessing Green IT: Principles and Practices, Wiley-IEEE Press

**Weblinks and Video Lectures (e-Resources):**

- [https://www.youtube.com/watch?v=kvn\\_-mJ2tSo](https://www.youtube.com/watch?v=kvn_-mJ2tSo)
- <https://www.youtube.com/watch?v=kxngsYn5N3Y>
- <https://www.youtube.com/watch?v=EgdFi3sCgzU>
- <https://www.brightest.io/sustainability-measurement>
- <https://www.youtube.com/watch?v=S2m49Op25Zw>

**Course Code: BCS456B****Course Name: Capacity Planning for IT Values and Professional Ethics****L:T:P - 1:0:0****CIE: 50 Marks****Credits: 1****SEE: NA**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Nil  |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>Understand requirement and measurements for capacity planning, measurement and monitoring.</li> <li>Measurement of data for prediction towards the planning process.</li> <li>Understand concepts related to deployment, installation, configuration, and management.</li> <li>Role of virtualization and cloud services in capacity planning.</li> </ul> |

**Course Outcomes:**

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

| COs |   | Bloom's level |
|-----|---|---------------|
| CO1 | Identify the requirement and measurements for capacity planning by considering the goal, issues, and processes and explain capacity measurement and monitoring. | Understand    |
| CO2 | Make use of measurement data for prediction towards overall planning process  | Apply         |
| CO3 | Explain the concepts related to deployment, installation, configuration, and management.  | Apply         |
| CO4 | Demonstrate how the virtualization and cloud services fit into a capacity plan.   | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | -   | 2   | -   | 3   | -   | -   | -   | 3   | 3   | -    | -    | -    | 3    | -    |
| CO2 | -   | 2   | -   | -   | -   | -   | -   | 3   | -   | -    | -    | -    | -    | -    |
| CO3 | 3   | -   | 3   | -   | -   | -   | -   | 3   | -   | -    | 3    | -    | 3    | -    |
| CO4 | 3   | 3   | 3   |     | 3   | -   | -   | 3   | -   | -    | 2    | -    | 2    |      |

**S – Strong (3) M – Medium (2) L – Low (1)**

**Course Structure**

| Module-1: Goals, Issues, and Processes        |  | No. of Lecture Hours | No. of Tutorial Hours |
|---|--|----------------------|-----------------------|
| 1.1   | <b>Goals, Issues, and Processes:</b> capacity planning, Quick and Dirty Math, Predicting When Your Systems Will Fail, Make Your System Stats Tell Stories, | 2                    | Nil                   |
| 1.2   | Buying Stuff: Procurement Is a Process, Performance and Capacity: Two Different Animals, The Effects of Social Websites and Open                           | 2                    | Nil                   |
| 1.3   | Setting Goals for Capacity: Different Kinds of Requirements and Measurements, Architecture Decisions   | 1                    | Nil                   |
| Module – 2: Measurement and Predicting Trends |  |                      |                       |
| 2.1   | API Usage and Its Effect on Capacity, Examples and Reality. Predicting Trends: Riding Your Waves.  | 2                    | Nil                   |
| 2.2   | Predicting Trends: Procurement, The Effects of Increasing Capacity, Long-Term Trends, Iteration and Calibration.   | 2                    | Nil                   |

|   |  |    |     |
|---|--|----|-----|
| 2.3   | <b>Deployment:</b> Automated Deployment Philosophies   | 1  | Nil |
| <b>Module – 3: Virtualization and Cloud Computing</b> |  |    |     |
| 3.1   | Automated Installation Tools, Automated Configuration  | 1  | Nil |
| 3.2   | Virtualization, Cloud Computing, Computing Resource Evolutions, Mixed Definitions,   | 2  | Nil |
| 3.3   | Cloud Capacity, Use it or lose it (your wallet), Measuring the clouds, Cloud Case Studies, Cloud Use Case: Anonymous Desktop Software Company. | 2  | Nil |
| <i>Total No. of Lecture Hours</i>                     |  | 15 |     |
| <i>Total No. of Tutorial Hours</i>                    |  |    | Nil |

**Textbooks:**

1. John Allspaw, The Art of Capacity Planning, 2008, O'Reilly

**Web links and Video Lectures (e-Resources):**

- <https://www.youtube.com/watch?v=w0cD26CLBA0>
- <https://www.youtube.com/watch?v=5-hhfBXykec>
- <https://www.youtube.com/watch?v=9e4IohiFmZ8&t=63s>
- <https://www.youtube.com/watch?v=qj4ziswxupE>
- <https://www.youtube.com/watch?v=jTW79ofC6Go>
- [https://www.youtube.com/watch?v=\\_pPlanX5wQY](https://www.youtube.com/watch?v=_pPlanX5wQY)

**Code: BCS456C****Course: UI/UX Lab****Credits: 01****L:T:P - 0:0:2****SEE: NA****CIE: 50 Marks****SEE Hours: NA****Max. Marks: 50**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | NIL   |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>Understand user experience design requirements, with design goals, metrics and targets.</li> <li>Explore different prototyping methods, UX design principles with case examples.</li> <li>Understand the role of design thinking concepts and mental models in UX design.</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Apply different UI design approaches for creating and designing different use cases. . | Apply         |
| CO2 | Illustrate the importance of user experience through stories.                          | Apply         |
| CO3 | Demonstrate different prototyping in relation to software engineering.                 | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | 2   | -   | 3   | -   | 2   | -   | -   | -    | 3    | 3    | -    | 2    |
| CO2 | 3   | -   | 3   | -   | 3   | -   | 2   | -   | -   | -    | 3    | 2    | -    | 2    |
| CO3 | 3   | -   | 3   | -   | 3   | -   | 2   | -   | 3   | 3    | 2    | 2    | -    | 2    |

