

Vidyavardhaka Sangha ®, Mysuru

**VIDYAVARDHAKA COLLEGE OF ENGINEERING**

**Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi**

**(Approved by AICTE, New Delhi & Government of Karnataka) Accredited by NBA/NAAC with ‘A’ Grade**

**P.B. No.206, Gokulam 3rd Stage, Mysuru -570 002, Karnataka, India**

**Phone+91 821-4276200, 225 Email:** [**principal@vvce.ac.in**](mailto:principal@vvce.ac.in)

**website:**[**www.vvce.ac.in**](http://www.vvce.ac.in/)

**Bachelor of Engineering**

**III and IV Semester Scheme and Syllabus**

**(2022 Scheme)**

**(2023 Batch)**



DEPARTMENT OF

CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scheme of Teaching and Examination for BE (Autonomous Scheme: 2022)** | | | | | | | | | | | | | | | |
| ***III SEMESTER -* CSE (*Artificial Intelligence & Machine Learning)*** | | | | | | | | | | | | | | | |
| **Sl.**  **No.** | | **Course Area** | **Course Code** | **Course Name** | | **Teaching Department** | | **Contact**  **Hours/week** | | | **Examination** | | | | **Credits** |
| **L** | **T** | **P** | **Duration (Hrs.)** | **CIE**  **Marks** | **SEE**  **Marks** | **Total** |
| 1 | | BS | BITMA301\* | Mathematics -III for IT Stream | | Mathematics | | 2 | 2 | 0 | 3 | 50 | 50 | 100 | 3 |
| 2 | | IPCC | BCIAJ302 | Advanced Java | | CI | | 3 | 0 | 2 | 3 | 50 | 50 | 100 | 4 |
| 3 | | IPCC | BCIDS303\* | Data Structures and its Applications | | CI | | 3 | 0 | 2 | 3 | 50 | 50 | 100 | 4 |
| 4 | | PCC | BCICO304 | Computer Organization | | CI | | 3 | 0 | 0 | 3 | 50 | 50 | 100 | 3 |
| 5 | | HS | BGPEK305 | General Proficiency Enhancement Course – I | | Math | | 0 | 2 | 0 | 2 | 50 | 50 | 100 | 1 |
| 6 | | HS | BUHVK306\* | Universal Human Values | | CI | | 2 | 0 | 0 | 2 | 50 | 50 | 100 | 2 |
| 7 | | ESC/  ETC/PLC | BCIAW317\*  /BCIPM317 | Professional Elective- I | | CI | | 0 | 0 | 4 | 2 | 50 | 50 | 100 | 2 |
| 8 | | AEC | BITTP308 | Technical Proficiency Enhancement Course- I | | CI/TAP | | 0 | 0 | 2 | 2 | 50 | 50 | 100 | 1 |
| **TOTAL** | | | | | | | | **13** | **4** | **10** | **-** | **400** | **400** | **800** | **20** |
| 9 | | Schedu led activities for III to VIII  semest ers | BNSSK309/ BPEDK309/ BYOGK309 | NSS/ PE / Yoga \* | | NSS/ PE /  Yoga | | All students have to register for any one of the courses namely National Service Scheme, Physical Education (PE)(Sports and Athletics), and Yoga with the concerned coordinator of the course during the first week of III semester. The activities shall be carried out between III semester to VIII semester (for 5 semesters). Successful completion of the registered course is mandatory for the award of the degree. The events shall be appropriately  scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities | | | | | | | |
|  |  | | | | **PC:** Professional Core | | **CREDIT Definition** | | | | | | | |  |
| **BS:** Basic Science | | | | **PE :** Professional Electives | | 1 hour Lecture per week per semester = 1 credit | | | | | | | |
| **ES:** Engineering Science | | | | **OE :** Open Electives | | 2 hours Tutorials per week per semester = 1 credit | | | | | | | |
| **HS:** Humanities & Social Science and management course | | | | **PRI:** Project & Internship | | 2 hours Practical / lab / drawing per week per semester = 1 credit | | | | | | | |

|  |  |
| --- | --- |
| **Professional Elective- I** | **Technical Proficiency Enhancement Course- I** |
| BCIAW317\* - Advanced Web Programming laboratory | BITTP308 - Technical Proficiency Enhancement Course- I |
| BCIPM317 - Python Programming for Machine Learning Laboratory |  |

\*common to (AI&ML/CSE/ISE), #common to (CSE/ISE), $common to (AI&ML/CSE), @common to (AI&ML/ISE)

**\*Non–credit mandatory courses (NCMC):**

1. **National Service Scheme/Physical Education (Sport and Athletics)/ Yoga:**
   1. Securing 40 % or more in CIE, 35 % or more marks in SEE and 40 % or more in the sum of CIE + SEE leads to successful completion of the registered course.
   2. In case, students fail to secure 35 % marks in SEE, they have to appear for SEE during the subsequent examinations conducted by the University.
   3. In case, any student fails to register for NSS, PE or Yoga/fails to secure the minimum 40 % of the prescribed CIE marks, he/she shall be deemed to have not completed the requirements of the course. In such a case, the student has to fulfill the course requirements during subsequent semester/s to earn the qualifying CIE marks.
   4. Successful completion of the course shall be indicated as satisfactory in the grade card. Non-completion of the course shall be indicated as Unsatisfactory.
   5. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the courses shall be mandatory for the award of degree.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scheme of Teaching and Examination for BE (Autonomous Scheme: 2022)** | | | | | | | | | | | | |
| ***IV SEMESTER -* CSE (*Artificial Intelligence & Machine Learning)*** | | | | | | | | | | | | |
| **Sl. No** | **Course Area** | **Course Code** | **Course Name** | **Teaching Department** | **Contact**  **Hours/ week** | | | **Examination** | | | | **Credits** |
| **L** | **T** | **P** | **Duration (Hrs.)** | **CIE**  **Marks** | **SEE**  **Marks** | **Total** |
| 1 | BS / PCC | BITMA401\* | Mathematics - IV for IT Stream | Mathematics  /CI | 2 | 2 | 0 | 3 | 50 | 50 | 100 | 3 |
| 2 | IPCC | BCIDA402\* | Design and Analysis of Algorithms | CI | 3 | 0 | 2 | 3 | 50 | 50 | 100 | 4 |
| 3 | IPCC | BCIDB403\* | Database Management System | CI | 3 | 0 | 2 | 3 | 50 | 50 | 100 | 4 |
| 4 | PCC | BCIPA404 | Principles of Artificial Intelligence | CI | 3 | 0 | 0 | 3 | 50 | 50 | 100 | 3 |
| 5 | HS | BGPEK405 | General Proficiency Enhancement Course– II | Math | 0 | 2 | 0 | 2 | 50 | 50 | 100 | 1 |
| 6 | AEC | BITBE406 | Biology for IT Engineers | CI | 2 | 0 | 0 | 2 | 50 | 50 | 100 | 2 |
| 7 | ESC/ETC  /PLC | BCIAW417\*/  BCIPM417 | Professional Elective- II | CI | 0 | 0 | 4 | 2 | 50 | 50 | 100 | 2 |
| 8 | AEC | BITTP408 | Technical Proficiency Enhancement Course-II | CI/TAP | 0 | 0 | 2 | 2 | 50 | 50 | 100 | 1 |
| **TOTAL** | | | | | **13** | **4** | **10** | **-** | **400** | **400** | **800** | **20** |
| 9 | Scheduled activities for IIIto VIII  semesters | BNSSK409 / BPEDK409 / BYOGK409 | NSS/ PE / Yoga \* | NSS/ PE /  Yoga | All students have to register for any one of the courses namely National Service Scheme, Physical Education (PE)(Sports and Athletics), and Yoga with the concerned coordinator of the course during the first week of III semester. The activities shallbe carried out between III semester to VIII semester (for 5 semesters). Successful completion of the registered course is mandatory for the award of the degree. The events shall be appropriately  scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities | | | | | | | |

|  |  |
| --- | --- |
| **Professional Elective- II** | **Technical Proficiency Enhancement Course-II** |
| BCIAW417\* - Advanced Web Programming Laboratory | Technical Proficiency Enhancement Course- II |
| BCIPM417 - Python Programming for Machine Learning Laboratory |  |

\*common to (AI&ML/CSE/ISE), #common to (CSE/ISE), $common to (AI&ML/CSE), @common to (AI&ML/ISE)

2023 Batch

III and IV SEMESTER SYLLABUS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SEMESTER – III** | | | | |
| **Course Name** | Mathematics -III for IT Stream | **Course Code** | | BITMA301\* |
| **Number of Lecture Hours /**  **Week** | 02 | **CIE Marks** | | 50 |
| **Number of Tutorial Hours /**  **Week** | 02 | **SEE Marks** | | 50 |
| **Total Number of Lecture Hours**  **+ Tutorial Hours** | 40 + 10 = 50 | **SEE Duration** | | 03 Hrs. |
| **L:T:P** | 2:2:0 | **Credits** | | 03 |
| **Course Prerequisites**  The students should have strong basis of integral calculus, trigonometry, partial derivative and  permutation and combination | | | | |
| **Course Overview:**  **Transforms, Partial Differential Equations and Combinatorics** is a course which provides mathematical techniques in the advanced areas of mathematics like Fourier, Z-transformation, numerical methods, partial differential equations, advanced linear algebra and combinatorics that are of at most relevance tothe Engineering disciplines. The purpose of this course is to provide the skills and knowledge required to  perform mathematical procedures and processes for solution of Engineering problems. | | | | |
| **Course Learning Objectives (CLOs):**  **The objective is to enable the students to apply the knowledge of Mathematics in various fields of Engineering by the following means:**   1. Explain the concept of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equationsand combinatorics applying it appropriately in solving Engineering problems. 2. Explain how to analyze the Engineering problems by making use of the concepts of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations and combinatorics. 3. Explain the usage of modern tools to understand the concepts of Fourier, Z-Transformation, Numerical   Methods, Partial Differential Equations and combinatorics. | | | | |
| **MODULES** | | | **TEACHING**  **HOURS** | |
| **Module 1: Fourier Series and Harmonic Analysis**  **Fourier Series**: Periodic functions, Dirichlet’s condition. Fourier series of periodic functions. Half range Fourier series, deducing some important series.  **Practical Harmonic Analysis:** Harmonic Analysis.  **SLT: Half range harmonic series.** | | | **08** | |

|  |  |
| --- | --- |
| **Module 2: Fourier Transforms and Z –Transforms (I-C)**  **Fourier Transforms:** Fourier transforms, Fourier sine and cosine transforms, Inverse Fourier transforms (direct method).  **Z-Transforms:** Basic definition, problems on Z-Transforms of standard functions (without proof), Damping and shifting rules (Problems only). Inverse Z-Transform and applications to solve difference equations.  **SLT: Derivation Z-Transforms of standard functions.** | **08** |
| **Module 3: Partial Differential Equations & Numerical solution of PDE(I-C)** Formation of PDE's by elimination of arbitrary functions. Solution of non- homogeneousPDE by direct integration, Solution of Linear PDE by Lagrange’s multiplier method.  **Numerical Methods to Solve PDE’s:** Hyperbolic and Laplace equations.  **SLT: Numerical methods to solve Parabolic equation.** | **08** |
| **Module 4**: **Sets, Relations, and Functions**  **Set theory**: Operations and laws of set theory, inclusive and exclusive principles. **Relation and Function**: Relations and properties, posets, functions and properties, special functions, pigeonhole principles.  **SLT: Venn diagram analysis** | **08** |
| **Module 5**: **Combinatorics**  Permutation & combination (with and without repetition), Multinomial expansion, rook’s polynomial, generating functions, recurrence relations using generating function.**SLT: Forbidden analysis using rook’s polynomial.** | **08** |
| **Textbooks** | |
| 1. **B. S. Grewal**, *Higher Engineering Mathematics*, Latest edition, Khanna Publishers. 2. **B. V. Ramana**, *Higher Engineering Mathematics*, Latest edition, Tata Mc. Graw Hill Publications 3. **E. Balaguruswamy**, *Numerical Methods*, Tata-McGraw-Hill Publication Limited 4. **Dr D S Chandrashekaraiah,** Combinatorics and Graph Theory, Prism Books Pvt. | |
| **Reference Book** | |
| 1. Erwin Kreyszig, *Advanced Engineering Mathematics*, Latest edition, Wiley Publications. 2. Peter V. O’Neil, Advanced *Engineering Mathematics*, 7th edition, CENGAGE Learning India Pvt. Ltd.Publishers. 3. V. N. Ghougule, M. T. Puranic, *Numerical Methods and Optimization*, Niralli Publication. 4. M. K. Jain, S. R. K. Iyengar, *Numerical Methods,* sixth ed., New Age, International, Pub. 5. Ralph P. Grimaldi, Discrete and combinatorial Mathematics, Pearsons Educations. | |

|  |  |
| --- | --- |
| **Course Outcomes (COs):**  **Upon completion of the course, students will be able to** | |
| **CO1** | **Understand** the basic concepts of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations and combinatorics (PO-1). |
| **CO2** | **Apply** the concept of Fourier, Z-Transformation, Numerical Methods, Partial Differential Equations and combinatorics to solve the problems arising in Engineering field (PO-1). |
| **CO3** | **Analyze** mathematical problems arising in Engineering, using the concepts of Fourier, Z- Transformation, Numerical Methods, Partial Differential Equations and combinatorics  (PO-2). |
| **CO4** | Use **modern tool** to **solve/visualize** mathematical problems arising in Engineering (PO-5). |

**CO – PO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** |
| **CO1** | 1 |  |  |  |  |  |  |  |  |  |  |
| **CO2** | 2 |  |  |  |  |  |  |  |  |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  |
| **CO4** |  |  |  |  | 2 |  |  |  |  |  |  |
| **Avg.** | **1.5** | **2** |  |  | **2** |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : Advanced Java | **Course Code:** | BCIAJ302 |
| **Number of Lecture Hours /**  **Week** | **:** 03 | **CIE Marks:** | 50 |
| **Number of Tutorial / Practical Hours / Week** | **:** 02 | **SEE Marks:** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | **:** 40+24=64 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | **:** 3:0:2 | **Credits:** | 04 |
| **Course Prerequisites**  Basic knowledge of datatypes, control, looping statements and Oops concepts in Java programminglanguage. | | | |
| **Course Overview**  This course equips students to take advantage of built-in classes in java through collections. Students shall also learn to store data in relational format through Java program. Learn the fundamentals of JDBC and using the different interfaces in the JDBC API. Learn how to use Java servlets in the role of Web application  control. | | | |
| **Course Learning Outcomes (CLO)**  This course will enable students to,   * Understanding the fundamental concepts of Enumerations and Annotations. * Apply the concepts of String classes in Java programs. * Demonstrate the fundamental concepts of String operations. * Design and develop web applications using Java servlets. * Apply database interaction through Java database Connectivity. | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1**  **Multi-Threaded Programming:** Java Thread Model, the main thread, Creating a thread,Creating multiple threads, Using isAlive() and join(), Thread priorities.  **SLT:** Thread priorities  **Textbook1:** Ch.11 | | | **08** |
| **Module 2**  **The collections Framework:** Collections Overview, for each, The Collection Interfaces- list, set, queue; The Collection Classes- arraylist, linkedlist, hashset; Working With Maps,legacy Classes and Interfaces- vector, dictionary, hashtable. **SLT:** Parting Thoughts on Collections  **Textbook 1:** Ch. 18 | | | **08** |

|  |  |
| --- | --- |
| **Module 3**  **String Handling:** The String Constructors, String Length, Special String Operations-String Literals, String Concatenation, String Concatenation with Other Data Types, String Conversion and  toString( ); Character Extraction-charAt( ), getChars( ), getBytes( ) toCharArray(); String Comparison-equals( ) and equalsIgnoreCase( ), startsWith( ) and endsWith( ), equals( ) Versus == , compareTo( ); Modifying a String-substring( ), concat( ), replace( ), trim( ); Data Conversion Using valueOf( ), Changing the Case of Characters Within a String, StringBuffer –StringBuffer Constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ), append( ), insert( ), reverse( ), delete(  ) and deleteCharAt( ), replace( ), substring( ), StringBuilder  **SLT:** Additional String Methods  **Textbook 1:** Chapter 16 | **08** |
| **Module 4**  **Introducing Servlets:** Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses.  **SLT**: ; Using Cookies  **Textbook 1:** Chapter 38 | **08** |
| **Module 5**  **JDBC Objects:** The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overviewof the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing. **SLT:** Metadata  **Textbook 2:** Chapter 6 | **08** |
| **PRACTICAL MODULE**  **PART A**   1. Create a class Book which contains four members: name, author, price, num of\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects. 2. Create a Java program that demonstrates overloading of sum() with 2 integers in argument list, with 3 integers in argument list and with object in argument list. 3. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extendsthe class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape. 4. Develop a Java program through interface that can demonstrate the concept of implementing multiple inheritance. | **24** |