**Mapping Strength: Strong – 3 Medium – 2 Low – 1**

| Sl. No.                                | CO's | Experiments   | No. of Practical Hours |
|--|------|---|------------------------|
| 1.                                     | CO1  | Create and Design A Logo For E-Commerce App.  | 2                      |
| 2.                                     | CO1  | Create and Design A Basic Email Template.   | 2                      |
| 3.                                     | CO1  | Design A Brochure That Showcases Different Features Of The E-Commerce App.  | 2                      |
| 4.                                     | CO2  | Create User Personas & User Stories To Define A Problem Statement, Scope And Understand The Problem.                              | 2                      |
| 5.                                     | CO2  | Create Sketches And Low-Fidelity Wireframes Of The Scoped Solution.   | 2                      |
| 6.                                     | CO2  | Create High-Fidelity Prototypes From The Wireframes.  | 2                      |
| 7.                                     | CO3  | Create The Basic Responsive Elements Like Buttons, Input Elements To Understand Frames, Groups And Layouts.                       | 2                      |
| 8.                                     | CO3  | Design A Basic Clickable Prototyping Using Figma.   | 2                      |
| 9.                                     | CO3  | Create A Design System For E Commerce App Using Grid And Spacing, Color System, And Ui Elements Like Icons, Images, Buttons, Etc. | 2                      |
| <b>Total number of practical hours</b> |      |   | <b>18</b>              |

**Textbook:**

1. REX HARTSON and PARDHA S. PYLA, The UX Book-Process and Guidelines for Ensuring a Quality User Experience, Morgan Kaufmann, Elsevier, 2012.

**Online Resources:**

1. <https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-withwireframe-prototype-figma/>
2. <https://www.edureka.co/blog/ui-ux-design-tutorial/>  
<https://www.udemy.com/course/introtoux/>

**Code: BCS456D****Course: Technical Writing using LaTeX****Credits: 1****L: T: P – 0:0:2****SEE: NA****CIE: 50 Marks****SEE Hours:NA****Max. Marks: 50**

|                             |   |
|-----------------------------|---|
| <b>Prerequisites if any</b> | Fundamentals of Web development and Client server technology  |
| <b>Learning objectives</b>  | <ul style="list-style-type: none"> <li>To introduce the basic syntax and semantics of the LaTeX scripting language</li> <li>To understand the presentation of tables and figures in the document</li> <li>To illustrate the LaTeX syntax to represent the theorems and mathematical equations</li> <li>To make use of the libraries (Tikz, algorithm) to design the diagram and algorithms in the document</li> </ul> |

**Course Outcomes:**

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

| COs | Course Outcomes  | Bloom's level |
|-----|--|---------------|
| CO1 | Understand the basic LaTeX command to develop simple document  | Understanding |
| CO2 | Develop LaTeX script to present the tables, figures, theorems and mathematical equations in the document | Apply         |
| CO3 | Develop programs to generate the complete report with citations and a bibliography                       | Apply         |
| CO4 | Illustrate the use of Tikz and algorithm libraries to design graphics and algorithms in the document     | Analyze       |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 2   | -   | -   | -   | 3   | -   | -   | -   | -   | 3    | -    | 3    | -    | -    |
| CO2 | 2   | -   | -   | -   | 3   | -   | -   | -   | -   | 3    | -    | 2    | -    | -    |
| CO3 | 2   | -   | -   | -   | 3   | -   | -   | -   | -   | 3    | -    | 2    | -    | -    |
| CO4 | 3   | -   | -   | -   | 3   | -   | -   | -   | -   | 3    | -    | 2    | -    | -    |

**Mapping Strength:****Strong – 3****Medium – 2****Low – 1****Course Structure**

| SL. No. | CO  | Experiments  | No. of Practical Hours |
|---------|-----|--|------------------------|
| 1       | CO1 | a. Develop a LaTeX script to create a simple document that consists of 2 sections [Section1, Section2], and a paragraph with dummy text in each section. And also include header [title of document] and footer [institute name, page number] in the document. | 2                      |
| 2       | CO1 | a. Develop a LaTeX script to create a document that displays the sample Abstract/Summary   | 2                      |
| 3       | CO1 | Develop a LaTeX script to create a simple title page of the VTU project Report [Use suitable Logos and text formatting]  | 1                      |
| 4       | CO2 | Develop a LaTeX script to create the Certificate Page of the Report [Use suitable commands to leave the blank spaces for user entry]   | 2                      |

|                                     |                |   |                |                     |                 |   |                 |                 |
|-------------------------------------|----------------|---|----------------|---------------------|-----------------|---|-----------------|-----------------|
| 5                                   | CO2            | Develop a LaTeX script to create a document that contains the following table with proper labels.   |                |                     |                 | 2 |                 |                 |
|                                     |                | <b>S.No</b>   | <b>USN</b>     | <b>Student Name</b> | <b>Marks</b>    |   |                 |                 |
|                                     |                |   |                |                     | <b>Subject1</b> |   | <b>Subject2</b> | <b>Subject3</b> |
|                                     |                | 1   | 4XX22XX00<br>1 | Name 1              | 89              |   | 60              | 90              |
|                                     |                | 2   | 4XX22XX00<br>2 | Name 2              | 78              |   | 45              | 98              |
| 3                                   | 4XX22XX00<br>3 | Name 3  | 67             | 55                  | 59              |   |                 |                 |
| 6                                   | CO2            | Develop a LaTeX script to include the side-by-side graphics/pictures/figures in the document by using the subgraph concept                                |                |                     |                 | 2 |                 |                 |
| 7                                   | CO1            | $= \frac{-2 \pm \sqrt{4+32}}{2 \times 1}$ to create<br>ons<br>$= \frac{-2 \pm \sqrt{4+32}}{2}$  |                |                     |                 | 2 |                 |                 |
| 8                                   | CO3            | Develop a LaTeX script to demonstrate the presentation of Numbered theorems, definitions, corollaries, and lemmas in the document                         |                |                     |                 | 2 |                 |                 |
| 9                                   | CO4            | Develop a LaTeX script to create a document that consists of two paragraphs with a minimum of 10 citations in it and display the reference in the section |                |                     |                 | 2 |                 |                 |
| 10                                  | CO4            | Develop a LaTeX script to design a simple tree diagram or hierarchical structure in the document with appropriate labels using the Tikz library           |                |                     |                 | 1 |                 |                 |
| <b>Total No. of Practical Hours</b> |                |   |                |                     | <b>18</b>       |   |                 |                 |