|  |  |
| --- | --- |
| **5.** Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extendsthe base class. In Father class, implement a constructor which takes the age andthrows the exception Wrong Age( ) when the input age=father’s age.  **PART B**   1. Write a program which creates two threads, one thread displaying “Vidyavardhaka College   of Engineering” once every ten seconds and another displaying “CSE(AI&ML)”once every two seconds.   1. Write a Java program to print duplicate characters from the string. 2. Write a Java Program to demonstrate iterating through the set and print the setelements. 3. Write a Java program to compare two array lists, add and remove the elementfrom specified index of Java Array List. 4. Write a Java Program to add element, remove element and check whether aparticular key/value exists in a HashMap. 5. Write a java program to create a priority queue of integers and demonstratethe following operations:    1. Insertion.    2. Fetching the highest priority element without removal.    3. Popping the most prioritized element.    4. Displaying the priority queue after every operation. 6. Write a Java Program in Servlets to get and display value from an HTML page. 7. Write a Java Program to show connectivity with database using JDBC/ODBC driver.   **Open Ended Programs**   1. Write a JAVA Servlet Program to implement dynamic HTML using Servlet (username and Password should be accepted using HTML and displayed using a Servlet). 2. Write a JAVA Servlet Program to Auto Web Page Refresh (Consider a webpage which is displaying Date and time or stock market status. For all such type of pages, you would need to refresh your web page regularly; Java Servlet makes this job easy by providing refresh automatically after a given interval). 3. Write a JAVA Servlet Program to implement and demonstrate get () and Post methods (Using HTTP Servlet Class). 4. Write a JAVA Servlet Program using cookies to remember user preferences. 5. Write a JAVA JSP Program which uses tag to run an applet. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experimental Weightage** | | | | | |
|  | | | | | |
|  | **Type of Experiment** | | **Program -No** | **Weightage** |  |
| Exercise | | 1,2, | 23% |
| Structure Enquiry | | 3,4,5,6 | 45% |
| Demonstration | | 7,8 | 22% |
| Open ended | | -- | 10% |
| **Textbooks**   1. Herbert Schildt, “JAVA the Complete Reference”, 9th Edition, Oracle Press, 2017. 2. Jim Keogh, “J2EE-TheCompleteReference”, McGraw Hill, 2017 | | | | | |
| **Reference Books**   1. E Balaguruswamy, “Programming with Java”, 5th Edition, McGraw Hill Education, 2017 2. R Nageswara Rao, “Core and Advanced Java, Black Book, Dreamtech Press, 2018 | | | | | |
| **COURSE OUTCOMES (COs)**  At the end of the course students will be able to: | | | | | |
| **CO1** | | Understanding the fundamental concepts of synchronization, string methods, enumerations, servlets and JDBC integration (L2). | | | |
| **CO2** | | **Apply** the concepts of threads, collection frameworks, strings, servlets and JDBC in Java programs(L3). | | | |
| **CO3** | | **Analyze** the concepts of operations in creation of threads, collection classes, String class, http request and response and database connectivity using JDBC(L4). | | | |
| **CO4** | | **Design** and develop application to solve the given problem in a team. (Additional CO) | | | |

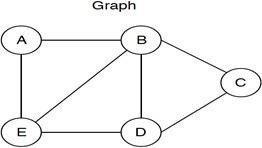
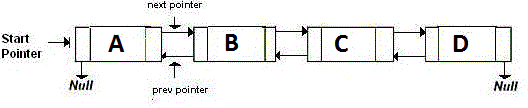
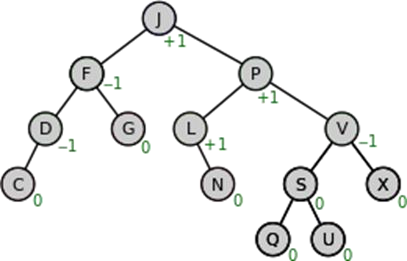
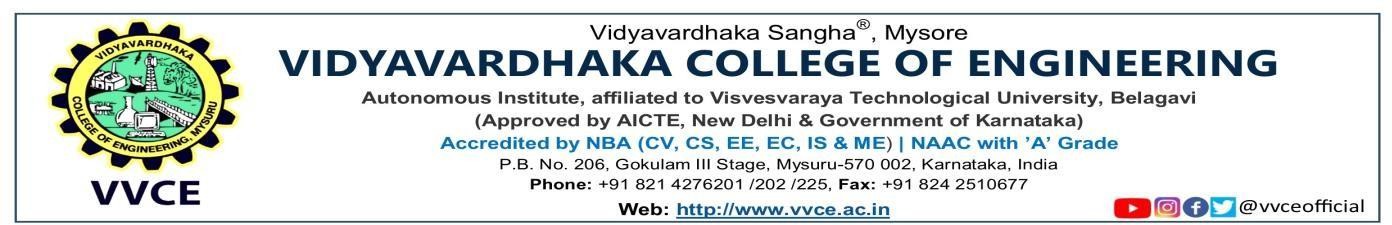
**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO4** |  |  | 2 |  |  |  |  | 2 |  |  |  | 2 |  |  |
| **Avg.** | **2.5** | **2** | **2** |  |  |  |  | **2** |  |  |  | **2.25** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : Data Structures and its  Applications | **Course Code:** | BCIDS303\* |
| **Number of Lecture Hours /**  **Week** | : 03 | **CIE Marks:** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 02 | **SEE Marks:** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 40+24=64 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 3:0:2 | **Credits:** | 04 |
| **Course Prerequisites**  Basic knowledge of C programming language, mathematics, organizing and problem-solving ability. | | | |
| **Course Overview**  This Course focuses on basic algorithms and data structures and learn how to implement them in C. Some of the data structures we will encounter include linked lists, stacks, queues, trees, heaps, hash tables and graphs. Course will also focus on how to analyse algorithms for searching, traversing trees, hashing, manipulating priority queues and finding shortest paths in graphs**.** | | | |
| **Course Learning Outcomes (CLO)**  This course will enable students to,   * Understand the foundations of data structure and how different data structures are used foreffective data access and data manipulation. * Investigate various data structures such as stacks, queues, link lists, trees and graphs. * Understand the context of problem definition and implement a suitable data structure to solve it. | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1**  **Introduction**: Data structures, pointers and dynamic memory allocation, data abstraction.  **Arrays:** Dynamically allocated arrays, Sparse matrix and polynomial representation. **String Processing & Pattern matching Algorithms:** Naive Pattern Searching, KMPAlgorithm.  **Sorting Techniques:** Insertion Sort, Radix Sort  **SLT:** String Operations  **Textbook 1:** Ch.1-1.2,1.4, Ch.2-2.2,2.4,2.5,2.7.3  **Textbook 2:** Ch.1-1.4, Ch.3-3.5, 3.7, Ch. 9-9.3,9.7 | | | **08** |
| **Module 2**  **Stacks:** Introduction, Array representation, Applications of stacks: Infix to Postfix, Evaluation of Postfix, Recursion, Tower of Hanoi**.**  **Queues:** Introduction, Circular queues, Deques, Priority queue.  **SLT:** Multiple stacks and queues  **Textbook 1:** Ch. 3- 3.4,3.6,3.7  **Textbook 2:** Ch. 6-6.1 to 6.3,6.5, 6.7, 6.8, 6.10, 6.12,6.13 | | | **08** |

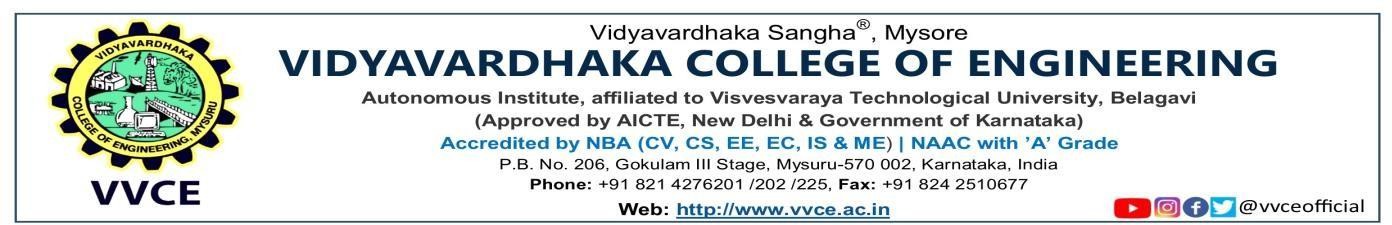
|  |  |
| --- | --- |
| **Module 3**  **Linked Lists:** Introduction, Representation of linked list in memory, traversing and searching linked list, insertion, deletion from the linked list, header linked list, two- way linked list**,** Linked list representation of stacks and queues, Circular linked list **SLT:** Addition and concatenation of two lists  **Textbook 1:** Ch.4 - 4.5  **Textbook 2:** Ch. 5 - 5.1 to 5.5, 5.7 to 5.10, Ch.6 – 6.4, 6.11 | **08** |
| **Module 4**  **Trees:** Introduction, Binary Trees, Binary tree Traversal, Additional binary tree operations,Threaded Binary trees, Binary search Trees  **SLT**: Expression trees, Heaps  **Textbook 1:** Ch. 5- 5.1 to 5.7 | **08** |
| **Module 5**  **Trees:** Selection Trees, Forest, Counting binary tree **Efficient Binary Search tree**: AVL Tree, Red Black tree **Graphs:** ADT, Elementary graph operations:  BFS, DFS **Hashing**: Introduction, Static Hashing, Dynamic Hashing.**SLT:** Optimal Binary Search trees  **Textbook 1:** Ch. 5 - 5.8, 5.9, 5.11, Ch.6 – 6.1,6.2.1,6.2.2, Ch.8- 8.1 to 8.3, Ch.10 - 10.1,  10.2, 10.3.1, 10.3.2, 10.3.4 | **08** |
| **PRACTICAL MODULE**  **Exercise Experiments:**   1. Design, Develop and Implement a Program in C for the following operations on Strings:    1. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)    2. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR.    3. Pattern Matching Algorithm: Brute Force / KMP    4. Support the program with functions for each of the above operations. Don't use Built-in functions.    5. Check the following test cases.   **Test Case 1:** STR = “VVCE MYSURU”, PAT=” MYSURU”, REP=” KARNATAKA”, OUTPUT=”VVCE KARNATAKA”  **Test Case 2:** STR = “COMPUTER SCIENCE”, PAT=” COMPUTER”, REP=” BASIC”,OUTPUT=” BASIC SCIENCE”   1. Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX).    1. Push an Element on to Stack.    2. Pop an Element from Stack.    3. Demonstrate how stack can be used to check palindrome. | **24** |

|  |  |
| --- | --- |
| 1. Demonstrate Overflow and Underflow situations on Stack. 2. Display the status of Stack. 3. Exit 4. Design, Develop and Implement a Program in C for the following operations onexpression.    1. Read infix expression String (INFIX)    2. Convert the infix expression (INFIX) to a postfix expression using stacks.    3. Evaluate the postfix expression using stacks.    4. Check the following test cases.   **Test Case 1:** Infix = “(1+ (2-3) \*4)”, Postfix=”123-4\*+”, Result = -3  **Test Case 2:** Infix = “4/2-2+3\*3-4\*2”, Postfix=”42/233\*42\*-+-”, Result = -1 **Note:** Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, \*, /, % (Remainder), ^ (Power) and alphanumeric operands.  **Demonstration Experiments:**   1. Design, Develop and implement priority queue for job processing.   **Note:**   * The Program should allow users to add or remove items from the queue. * It should also display current status i.e. the total number of items in the queue.  1. Design, Develop and implement c program using singly linked list for the followingscenario    1. There are two linked list A and B containing the following data:   **A: 3,7,10,15,16,09,22,17,32 and B: 16,02,09,13,37,08,10,01,28**   * 1. Create a linked list C that contains only those elements that are common in linked listA and B   2. Create a linked list D which contains all elements of A as well as B ensures that thereis no repetition of elements.  1. Design, Develop and Implement a menu driven Programming in C for the followingoperations on Binary Search Tree (BST) of Integers.    1. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2    2. Traverse the BST in In-order, preorder, post-Order, zigzag order    3. Search the BST for a given element (KEY) and report the appropriate message.    4. Display the height of binary trees.    5. Exit |  |



**Structured Enquiry:**

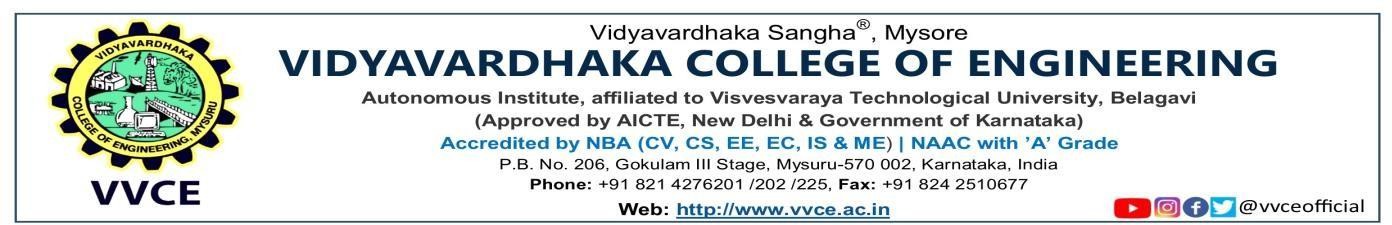
1. Design, Develop and implement C program for the following operations on doublylinked list.
   1. Create doubly linked list of N nodes with integer data by adding each node at the front.
   2. Delete the node of a given data if it is found, otherwise display appropriate message.
   3. Insert a node to the left of the node whose key value is read as input.
   4. Display the contents of the list.
2. Design, Develop a program in C to implement **AVL tree** operations.
3. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F.
   1. Assume that file F is maintained in memory by a Hash Table (HT) of M memory locations with L as the set of memory addresses (2-digit) of locations in HT.
   2. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash Function H: K%L as I (remainder method), and implement hashing techniques to map a given key K to the address space L. Resolve the collision (if any) using linear probing



|  |  |  |
| --- | --- | --- |
| **Type of Experiment** | **Program Numbers** | **Weightage** |
| Demonstration | 1 & 2 | 18% |
| Exercise | 3,4,5 & 10 | 36% |
| Structured Enquiry | 6,7,8,9 | 36% |
| Open Ended |  | 10% |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Open Ended Enquiry:**   * **Design and Develop C program to implement following (Not limited to)**   1. Red Black Tree.   2. Addition of Two polynomial using Linked List.   3. Maze problem.   4. Undo & Redo operation using stack.   5. Dynamic Hashing. * **Design and Develop C program to implement following search algorithms:**   1. Jump Search.   2. Interpolation Search.   3. Exponential Search.   4. Sub list Search (Search a linked list in another list) Fibonacci Search.   5. The Ubiquitous Binary Search.   **Weightages:** | |
| **Textbooks**   1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, reprint 2018 2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014 | |
| **Reference Books**   1. Programming and Data Structure by Jackulin C Salini etal., Ane books publishers, 2019 2. Learning JavaScript data structures and algorithms hone your skills by learning classic data   structures by Loiane Groner, Pack T publishing, 2019 | |
| **Course Outcomes (COs)**  At the end of the course students will be able to: | |
| **CO1** | **Explain** the fundamentals of basic data structures(L2). |
| **CO2** | **Apply** the various data structures to solve the given problem(L3). |
| **CO3** | **Analyze** the applications of various data structures(L4). |
| **CO4** | **Design** appropriate solution by implementing suitable data structure for a given problem  as individual/team**.** (Additional CO) |



**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO4** |  |  | 3 |  |  |  |  | 2 |  |  |  | 3 |  |  |
| **Avg.** | **2.5** | **2** | **3** |  |  |  |  | **2** |  |  |  | **2.5** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : Computer Organization | **Course Code:** | BCICO304 |
| **Number of Lecture Hours /**  **Week** | : 03 | **CIE Marks:** | 50 |
| **Number of Tutorial / Practical Hours / Week** | : 00 | **SEE Marks:** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 40 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 3:0:0 | **Credits:** | 03 |
| **Course Prerequisite**  Basic knowledge of digital logic design, electronic circuits and functional components of computer system are required to learn the subject | | | |
| **Course Overview**  Computer organization course deals with the selection of the basic functional units such as the processor and memory, and how they should be interconnected into a computer system. Computer organization is also concerned with the various hardware components operations along with their interconnection to implement the architectural specifications. | | | |
| **Course Learning Outcomes (CLOs) The students will be able to learn.**   * Understand the basic sub systems of a computer, their organization, structure and operations. * Illustrate the concept of programs as sequences of machine instructions. * Demonstrate different ways of communicating with I/O devices and memory systems. * Analyze arithmetic and logic units. | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1**  **Basic Structure of Computers**: Basic Operational Concepts, Bus Structures, Performance-Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement.  **Machine Instructions and Programs**: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.  **SLT**: Additional instructions  **Textbook : Ch: 1.3-1.4,1.6(1.6.1-1.6.4,1.6.7),2.2-2.5** | | | **08** |
| **Module 2**  **Input/Output Organization:** Accessing I/O Devices, Interrupts-Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces-PCI Bus.  **SLT**: USB  **Textbook: Ch.: 4.1,4.2.1,4.4-4.7.1** | | | **08** |