**Textbook:**

1. A Short Introduction to LaTeX BY FIRUZA KARMALI (AIBARA), A book for beginners, 2019
2. Formatting Information: A Beginner's Introduction to Typesetting with LaTeX, BY PETER FLYNN, Comprehensive TeX Archive Network (2005)

**Online Resources:**

LaTeX TUTORIAL: [<https://latex-tutorial.com/tutorials/>]  
LaTeX TUTORIAL: [<https://www.javatpoint.com/latex>]

**Code: BCS456E****Course: UNIX and Shell Programming****Credits: 1****CIE: 50 Marks****L: T:P - 0:0:2****SEE: NA****SEE Hours: NA****Total Marks: 50**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | <b>NIL</b>   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>Understand the Unix architecture, file system, and multi-user environment.</li> <li>Manage file permissions and ownership effectively, Write and execute basic shell scripts using variables, loops, and conditionals.</li> </ol> |

**Course Outcomes:**

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

| <b>COs</b> | <b>Course Outcomes</b>   | <b>Bloom's Level</b> |
|------------|--|----------------------|
| CO1        | To understand the UNIX environment with basic commands, file systems and multi-user environment.                     | Apply                |
| CO2        | To understand the concepts of filters and regular expressions, acquire knowledge about Regular expressions and grep. | Apply                |
| CO3        | To acquire knowledge about awk command and to acquire programming skills in shell scripts.                           | Apply                |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | -   | -   | 1   | -   | -   | -   | 1   | -    | 3    | 1    | 1    | 3    |
| CO2 | 3   | 2   | -   | 2   | 1   | -   | -   | 2   | 1   | -    | 3    | 1    | 2    | 3    |
| CO3 | 2   | 1   | 1   | -   | 1   | -   | -   | -   | 1   | -    | 3    | 1    | 3    | 2    |

**Mapping Strength: Strong – 3      Medium – 2      Low – 1**

**List of Experiments**

| Sl. No.                                | CO's | Experiments  | No. of Practical Hours |
|--|------|--|------------------------|
| 1                                      | CO1  | Illustrate the Usage of basic Unix Shell Commands: <i>cal, who, whoami, passwd, echo, man, lpr</i> .<br>Other commands: <i>tty, stty, script, uname, bc</i> (calculator).  | 1                      |
| 2                                      | CO1  | Illustrate the functionality of the following with appropriate commands<br>(i) Wildcards,<br>(ii) Different types of File Pathname   | 1                      |
| 3                                      | CO1  | Demonstrate the workings of commands <i>pwd, ls, mkdir, rmdir, cd, vi, more, lpr, cp, mv, ln, rm, find</i> .   | 1                      |
| 4                                      | CO2  | Use the security and file permission commands <i>group, chmod</i> , symbolic and octal codes, <i>umask, chown, chgrp</i> to demonstrate the multi-user environment.  | 1                      |
| 5                                      | CO2  | Implement the Filter command such as <i>cat, cmp, comm, cut, diff, paste, head, sort, tail, tr, wc</i> .   | 2                      |
| 6                                      | CO2  | Implement the following with <i>grep</i> command:<br>(i) Case insensitive search, Print n Specific Lines from a File.<br>(ii) Displaying the Count of Number of Matches.<br>(iii) Display the File Names that Matches the Pattern.<br>(iv) Checking for the Whole Words in a File.<br>(v) Displaying only the matched pattern.<br>(vi) Search Recursively for a Pattern in the Directory.<br>(vii) Matching the Lines that End with a String<br>(viii) Inverting the Pattern Match | 1                      |
| 7                                      | CO3  | Write an awk script to compute gross salary of an employee accordingly to rule given below. If basic salary is < 10000 then HRA=15% of basic & DA=45% of basic If basic salary is >=10000 then HRA=20% of basic & DA=50% of basic.   | 2                      |
| 8                                      | CO3  | Write an awk script that accepts date argument in the form of dd-mm-yy and displays it in the form if month, day and year. The script should check the validity of the argument and in the case of error, display a suitable message.  | 2                      |
| 9                                      | CO3  | Shell Programming: shell script exercise based on following:<br><ul style="list-style-type: none"> <li>• Positional parameters</li> <li>• Arithmetic</li> <li>• If-then-fi, if-then-else-fi, nested if-else</li> <li>• Logical operators</li> <li>• Else + if equals elif, case structure</li> <li>• While ,for loop</li> </ul>  | 2                      |
| 10                                     | CO3  | Write a Shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.  | 2                      |
| <b>Total number of practical hours</b> |      |  | <b>15</b>              |

**Textbook:**

1. UNIX and Shell Programming – A Textbook by Behrouz A Forouzan, Richard F Gilberg, Cengage Learning, I Edition, 2003.

**Reference Book:**

1. The Complete Reference UNIX by Kenneth Rosen, Douglas Host, James Farber and Richard Rosinski, Tata McGraw- Hill, Edition 2000.
2. E-book: Shell Scripting – Expert Recipes for Linux, Bash and More by Steve Parker, Wrox Publications.
3. E-book: Linux Shell Scripting Cookbook by Shantanu Tushar and Sharath Lakshman, II edition, Packt Publications, 2013.