|  |  |  |
| --- | --- | --- |
| **Module 3 Memory System:**  Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size and Cost, Cache Memories-Mapping Functions, Replacement Algorithm, Performance Considerations.  SLT: Virtual Memory  **Textbook: Ch.:5.1-5.4,5.5(5.5.1,5.5.2),5.6** | | **08** |
| **Module 4**  **Arithmetic:** Addition & subtraction of signed numbers, Design of fast adders; Multiplication of positive numbers, Signed-operand multiplication, Fast multiplication, Integer division  **SLT:** Carry-Save addition of summands  **Textbook : Ch.: 6.1-6.5.1, 6.6** | | **08** |
| **Module 5**  **Basic Processing unit:**  Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization,  Hard-wired Control  **Pipelining:** Basic concepts of pipelining.  **SLT: Data Hazards Textbook : Ch.: 7.1- 7.4,8.1** | | **08** |
| **Textbooks**  1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, Fifth Edition,  McGraw HillEducation (India), 2012. | | |
| **Reference Books**  2. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015 | | |
| **COURSE OUTCOMES (COs)**  At the end of the course students will be able to | | |
| **CO1** | **Explain** the basic structure and machine Instructions of a computer system**.** | |
| **CO2** | **Identify** the functioning of different sub systems such as Input/output devices and  processingunit. | |
| **CO3** | **Analyze** the appropriate design strategies for arithmetic operation and memory. | |
| **CO4** | Ability to independently assemble, disassemble and upgrade the system  configuration by applying the knowledge of Computer Organization **(PO11 - Lifelong Learning)** | |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO4** |  |  |  |  |  |  |  | 2 |  |  | 2 | 2 |  |  |
| **Avg.** | **2.5** | **2** |  |  |  |  |  | **2** |  |  | **2** | **2.25** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : General Proficiency  Enhancement Course – I | **Course Code :** | BGPEK305 |
| **Number of LectureHours /**  **Week** | : 00 | **CCE Marks :** | 50 |
| **Number of Tutorial /Practical Hours / Week** | : 02 |  |  |
|  | **SEE Marks :** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 30 | **SEE Duration :** | 02 Hrs. |
| **L:T:P** | : 0:2:0 | **Credits :** | 01 |
| **Course Prerequisites**: Knowledge of basic mathematics. | | | |
| **Course Overview:**  Aptitude training is designed to enhance an individual's problem-solving, analytical, and critical thinkingskills. It is often a crucial component of career development, competitive exams, and job recruitment processes. | | | |
| **Course Learning Objectives (CLO) :**   1. Develop Problem Solving Skills. 2. Enhance Quantitative Aptitude, Logical Reasoning and Verbal Ability Skills. 3. Prepare students for job recruitment process and competitive exams. | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1:**  **Quantitative Aptitude:** Number System – Divisibility & Remainder, Multiples &Factors, Integers, HCF & LCM, Decimal Fractions, Surds & Indices, Simplification. | | | **06** |
| **Module 2:**  **Quantitative Aptitude:** Percentages, Profit and Loss.  **Logical Reasoning:** Blood Relations, Direction Sense Test. | | | **06** |
| **Module 3:**  **Verbal Ability:** Change of Speech & Voice, Sentence Correction, Sentence Completion, Closet Test. | | | **06** |
| **Module 4:**  **Quantitative Aptitude:** Simple and Compound Interest, Averages.  **Logical Reasoning:** Number & Letter Series, Coding and Decoding, Analogy. | | | **06** |
| **Module 5:**  **Quantitative Aptitude:** Allegations and Mixtures, Ratios, Proportions and Variations,Partnership.  **Logical Reasoning:** Seating Arrangements. | | | **06** |

|  |  |
| --- | --- |
| **Textbooks:**   1. Quantitative Aptitude for Competitive Examinations by R.S Aggarwal 2. A Modern Approach to Verbal & Non-Verbal Reasoning by R.S. Aggarwal | |
| **Course Outcomes (COs):**  **Upon completion of the course, students will be able to** | |
| **CO1** | **Solve** complex problems using logical and structured approaches. |
| **CO2** | **Analyze** complex problems using logical and structured approaches |
| **CO3** | **Develop** critical thinking skills, enabling them to evaluate information and make effective  decisions. |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 2 |  |  |  |  |  |  |  |  |  |
| **CO3** |  |  | 2 |  |  |  |  |  |  |  |  |
| **Avg.** | **2** | **2** | **2** |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : Universal Human  Values | **Course Code :** | BUHVK306\* |
| **No. of Lecture Hours / Week** | : 02 | **CIE Marks :** | 50 |
| **No. of Tutorial / Practical Hours / Week** | : 00 | **SEE Marks :** | 50 |
| **Total No. of Lecture + Tutorial**  **/ Practical Hours** | : 25 | **SEE Duration :** | 02 Hrs. |
| **L:T:P** | : 2:0:0 | **Credits :** | 02 |
| **Course Overview:** The present education system has become largely skill based. The prime emphasis is on science and technology. However, science and technology can only help to provide the means to achieve what is considered valuable in terms of facilities. The course  addresses the deficiencies and inspires to know, explore, verify and practice universal human values and professional ethics for leading a happy and successful life. | | | |
| **Course Learning Objectives (CLOs):**   * Development of a holistic perspective based on self-exploration about themselves (human being), family, society and nature/existence. * Understanding (or developing clarity) of the harmony in the human being, family, society and nature/existence * Strengthening of self-reflection. * Development of commitment and courage to act. | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **Module 1: Course Introduction - Need, Basic Guidelines, Content and Process for Value Education**  Purpose and motivation for the course, recapitulation from Universal Human Values-I, Self-Exploration–what is it? - Its content and process; ‘Natural Acceptance’ and Experiential Validation- as the process for self-exploration, Continuous Happiness and Prosperity- A look at basic Human Aspirations, Right understanding, Relationship and Physical Facility- the basic requirements for fulfillment of aspirations of every human being with their correct priority. Understanding Happiness and Prosperity correctly- A critical appraisal of the  current scenario, Method to fulfill the above human aspirations: understanding and living in harmony at various levels. | | | **05** |
| **Module 2: Understanding Harmony in the Human Being - Harmony in Myself** Understanding human being as a co-existence of the sentient ‘I’ and the material ‘Body’, Understanding the needs of Self (‘I’) and ‘Body’ - happiness and physical facility, Understanding the Body as an instrument of ‘I’ (I being the doer, seer and enjoyer), Understanding the characteristics and activities of ‘I’ and harmony in ‘I’, Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail, Programs to ensure  Sanyam and Health. | | | **05** |

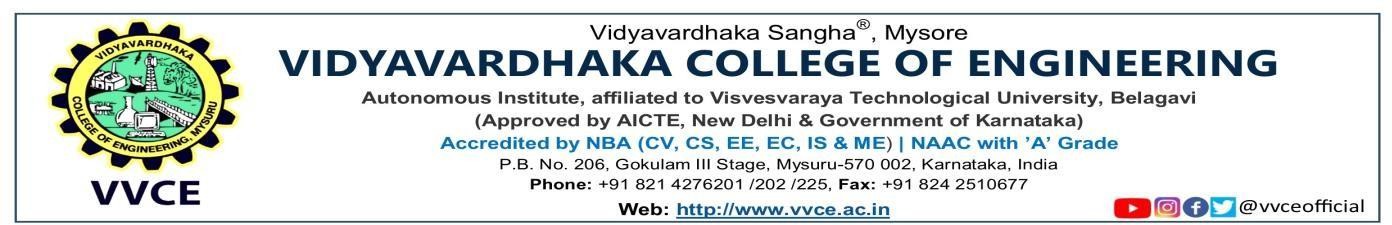
|  |  |
| --- | --- |
| **MODULE 3: Understanding Harmony in the Family and Society- Harmony in Human-Human Relationship**  Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfillment to ensure mutual happiness; Trust and Respect as the foundational values of relationship. Understanding the meaning of Trust; Difference between intention and competence. Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship. Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals. Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family. | **05** |
| **MODULE 4: Understanding Harmony in the Nature and Existence - Whole existence as Coexistence**  Understanding the harmony in the Nature, Interconnectedness and mutual fulfillment among the four orders of nature recyclability and self-regulation in nature. Understanding Existence as Co-existence of mutually interacting units in all pervasive space. Holistic perception of harmony at all levels of existence. Discussion on human being as cause of imbalance in nature, pollution, depletion of resources and role of technology etc. | **05** |
| **MODULE 5: Implications of the Holistic Understanding of Harmony on Professional Ethics**  Natural acceptance of human values, Definitiveness of Ethical Human Conduct. Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order. Competence in professional ethics: a. Ability to utilize the professional competence for augmenting universal human order b. Ability to identify the scope and characteristics of people friendly and eco-friendly production systems, c. Ability to identify and develop appropriate technologies and management patterns for above production systems.  Case studies of typical holistic technologies, management models and production systems.  Strategy for transition from the present state to Universal Human Order:   1. At the level of individual: as socially and ecologically responsible engineers, technologistsand managers. 2. At the level of society: as mutually enriching institutions and organizations. | **05** |
| **Textbooks:**  **1.** Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010. | |