**Course Code:** BBOK407  
**Credits:** 2  
**SEE:** 50 Marks  
**SEE Hours:** 2 Hrs

**Course:** Biology for Engineers  
**L:T:P:S** 2:0:0  
**CIE:** 50 Marks  
**Max.Marks:** 100

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | None   |
| <b>Learning objectives</b>  | 1. Review the basics of cell biology and role of biomolecules.<br>2. Elucidate the significance of Biomechanics and Biomaterials |

#### Course Outcomes:

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

| Course Outcomes |  | Bloom's level |
|-----------------|--|---------------|
| CO1             | Explain the fundamentals of Life, Evolution, Biomolecules, Cell Biology & Genetics | Understand    |
| CO2             | Outline the physical principles regulating the systems of the human bodies         | Understand    |
| CO3             | Describe the impact of Biomaterials on the fields of Engineering & Medicine        | Understand    |
| CO4             | Summarize tissue engineering and clinical applications of materials                | Understand    |

#### Mapping with POs and PSOs:

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|
| CO1 | 3   | -   | -   | -   | -   | 3   | 2   | -   | -   | -    | -    | 3    | -    | -    |
| CO2 | 3   | -   | -   | -   | -   | 3   | 2   | -   | -   | -    | -    | 3    | -    | -    |
| CO3 | 3   | -   | -   | -   | -   | 3   | 2   | -   | -   | -    | -    | 3    | -    | -    |
| CO4 | 3   | -   | -   | -   | -   | 3   | 2   | -   | -   | -    | -    | 3    | -    | -    |

**Strong: 3      Medium: 2      Low: 1**

#### Course Content

| Sl.No             | Module - 1  | No. of Lecture Hours | No. of Tutorial Hours | Self-Learning Hours |
|-------------------|---|----------------------|-----------------------|---------------------|
| 1.1               | <b>Cell basic unit of life:</b> Introduction, Origin, and evaluation of life. Structure and functions of a cell. Stem cells and their application.  | 2                    | -                     | -                   |
| 1.2               | <b>Biomolecules:</b> Properties and functions of Carbohydrates, Nucleic acids, proteins, lipids. Importance of special biomolecules: Properties and functions of enzymes, vitamins, and hormones. | 3                    | -                     | -                   |
| <b>Module - 2</b> |   |                      |                       |                     |
| 2.1               | <b>Application of biomolecules:</b> Carbohydrates in cellulose-based water filters production, PHA and PLA in bioplastics production, Nucleic acids in vaccines and diagnosis,                    | 2                    | -                     | -                   |
| 2.2               | Proteins in food production, lipids in biodiesel and detergents production, Enzymes in biosensors fabrication, food processing, detergent formulation, and textile processing.                    | 3                    | -                     | -                   |
| <b>Module - 3</b> |   |                      |                       |                     |
| 3.1               | <b>Adaptation of anatomical principles for bioengineering design:</b> Brain as a CPU system, Eye as a Camera system, Heart as a pump system.  | 3                    | -                     | -                   |
| 3.2               | Lungs as purification system, Kidney as a filtration system.  | 2                    | -                     | -                   |
| <b>Module - 4</b> |   |                      |                       |                     |
| 4.1               | <b>Nature-bioinspired materials and mechanisms:</b> Echolocation, Photosynthesis, Bird flying, Lotus leaf effect, Plant burrs, Shark skin, Kingfisher beak.                                       | 3                    | -                     | -                   |

|   |  |    |   |   |
|---|--|----|---|---|
| 4.2                                     | Human Blood substitutes - haemoglobin-based oxygen carriers (HBOCs) and perfluorocarbons (PFCs).   | 2  | - | - |
| <b>Module - 5</b>                       |  |    |   |   |
| 5.1                                     | <b>Trends in bioengineering:</b> Muscular and Skeletal Systems as scaffolds, and tissue engineering, Bioprinting techniques and materials. | 3  | - | - |
| 5.2                                     | Electrical tongue and electrical nose in food science, DNA origami and   | 2  | - | - |
|   | Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis, Bio concrete, Bioremediation, Biomining.                       |    |   |   |
| <b>Total No. of Lecture Hours</b>       |  | 25 | - | - |
| <b>Total No. of Tutorial Hours</b>      |  |    | - | - |
| <b>Total No. of Self learning Hours</b> |  |    |   | - |

**Detailed Lesson Plan:**

| Sr No. of Module | Number of related learning Objectives | Weeks / Dates | Online Mode  |  | ICT Tool/ Platform/ LMS  | Face-to-face Mode           |   |
|------------------|---------------------------------------|---------------|--|--|--------------------------|-----------------------------|---|
|                  |                                       |               | Resource (OER/ URL/ IM/ CP)                        | Activity (Describe activity in detail) |                          | Resource (OER/ URL/ IM/ CP) | Activity                                |
| 1.1              | 1                                     | 1             | Refer succeeding section titled 'Online Resources' | -                                      | PPT, Smart Board, Moodle | -                           | Student Led Discussions & Presentations |
| 1.2              | 1                                     | 1             |  | -                                      |                          |                             |   |
| 2.1              | 1                                     | 1             |  | -                                      |                          |                             |   |
| 2.2              | 1                                     | 1             |  | -                                      |                          |                             |   |
| 3.1              | 1                                     | 2             |  | -                                      |                          |                             |   |
| 3.2              | 1                                     | 2             |  | -                                      |                          |                             |   |
| 4.1              | 2                                     | 1             |  | -                                      |                          |                             |   |
| 4.2              | 2                                     | 1             |  | -                                      |                          |                             |   |
| 5.1              | 2                                     | 2             |  | -                                      |                          |                             |   |
| 5.2              | 2                                     | 2             |  | -                                      |                          |                             |   |

**Assessment Pattern:**

| Bloom's level | Continuous Internal Examination |        |                     | End Semester Examination |
|---------------|---------------------------------|--------|---------------------|--------------------------|
|               | Test 1                          | Test 2 | Assignment/Quiz/AAT |                          |
| Remember      | √                               | √      | √                   | √                        |
| Understand    | √                               | √      | √                   | √                        |
| Apply         |                                 |        |                     |                          |
| Analyze       |                                 |        |                     |                          |
| Evaluate      |                                 |        |                     |                          |
| Create        |                                 |        |                     |                          |

**Text Books:**

1. Biology for Engineers by G. K. Suraishkumar; Oxford University Press, 2019, First Edition

**Reference Books:**

1. Introductory Biomechanics: From Cells to Organisms by C. Ross Ethier and Craig A. Simmons; Cambridge University Press, 2012, Online Edition
2. Introduction to Biomaterials: Basic Theory with Engineering Applications, J. L. Ong, Mark R. Appleford, Gopinath Mani, Cambridge University Press, 2014, First Edition
3. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao, N Publishing, Bengaluru, 2023.
4. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
5. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
6. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
7. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
8. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
9. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
10. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
11. Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N Geetha A C Udayashankar Lambert Academic Publishing, 2019.
12. 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
13. Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016

**Online Resources:**

1. NOC: Biology for engineers and other non-biologists, IIT Madras; Dr. Madhulika Dixit, Prof. G.K. Suraishkumar, <https://nptel.ac.in/courses/121106008>
2. Introduction To Biological Engineering Design, MIT Open Courseware, <https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009>
3. Introduction To Bioengineering, MIT Open Courseware, <https://ocw.mit.edu/courses/20-010j-introduction-to-bioengineering-be-010j-spring-200>

**Course Code: BUHK408****Credits: 1****L:T:P:1:0:0****SEE: 50 Marks****Course Name: Universal Human Values and Professional Ethics****CIE: 50 Marks****SEE Hrs: 2**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | Nil  |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity, which are the core aspirations of all human beings.</li> <li>To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence.</li> <li>To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with nature</li> </ol> |

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

| COs |   | Bloom's level |
|-----|---|---------------|
| CO1 | To understand the core aspirations of all human beings                                      | Understand    |
| CO2 | To gain the universal human values and movement towards value-based living in a natural way | Apply         |
| CO3 | To fulfilling the human behavior and mutually enriching interaction with nature             | Apply         |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |  | PSO1 | PSO2 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|--|------|------|
| CO1 | -   | -   | -   | -   | -   | 3   | 1   | 2   | 3   | 3    | 1    | 2    |  | -    | -    |
| CO2 | -   | -   | -   | -   | -   | 3   | 2   | 2   | 3   | 3    | 1    | 2    |  | -    | -    |
| CO3 | -   | -   | -   | -   | -   | 3   | 2   | 2   | 3   | 3    | 1    | 2    |  | -    | -    |

**S – Strong (3) M – Medium (2) L – Low (1)****Course Structure**

| Module – 1: Introduction - Need, Basic Guidelines, Content and Process for Value Education |   | No. of Lecture Hours | No. of Tutorial Hours |
|--|---|----------------------|-----------------------|
| 1.1  | Understanding the need, basic guidelines, content and process for Value Education   | 1                    | Nil                   |
| 1.2  | Self-Exploration–what is it? - its content and process; 'Natural Acceptance' and Experiential Validation- as the mechanism for self-exploration                   | 1                    | Nil                   |
| 1.3  | Right understanding, Relationship and Physical Facilities- the basic requirements for fulfillment of aspirations of every human being with their correct priority | 1                    | Nil                   |
| 1.4  | Method to fulfill the above human aspirations: understanding and living in harmony at various levels . Practice session   | 2                    | Nil                   |
| <b>Module – 2: Understanding Harmony in Myself, Family, Society and Human Relationship</b> |   |                      |                       |

|   |  |    |     |
|---|--|----|-----|
| 2.1   | Understanding human being as a co-existence of the sentient 'I' and the material 'Body', Understanding the needs of Self ('I') and 'Body' - Sukh and Suvidha   | 1  | Nil |
| 2.2   | Understanding Harmony in the family – the basic unit of human interaction. Understanding values in human-human relationship; meaning of Nyaya and program for its fulfillment to ensure Ubhay-tripti; Trust (Vishwas) and Respect (Samman) as the foundational values of relationship        | 1  | Nil |
| 2.3   | Understanding the meaning of Vishwas and Samman; Difference between intention and competence; respect and differentiation ; Understanding the harmony in the society (society being an extension of family): Samadhan, Samridhi, Abhay, Sah-astitva as comprehensive Human Goals             | 1  | Nil |
| 2.4   | Visualizing a universal harmonious order in society- Undivided Society (Akhand Samaj), Universal Order (Sarvabhaum Vyawastha )- from family to world family! Practice session  | 2  | Nil |
| <b>Module – 3:</b>  |  |    |     |
| <b>Understanding Harmony in the Nature, Existence and Implications of the all Holistic on Professional Ethics</b> |  |    |     |
| 3.1   | Understanding the harmony in the Nature, Interconnectedness and mutual fulfillment among the four orders of nature recyclability and self-regulation in nature   | 1  | Nil |
| 3.2   | Understanding Existence as Co-existence (Sah-astitva) of mutually interacting units in all-pervasive space.  | 1  | Nil |
| 3.3   | Competence in professional ethics:<br>a) Ability to utilize the professional competence for augmenting universal human order<br>b) Ability to identify and develop appropriate technologies and management patterns for above production systems.  | 1  | Nil |
| 3.4   | Strategy for transition from the present state to Universal Human Order:<br>a) At the level of individual: as socially and ecologically responsible engineers, technologists and managers b) At the level of society: as mutually enriching institutions and organizations. Practice session | 2  | Nil |
| <b>Total No. of Lecture Hours</b>   |  | 15 |     |
| <b>Total No. of Tutorial Hours</b>  |  |    | Nil |