|  |  |
| --- | --- |
| **Reference Books:**   1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999. 2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004. 3. The Story of Stuff (Book). 4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi 5. Small is Beautiful - E. F Schumacher. 6. Slow is Beautiful - Cecile Andrews 7. Economy of Permanence - J C Kumarappa 8. Bharat Mein Angreji Raj - PanditSunderlal 9. Rediscovering India - by Dharampal 10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi 11. India Wins Freedom - Maulana Abdul Kalam Azad 12. Vivekananda - Romain Rolland (English) 13. Gandhi - Romain Rolland (English) | |
| **Course Outcomes (COs):**  At the end of course students will be able to | |
| **CO1** | Understand the need of human values to become more aware of themselves, andtheir  family, society, nature. |
| **CO2** | Identify the responsibilities in life, in handling problems with sustainable solutions  keeping human relationships and nature in mind. |
| **CO3** | Inculcate critical ability and ethics towards human relationship and society. |
| **CO4** | Apply what they have learnt to their own self in different day-to-day scenarios in real  life. |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |
| **CO2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO3** |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |
| **CO4** |  |  |  |  |  |  |  |  |  |  | 2 |  |  |  |
| **Avg.** |  |  |  |  |  | **2** | **2** |  |  |  | **2** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER-III** | | | |
| **Course Name** | : Advanced Web  Programming Laboratory | **Course Code:** | BCIAW317 |
| **Number of Lecture Hours**  **/ Week** | : 00 | **CIE Marks:** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 04 | **SEE Marks:** | 50 |
| **Total Number of Lecture**  **+ Tutorial/Practical Hours** | : 48 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 0:0:4 | **Credits:** | 02 |
| **Course Prerequisites:**  Basic knowledge of programming, HTML, CSS and JavaScript are required to learn the course. | | | |
| **Course Overview:** The focus of this course is on the World Wide Web as a platform for interactive applications, content publishing and social services. This course is about the HTTP communication protocol, the markup languages HTML, the CSS for formatting and transforming  web content, XML, AJAX, Node.js and PHP. | | | |
| **Course Learning Objectives (CLO)**  This course will enable students to,   * Learn the syntax and semantics of web technologies. * Demonstrate the use of built-in functions. * To understand the fundamentals of Nodejs, XML, AJAX and PHP. * Interpret the use of procedural statements like conditional statements, loops and function callsin JavaScript | | | |
| **PART A** | | | |
| 1. Design a HTML program to display two paragraphs with following specifications:   P1: display the heading called “ADVANCED WEB PROGRAMMING” and set the text color to white andbackground color to blue.  P2: A paragraph with the basic elements such as bold, italic, subscript, super script, mark and with ahorizontal line and line break   1. Create a webpage to display a table consisting of a customer detail such as customerID, Name, Experience and Salary. | | | |



* + 1. Create a webpage to display an online shopping website with the list of items available with the concept of ordered and unordered list.
    2. Design a webpage to create a form for a student to enroll for a course which includes the following fields: Name, USN, Gender Subject interested with the drop-down options, email, Contact Number,Address with submit and reset button.
    3. Design HTML webpage with following CSS specifications: Show any two-font family through text, decorated the text through underline, overline and line through, display the head text with shadowin red color.
    4. Develop javascript code to check whether the given number is prime or not.
    5. Write a JavaScript to design a simple calculator to perform the following operations: sum, product,difference, and quotient.
    6. Develop javascript code to find the largest of three numbers.

NOTE: The above programs will be assessed in CIE ONLY.

**PART B**

1. Develop a javascript to sort and accessing the array elements.Input: n=5

Unsorted array elements:

10,1,-5,4,15 Sorted Array: - 5,1, 4, 10, 15

1. Create a class by the name rectangle with 2 attributes length and breadth. Include a parameterized constructor to assign values to data members and a function to calculate area of the rectangle. Demonstrate creation of object of class rectangle and display its area.

Input: length=5, breadth=6 Output:

Area=30

1. Develop a javascript to demonstrate the working of callback and async functions.
2. Develop an arrow function in javascript that checks whether a year is leap year, alert the user with true if the year is leap year and false if year is non leap year. Validate centuries also.

Input: 2000, Output: Leap year Input: 2100, Output: Non Leap year Input: 2004, Output: Leap year Input: 2006, Output: Non Leap year

|  |  |  |
| --- | --- | --- |
| **Type of Experiment** | **Program -No** | **Weightage** |
| Exercise | 5, 7, 8 | 33.3% |
| Structure Enquiry | 1, 2, 4 | 33.3% |
| Demonstration | 3, 6 | 22.2% |
| Open ended |  | 11.1% |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| 1. Develop a javascript that accepts length and breadth of rectangle as parameter of an arrow functions. Call the function using spread and rest operator and alert the user with perimeter of the rectangle. 2. Develop a javascript to demonstrate the usage of optional and default parameters in a function. 3. Create a class by the name box with parameters length, breadth, and height. Create a class boxweight that extends box and include a new parameter weight. Create another class by the name boxcost that extends boxweight and has a parameter by the name shipmentcost. Include constructors in all the classes. Create an object of boxcost and display values of all parameters that represent multilevel inheritance. 4. Cretae a database with studentname and usn. Develop a javascript that accesses this database using get methods of REST API to display the contents of database in webpage. | |
| **Open Ended Experiments**  Students shall solve a problem (either given by the staff or student may come up with their own problem) using the design techniques.   1. Weather forecast website. 2. Javascript Music Player. 3. Hospital Management System. 4. Online Shopping Webpage. | |
| **Experimental Weightage:** | |
| **Course Outcomes:** | |
| **CO1** | **Apply** and demonstrate the usage of various programming constructs |
| **CO2** | **Design** and **Develop** applications to solve problems across various technical and real- world domains |
| **CO3** | **Explore** various programming tools and techniques in a team |

**CO-PO-PSO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** |  |  | 3 |  |  |  |  |  |  |  |  |  | 2 |  |
| **CO3** |  |  |  |  | 2 |  |  | 2 |  |  |  |  | 2 |  |
| **Avg.** | **3** |  | **3** |  | **2** |  |  | **2** |  |  |  | **2** | **2** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | : Python Programming for  Machine Learning Laboratory | **Course Code:** | BCIPM317 |
| **Number of Lecture Hours /**  **Week** | : 00 | **CIE Marks:** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 04 | **SEE Marks:** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 48 | **SEE Duration:** | 03 Hr. |
| **L:T:P** | : 0:0:4 | **CREDITS:** | 02 |
| **Course Prerequisite:**  Basic knowledge of programming like python and C. | | | |
| **Course Overview**  This course provides hands-on coding exercises on programming concepts like classes, custom functionsand libraries such as numpy, pandas and Data Visualization. | | | |
| **Course Learning Objectives (CLO)**  The course should enable the students to,   * Understand and implement the basic constructs of object-oriented programming. * Develop programs for data manipulation and data operations using numpy and pandas. * Perform pattern matching using regular expressions | | | |
| **PRACTICAL MODULE PART-A**   1. Write a python program to check whether the number is prime or not.   *(Concepts Explored : Looping and Conditional Statements )*   1. Write a Python program to calculate the length of a string without using built in function and checkwhether it is a palindrome or not.   *(Concepts Explored : Strings, Slicing)*   1. Write a Python function that accepts a sentence from the user and counts the number of words inthe sentence.   *(Concepts Explored : String splitting, List)*   1. Write a Python program to sum all the numbers in a list.   *(Concepts Explored : List )*   1. Write a Python program to store phone numbers of persons in a dictionary and display the phonenumber of the requested person.   *(Concepts Explored : Dictionary)*   1. Develop a python program to copy the contents of only odd lines from one file to another file.   *(Concepts Explored : File Handling)*   1. Write a python program to check if the number is even or odd using a user defined function.   *(Concepts Explored : User defined function)* | | | |

8. Develop a python program to count occurrence of all characters present in a string.

*(Concepts Explored : Dictionary get() method)*

**NOTE:** The above programs will be assessed in CIE-1 ONLY.

**PART-B**

# A- Demonstration

**A1.** Write a program to build a simple Student Management System using Python which can perform thefollowing operations: Accept, Display, Search, Delete, Update.

**A2. a)** Write a Pandas program to display a summary of the basic information about a specified Data Frame and its data.