## Guidelines and Content for Practice Sessions

### Practice Session 1:

Introduce yourself in detail. What are the goals in your life? How do you set your goals in your life? How do you differentiate between right and wrong? What have been your achievements and shortcomings in your life? Observe and analyze them.

**Expected outcome:** The students start exploring themselves; get comfortable to each other and to the teacher and start finding the need and relevance for the course.

### Practice Sessions 2:

1. a. Observe that any physical facility you use, follows the given sequence with time: Necessary & tasteful → unnecessary & tasteful → unnecessary & tasteless → intolerable b. In contrast, observe that any feeling in you is either naturally acceptable or not acceptable at all. If naturally acceptable, you want it continuously and if not acceptable, you do not want it any moment!
2. List down all your activities. Observe whether the activity is of 'I' or of Body or with the participation of both 'I' and Body.

#### Expected outcome:

1. The students are able to see that all physical facilities they use are required for a limited time in a limited quantity. Also they are able to see that in case of feelings, they want continuity of the naturally acceptable feelings and they do not want feelings which are not naturally acceptable even for a single moment.
2. The students are able to see that activities like understanding, desire, thought and selection are the activities of 'I' only, the activities like breathing, palpitation of different parts of the body are fully the activities of the body with the acceptance of 'I' while the activities they do with their sense organs like hearing through ears, seeing through body, sensing through touch, tasting through tongue and smelling through nose or the activities they do with their work organs like hands, legs etc. are such activities that require the participation of both 'I' and body

### Practice Session 3:

Form small groups in the class and in that group initiate dialogue and ask the eight questions related to trust. The eight questions are:

- 1a. Do I want to make myself happy
- 2a. Do I want to make others happy?
- 3a. Does the other want to make him happy?
- 4a. Does the other want to make me happy?

What is the answer? Intention (Natural Acceptance)

- 1b. Am I able to make myself always happy?
- 2b. Am I able to make others happy?
- 3b. Is the other able to make him always happy?
- 4b. Is the other able to make me always happy?

What is the answer? Competence

Let each student answer the questions for himself and everyone else. Discuss the difference between intention and competence. Observe whether you evaluate your intention & competence as well as the others' intention & competence

**Expected outcome:**

The students are able to see that the first four questions are related to our Natural Acceptance i.e. Intention and the next four to our Competence. They are able to note that the intention is always correct, only competence is lacking! We generally evaluate ourselves on the basis of our intention and others on the basis of their competence! We seldom look at our competence and others' intention as a result we conclude that I am a good person and other is a bad person.

**Textbooks:**

1. R.R Gaur, R Sangal, G P Bagaria, A foundation course in Human Values and professional Ethics, Excel books, New Delhi, 2010, ISBN 978-8-174-46781-2.

**References:**

1. IIT Delhi, Modern Technology – the Untold Story

**Course Code: BNSK459****Credits: Zero****SEE: NA****SEE Hours: NA****Course: National Service Scheme (NSS)****L:T:P 0:0:2****CIE: 100 Marks****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | 1. Students should have a service oriented mind set and social concern.<br>2. Students should have dedication to work at any remote place, anytime with available resources and proper time management for the other works.<br>3. Students should be ready to sacrifice some of the time and wishes to achieve service oriented targets on time  |
| <b>Learning objectives</b>  | 1. Understand the community in which they work<br>2. Identify the needs and problems of the community and involve them in problem- solving<br>3. Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems<br>4. Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes<br>5. Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony |

**Course Outcomes:**

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

| COs | Course Outcomes   |
|-----|---|
| CO1 | Understand the importance of his / her responsibilities towards society.  |
| CO2 | Analyse the environmental and societal problems/issues and will be able to design solutions for the same  |
| CO3 | Evaluate the existing system and to propose practical solutions for the same for Sustainable development and Implement government or self-driven projects effectively in the field. |

**Mapping with POs and PSOs:**

| COs | PO1 | PO2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO1 0 | PO1 1 | PO1 2 | PSO 1   | PSO 2 | PSO 3 | PSO 4 |  |
|-----|-----|-----|------|------|------|------|------|------|------|-------|-------|-------|---|-------|-------|-------|--|
| CO1 | -   | -   | -    | -    | -    | 3    | 2    | 2    | -    | 2     | -     | 2     | To be identified for each branch by Course Instructor |       |       |       |  |
| CO2 | -   | -   | -    | -    | -    | 3    | 2    | 2    | -    | 2     | -     | 2     |   |       |       |       |  |
| CO3 | -   | -   | -    | -    | -    | 3    | 3    | 2    | -    | 2     | -     | 2     |   |       |       |       |  |