**b)** Read the given data “churn.csv” and save it as a data frame called churn\_data. Perform followingoperations on the data frame i) Count total number of duplicate records in the data frame ii) Count the no. of duplicate records in the churn data frame based on the cutomerID column iii) Count number of missing values in each columns iv) Count the total no. of missing values for thevariable TotalCharges v) Average monthly charge paid by a customer for the services he/she has signed up for vi) Display the records having “1@#” under the variable Dependents vii) Replace null values in churn data frame by median value or by max count class category (Demonstration)

*(*[*https://drive.google.com/file/d/1JSYGIIkIZr4jyheDEH0X1\_TMSnQ2CzXc/view*](https://drive.google.com/file/d/1JSYGIIkIZr4jyheDEH0X1_TMSnQ2CzXc/view)

**A3.** Explore the automobile dataset and visualize the i) Distribution of the two and four door cars with respect to the type of fuel they use ii) Distribution of cars of different body styles with respect to the type of fuel they use ii) Total number of each type of body style cars categorized by fuel type iii) Horsepowerof each of the fuel type with reference to the type of drive wheel present in cars. (Demonstration)

*(https://*[*www.kaggle.com/datasets/toramky/automobile-dataset?resource=download)***.**](http://www.kaggle.com/datasets/toramky/automobile-dataset?resource=download))

# B- Exercise

**B1.** a) Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

b) Write a NumPy program to save a given array to a text file and load it.

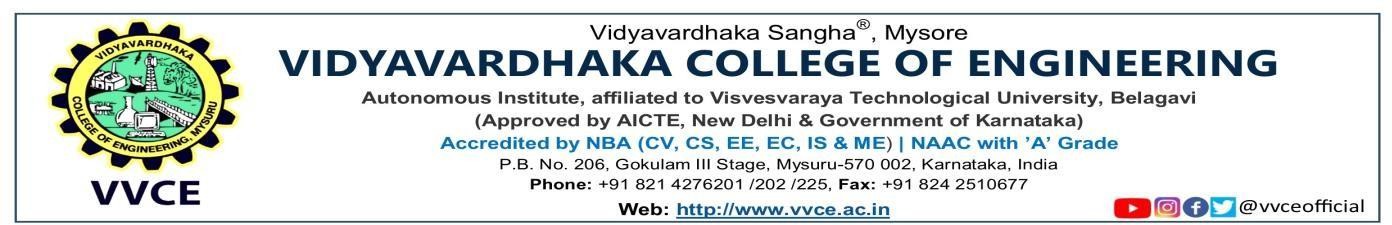
**B2.** Design a Python program using Regular expressions to

1. Extract Email IDs from a given text.
2. Validate the user password with minimum length=6 and maximum length=16 and must have atleast one lower-case letter, upper-case letter, number and special symbol (#, @, $, \_).

**B3.** Write a NumPy program to create a structured array from given student name, height, class and theirdata types. Finally sort the array on height.

# C- Structured Inquiry

**C1.** Develop a pandas program to read CSV file (items.csv, given below) and replace the null values in *quantity* by mean, *price* by median, *bought* by standard deviation, *forenoon* by min and *afternoon* by max.



|  |  |
| --- | --- |
| **C2.** Using the data on births in the United States, provided by the Centers for Disease Control (CDC), Find   1. Total number of US births by year and gender ii) Average daily births by day of week and decade iii) Average daily births by date. (Demonstration)   *(https://raw.githubusercontent.com/jakevdp/data-CDCbirths/ master/births.csv)*  ***C3.*** Visualize and understand finishing results from a marathon race with respect to distribution of split fractions, distribution of split fractions by gender, distribution of split fractions by gender and age. (Exercise)  *(*[*https://raw.githubusercontent.com/jakevdp/marathon-data/master/marathon-data.csv*](https://raw.githubusercontent.com/jakevdp/marathon-data/master/marathon-data.csv)*)*  ***D- Open Ended Experiments***  The student can choose to solve any one open-ended problem to illustrate python application in thedomains specified below (but not restricted to) using various python packages.   * 1. Excel file handling   2. PDF/word file manipulation 3.CSV file analysis using pandas.   4.Simple Games 5.Demonstrate polymorphism | |
| **Course Outcomes (COs)**  At the end of the course students will be able to | |
| **CO1** | **Apply** the concepts of object-oriented programming for modelling and solving real  world problems |
| **CO2** | **Demonstrate** the use of regular expression for pattern matching |
| **CO3** | **Apply** various python packages for data exploration and manipulation techniques  using suitable tools. |

**CO – PO – PSO Matrix**

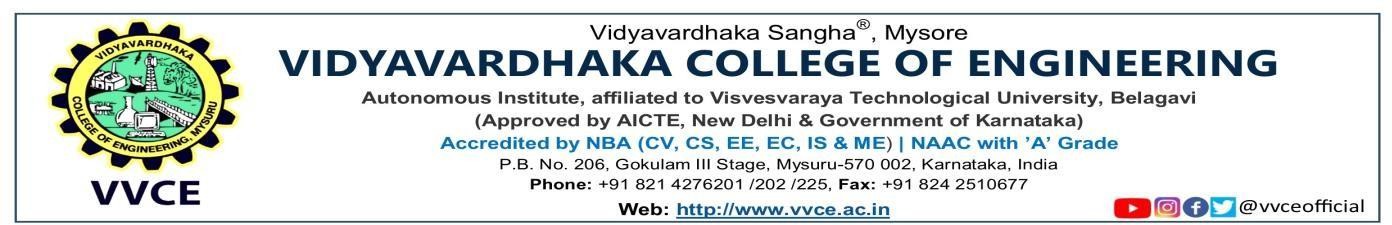
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** |  |  | 3 |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO3** |  |  |  |  | 2 |  |  | 2 |  |  |  | 2 |  |  |
| **Avg.** | **3** |  | **3** |  | 2 |  |  | **2** |  |  |  | **2** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SEMESTER – III** | | | | |
| **Course Name** | **:** Technical Proficiency  Enhancement Course- I | **Course Code:** | | BITTP308 |
| **Number of Lecture Hours / Week** | **:** 00 | **CIE Marks:** | | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | **:** 02 | **SEE Marks:** | | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | **:** 30 | **SEE Duration:** | | 02 Hrs. |
| **L: T: P** | **:** 0:0:2 | **Credits:** | | 01 |
| **Course Prerequisites:** Working knowledge of C Programming. | | | | |
| **Course Overview:** This course is designed to enhance an individual’s problem-solving, analytical, and  critical thinking skills using C/C++ programming language. It also helps the students to acquire skills for career development, competitive exams, and job recruitment processes. | | | | |
| **Course Learning Objectives (CLO)**  This course will enable students to,   * To enhance problem-solving skills. * To develop programming skills using functions, strings, and Arrays. * To Prepare students for job recruitment process and competitive exams. | | | | |
| **MODULES** | | | **TEACHING**  **HOURS** | |
| **Module 1: Introduction to Programming:**  **Flowcharts:** Introduction to flowcharts, Decision making using flowcharts, Loops, Example problems.  **Variables and Data types:** First program, Variables and data types, Taking input, How data is stored in memory, Arithmetic Operators.  **Conditional statements:** Introduction to If else, Relational and logical operators, Nested conditionals. | | | **06** | |
| **Module 2: Loops and Functions 1**  **While Loops:** While loops, Flow of execution of statements in while loop, Example problems using while loop.  **Patterns:** Introduction to patterns, Basic Patterns, Square Patterns, Triangular Patterns, Character Patterns, Reverse Triangle, Inverted patterns, Isosceles triangles. | | | **06** | |
| **Module 3: Loops and Functions 2**  **For Loops:** For loops, Break and Continue, increment - decrement operators. **Functions:** Introduction to functions, Working of function calling, Variables and its scope, Pass by value. | | | **06** | |