**Mapping Strength: 2 Strong: 3 Medium –2 Low – 1**

**Course Structure**

|                       |   | No. of Lecture Hours | No. of Tutorial Hours | No. of Practical Hours |
|-----------------------|---|----------------------|-----------------------|------------------------|
| <b>List of Events</b> |   |                      |                       |                        |
| 1                     | Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing.                            |                      |                       |                        |
| 2                     | Waste management– Public, Private and Govt organization, 5 R's.   |                      |                       |                        |
| 3                     | Setting of the information imparting club for women leading to contribution in social and economic issues.            |                      |                       |                        |
| 4                     | Water conservation techniques – Role of different stakeholders– Implementation.                                       |                      |                       |                        |
| 5                     | Preparing an actionable business proposal for enhancing the village income and approach for implementation.           |                      |                       |                        |
| 6                     | Helping local schools to achieve good results and enhance their enrolment in Higher/ technical/ vocational education. |                      |                       |                        |
| 7                     | Developing Sustainable Water management system for rural areas and implementation approaches.                         |                      |                       |                        |

|                                     |  |   |   |           |
|-------------------------------------|--|---|---|-----------|
| 8                                   | Contribution to any national level initiative of Government of India. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc. |   |   |           |
| 9                                   | Spreading public awareness under rural outreach programs.(minimum5 programs)   |   |   |           |
| 10                                  | Social connect and responsibilities.   |   |   |           |
| 11                                  | Plantation and adoption of plants.   |   |   |           |
| 12                                  | Govt. school Rejuvenation and helping them to achieve good infrastructure  |   |   |           |
| 13                                  | Sustainable agriculture practices - Organic farming, Agroforestry and crop rotation.   |   |   |           |
| 14                                  | Rural finance – role of local bodies, need for agricultural finance and sources of agricultural finances.  |   |   |           |
| 15                                  | Strategies for the development of rural markets and emerging issues in rural marketing   |   |   |           |
| 16                                  | Rural energy system – conventional and non-conventional, Rural electrification-policies, achievements and targets.   |   |   |           |
| 17                                  | Livestock economies - fishery and poultry development, forestry and horticulture.  |   |   |           |
| 18                                  | Role of NGO's in rural development, the role of voluntary organization in India's development processes.   |   |   |           |
| 19                                  | Issues in rural industrialization and development of agro-based industries, rural non-farm sector  |   |   |           |
| <b>Total No. of Lecture Hours</b>   |  | - | - | -         |
| <b>Total No. of Tutorial Hours</b>  |  | - | - | -         |
| <b>Total No. of Practical Hours</b> |  |   |   | <b>26</b> |

**ONENSS – CAMP @ College /University /State or Central Govt Level /NGO's /General Social Camps**

- Students have to take up anyone activity on the above said topics and have to prepare content for awareness and technical contents for implementation of the projects and have to present strategies for implementation of the same. Compulsorily students have to attend one camp.
- CIE will be evaluated based on their presentation, approach and implementation strategies.

**Suggested Learning Resource:**

1. NSS Course Manual, Published by NSS Cell, VTU Belagavi.

**Course Code: BPEK459****Course: Physical Education (Sports & Athletics) – II****Credits: Zero****L:T:P 0:0:2****SEE: NA****CIE: 100 Marks****SEE Hours: NA****Max. Marks: 100****Course Outcomes:** At the end of the course, the student will be able to

1. Understand the ethics and moral values in sports and athletics
2. Perform in the selected sports or athletics of student's choice.
3. Understand the roles and responsibilities of organisation and administration of sports and games.

|  |                 |
|--|-----------------|
| <b>Module IV : Ethics and Moral Values</b>   | <b>5Hours</b>   |
| A. Ethics in Sports<br>B. Moral Values in Sports and Games   |                 |
| <b>Module V : Specific Games ( Any one to be selected by the student)</b>  | <b>20 Hours</b> |
| A. Volleyball – Attack, Block, Service, Upper Hand Pass and Lower hand Pass.<br>B. Throwball – Service, Receive, Spin attack, Net Drop & Jump throw.<br>C. Kabaddi – Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus.<br>D. Kho-Kho – Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 Up.<br>E. Table Tennis – Service (Fore Hand & Back Hand), Receive (Fore Hand & Back Hand), Smash.<br>F. Athletics (Track / Field Events) – Any event as per availability of Ground. |                 |
| <b>Module VI: Role of Organisation and administration</b>  | <b>5 Hours</b>  |

**Scheme and Assessment for auditing the course and Grades:**

| <b>Sl. No.</b> | <b>Activity</b>                             | <b>Marks</b> |
|----------------|---|--------------|
| 1.             | Participation of student in all the modules | 20           |
| 2.             | Quizzes – 2, each of 15 marks               | 30           |
| 3.             | Assignment                                  | 50           |
| <b>Total</b>   |   | <b>100</b>   |

**Course Code: BYOK459****Credits: Zero****SEE: NA****SEE Hours: NA****Course: Yoga****L:T:P 0:0:2****CIE: 100 Marks****Max. Marks: 100**

|                             |  |
|-----------------------------|--|
| <b>Prerequisites if any</b> | None   |
| <b>Learning objectives</b>  | <ol style="list-style-type: none"> <li>To enable the student to have good health and mental hygiene.</li> <li>To possess emotional stability</li> <li>To integrate moral values</li> <li>To attain higher level of consciousness.</li> </ol> |

**Course Outcomes:**

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

| Course Outcomes |  | Bloom's level |
|-----------------|--|---------------|
| CO1             | Understand the meaning of Yoga, its origin, history, development and importance. | Understand    |
| CO2             | Perform various Surya namaskar and able to Teach its benefits                    | Apply         |
| CO3             | Perform various asanas and able to Teach its benefits                            | Apply         |
| CO4             | Understand Benefits of Yoga on fitness and health                                | Apply         |

**Course Content**

| Sl.No   | Module – 1   | No. of Lecture and Practical Hours | No. of Tutorial Sessions |
|---|--|------------------------------------|--------------------------|
| 1.1   | Role of yoga in controlling diseases   | 1                                  | -                        |
| 1.2   | Patanjali's Ashtanga Yoga, its need and importance.                                      | 1                                  | -                        |
| 1.3   | Yama :Ahimsa, satya, asteya, brahmacarya, aparigraha                                     | 1                                  | -                        |
| 1.4   | Niyama :shoucha, santosh, tapa, svaadhyaya, Eshvarapranidhan                             | 1                                  | -                        |
| <b>Module – 2</b>                               |  |                                    |                          |
| 2.1   | Warmup Exercise  | 2                                  | -                        |
| 2.2   | Yoga jogging   | 2                                  | -                        |
| 2.3   | Suryanamaskar 12 count- 4 rounds of practice   | 2                                  | -                        |
| 2.4   | Asana its meaning by name, technique, precautionary measures and benefits of each asana. | 2                                  | -                        |
| 2.5   | Sitting: 1.Sukhasana 2. Paschimottanasana 3.Bharadwajasana                               | 2                                  | -                        |
| 2.6   | Standing: 1. Ardhakati Chakrasana 2. Parshva Chakrasana                                  | 2                                  | -                        |
| 2.7   | Prone line: 1.Makarasana 2.Dhanurasana   | 2                                  | -                        |
| 2.8   | Supine line 1. Halasana 2. Karna Peedasana   | 2                                  | -                        |
| <b>Module – 3</b>                               |  |                                    |                          |
| 3.1   | Pranayama – Suryanuloma, Chandranuloma,  | 2                                  | -                        |
| 3.2   | Suryabhedana, Chandra Bhedana, Nadishodhana  | 2                                  | -                        |
| <b>Total No. of Lecture and practical Hours</b> |  | <b>24</b>                          | <b>-</b>                 |
| <b>No. of Tutorial Sessions</b>                 |  | <b>Nil</b>                         |                          |

**Detailed Lesson Plan**

| Sl. No. of Module | Number of related learning Objectives | Weeks / Dates | Online Mode                 |  | ICT Tool/ Platform / LMS | Face-to-face Mode           |                           | Duration in Minutes |
|-------------------|---------------------------------------|---------------|-----------------------------|--|--------------------------|-----------------------------|---------------------------|---------------------|
|                   |                                       |               | Resource (OER/ URL/ IM/ CP) | Activity (Describe activity in detail) |                          | Resource (OER/ URL/ IM/ CP) | Activity                  |                     |
| 1.1               | 1 & 4                                 | W 1           |                             |  | PPT, SMART BOARD, MOODLE |                             | Explanation               | 60                  |
| 1.2               | 1 & 4                                 | W 1           |                             |  |                          |                             | Explanation               | 60                  |
| 1.3               | 1 & 4                                 | W 2           |                             |  |                          |                             | Explanation               | 60                  |
| 1.4               | 1 & 4                                 | W 2           |                             |  |                          |                             | Explanation               | 60                  |
| 1.5               | 1 & 4                                 | W 3           |                             |  |                          |                             | Explanation               | 60                  |
| 1.6               | 1 & 4                                 | W 3           |                             |  |                          |                             | Explanation               | 60                  |
| 2.1               | 2 & 4                                 | W 4           |                             |  |                          |                             | Practicing Surya namaskar | 60                  |
| 2.2               | 2 & 4                                 | W 4           |                             |  |                          |                             | Practicing Surya namaskar | 60                  |
| 2.3               | 2 & 4                                 | W 5           |                             |  |                          |                             | Practicing Surya namaskar | 60                  |
| 2.4               | 2 & 4                                 | W 5           |                             |  |                          |                             | Practicing Surya namaskar | 60                  |
| 2.5               | 2 & 4                                 | W 6           |                             |  |                          |                             | Practicing Surya namaskar | 60                  |
| 2.6               | 2 & 4                                 | W 6           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.1               | 3 & 4                                 | W 7           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.2               | 3 & 4                                 | W 7           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.3               | 3 & 4                                 | W 8           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.4               | 3 & 4                                 | W 8           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.5               | 3 & 4                                 | W 9           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.6               | 3 & 4                                 | W 9           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.7               | 3 & 4                                 | W10           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.8               | 3 & 4                                 | W10           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.9               | 3 & 4                                 | W 11          |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.10              | 3 & 4                                 | W11           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.11              | 3 & 4                                 | W12           |                             |  |                          |                             | Practicing Asana          | 60                  |
| 3.12              | 3 & 4                                 | W 12          |                             |  |                          |                             | Practicing Asana          | 60                  |

**Assessment Pattern:**

| Bloom's level | Continuous Internal Examination |        |            | Semester End Examination |
|---------------|---------------------------------|--------|------------|--------------------------|
|               | Test 1                          | Test 2 | Assignment |                          |
| Remember      | -                               | -      | -          | -                        |
| Understand    | ✓                               | ✓      | ✓          | ✓                        |
| Apply         | ✓                               | ✓      | ✓          | ✓                        |

**Suggested Learning Resources:****Text Books:**

1. Yogapravesha in Kannada by Ajitkumar
2. Light on Yoga by BKS Iyengar
3. Teaching Methods for Yogic practices by Dr. M L Gharote & Dr. S K Ganguly
4. Yoga Instructor Course hand book published by SVYASA University, Bengaluru
5. Yoga for Children –step by step – by Yamini Muthanna

**Web links and Video Lectures (e-Resources): Refer links**

1. <https://youtu.be/KB-TY1gd1wE>
2. <https://youtu.be/aa-TG0Wg1Ls>

[/library/view/interactive-data-visualization/9781449340223/](https://library/view/interactive-data-visualization/9781449340223/)