|  |  |  |
| --- | --- | --- |
| **Module 4: Arrays**  **Introduction to Arrays:** Introduction to arrays, How arrays are stored in memory,Passing arrays to functions.  **Searching and Sorting:** Understanding Binary Search, Selection sort, Bubble sort,  Insertion sort, Merging two sorted arrays. | | **06** |
| **Module 5: Strings and 2D Arrays**  **Strings:** Introduction to strings, storage of strings and their inbuilt functions  **2D Arrays:** 2D arrays, Storage of 2D arrays, Example problems using 2D Arrays | | **06** |
| **Reference Books:**   1. The C Programming Language (2nd edition) by Brian Kernighan and Dennis Ritchie 2. C in Depth by S K Srivastava and Deepali Srivastava | | |
| **Course Outcomes (COs):**  **Upon completion of the course, students will be able to** | | |
| **CO1** | **Apply** and solve complex problems efficiently using logical and structured approaches | |
| **CO2** | **Develop** critical thinking skills, enabling them for job interviews and assessment tests | |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| **Avg.** | **3** | **3** |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name:** | Mathematics - IV for IT  Stream | **Course Code** | BITMA401\* |
| **Number of Lecture Hours /Week** | 02 | **CIE Marks** | 50 |
| **Number of Tutorial /**  **Hours /Week** | 02 | **SEE Marks** | 50 |
| **Total Number of Lecture**  **Hours + Practical Hours** | 40+ 10 = 50 | **SEE Duration** | 03 Hr. |
| **L:T:P :** | 3:2:0 | **Credits** | 03 |
| **Course Prerequisite**  Student should have knowledge on basics of stats, probability and matrix operations | | | |
| **Course Overview:**  Advanced linear algebra, Probability, statistics, and optimization is a course which provides mathematical techniques in the advanced areas of mathematics that are of the most relevance to engineering disciplines. The major focus of the course is statistical methods, probability distributions, sampling Theory, statistics, optimization. The purpose of this course is to provide the skills and knowledge required to perform mathematical procedures and processes for the solution of engineering  problems. The course aims to show the relevance of mathematics to engineering and applied science. | | | |
| **Course Learning Objectives (CLO)**  **The objective is to enable the students to apply the knowledge of mathematics in various fields of Engineering by the following means:**   1. Explain the concept of statistical methods, probability distributions, sampling Theory, statistics,optimization.in solving Engineering problems. 2. Explain how to apply statistical methods, probability distributions, sampling Theory, statistics,optimization. 3. Explain how to analyze the concept of the statistical methods, probability distributions, sampling   Theory, statistics, optimization. | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **Module 1: Statistical Methods**  **Correlation and Regression**-Karl Pearson’s coefficient of correlation and rank correlation (without repetitions) -Problems. Regression analysis- lines of regression – Problems.  **Curve Fitting**: Curve fitting by the method of least squares- fitting linear and nonlinear curves (y= 𝑎𝑥2 + 𝑏𝑥 + 𝑐 and 𝑦 = 𝑎𝑥𝑏).  **SLT: Fitting the curve** 𝒚 = 𝒂𝒙 + 𝒃 𝒂𝒏𝒅 𝒚 = 𝒂𝒃𝒙. | | | **08** |
| **Module 2: Advanced Linear Algebra:**  Orthogonalization (GS algorithm), QR-factorization, Singular value decomposition (SVD) (without proof).  SLT: Singular value decomposition of square matrix (2 x 2) | | | **08** |

|  |  |
| --- | --- |
| **Module 3: Probability Distribution and Joint Probability Distribution(I-C) Probability Distribution:** Review of basic probability theory. Random variables (discrete),Probability mass function, Statistical values (Mean and Variance). Poisson and Normal distributions- Problems (No derivation for mean and standard deviation).  **Joint Probability Distribution**: Joint probability distribution for two discrete random variables, Marginal distribution, Covariance, and Coefficient of correlation.  **SLT: Problems on exponential distribution.** | **08** |
| **Module 4: Stochastic Process and Sampling Theory (I-C)**  **Stochastic Theory:** Stochastic process, Markov chain, Transition matrix, Steady state analysis.  **Sampling Theory:** Introduction to sampling distributions, Standard error, Type-I and Type-II errors. Test of hypothesis, Z-test (proportions -one &two samples ), Chi-squaredistribution as a test of goodness of fit (Uniform, Poisson’s distribution, ratio and proportion).  **SLT: Student’s t-distribution (both one and two samples).** | **08** |
| **Module 5**  **Optimization Technique:** Basics of optimization, formulation of the problems, maximaand minima, convex function, global solution. Linear programming, simplex algorithm, degeneracy, Big-M method, two phase method.  **SLT**: Duality of LPP (no solving) | **08** |
| **Textbooks** | |
| 1. **B. S. Grewal**, *Higher Engineering Mathematics*, 43 Edition, Khanna Publishers. 2. **B. V. Ramana**, *Higher Engineering Mathematics*, Latest edition, Tata Mc. Graw Hill Publications. 3. **G. Haribhaskaran,** *Probability, Queueing Theory & Reliability Engineering***,** Lakshmi Publication 4. **Taha Hamdy A**, Operational Research, 6th edition, Prentice Hall of India. | |
| **Reference Book** | |
| 1. **Erwin Kreyszig**, *Advanced Engineering Mathematics*, Latest edition, Wiley Publications. 2. **Peter V. O’Neil**, Advanced *Engineering Mathematics*, 7th edition CENGAGE Learning India Pvt. Ltd.Publishers. 3. **A M Natarajan**, Probability, Random Processes and Queueing Theory*,* New Age International, 2005 | |

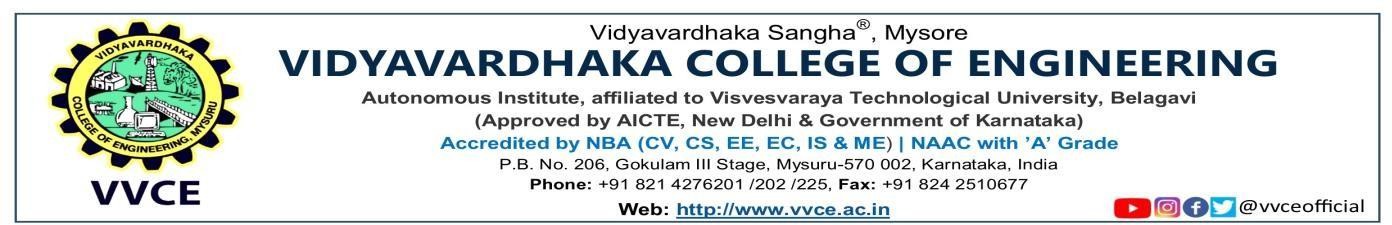
|  |  |
| --- | --- |
| 1. **Kishore S Trivedi**, *Probability and Statistics with Reliability, Queuing, and Computer ScienceApplications,* 2nd Edition, Kindle Edition 2. ***V. Sundarapandian****, Probability, Statistics and Queueing Theory, Sultan Chand & Sons Pub,* 3. **S. D. Sharma**, Operations research, Kedarnath and Ramanath and Com. | |
| **Course Outcomes (COs):** | |
| **CO1** | **Understand** the basic concepts of Complex Analysis, Statistical Methods, Probability Distributions, Sampling Theory, Statistics, Queueing Theory and Reliability Engineering /  Optimization (PO-1) |
| **CO2** | **Apply** the concept of Complex Analysis, Statistical Methods, Probability Distributions,  Sampling Theory, Statistics, Queueing Theory and Reliability Engineering / Optimization to solve the problems arising in Engineering field. (PO-1) |
| **CO3** | **Analyze** mathematical problems arising in Engineering using the  Complex Analysis, Statistical Methods, Probability Distributions, Sampling Theory,  Statistics, Queueing Theory and Reliability Engineering / Optimization (PO-2). |
| **C04** | Use **modern tools** to **solve/visualize** mathematical problems arising in Engineering (PO-5). |

**CO – PO – PSO Matrix:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO4** |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |
| **Avg.** | **1.5** | **2** |  |  | **2** |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name** | : Design and Analysis of Algorithms | **Course Code:** | BCIDA402\* |
| **No. of Lecture Hours / Week** | **:** 03 | **CIE Marks:** | 50 |
| **No. of Tutorial / Practical Hours /**  **Week** | **:** 02 | **SEE Marks:** | 50 |
| **Total No. of Lecture + Tutorial /**  **Practical Hours** | **:** 40+24=64 | **SEE Duration:** | 03 Hrs. |
| **L: T: P** | **:** 3:0:2 | **Credits:** | 04 |
| **Course Prerequisite**  Basic knowledge of Data Structures, C programming language, java and mathematics problem-solving ability. | | | |
| **Course Overview**  Design and Analysis of Algorithms is very important for designing algorithms to solve different types of problems in the branch of Computer Science and Information Technology. As a fundamental subject, this course equips the students with theory and hands on with respect to mathematical  analysis of algorithms and problem-solving skills. | | | |
| **Course Learning Objectives (CLO)**  This course will enable students to,   * Describe various methods of algorithm analysis. * Explain various computational problem-solving techniques. * Analyse the computational complexity of different algorithms. * Design algorithms using various strategies. | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **Module 1**  **Introduction:** What is an Algorithm? Fundamentals of algorithmic problem solving, The Analysis Framework, Asymptotic notations and Basic efficiency classes, Mathematical analysis of Non recursive algorithms, Mathematical analysis of recursive algorithms.  **SLT:** Important problem types.  **Textbook1:** Ch. 1.1,1.2,1.3, 2.1,2.2,2.3,2.4 | | | **08** |
| **Module 2**  **Divide and Conquer**: Recurrence equation for divide and conquer, Binary search, Merge sort, Quicksort, Strassen’s Matrix Multiplication.  **SLT:** Topological Sorting.  **Textbook1:** Ch.4.2,5.1,5.2, 5.4  **Textbook2:** Ch.3.3 | | | **08** |
| **Module 3**  **Greedy Technique:** Prim’s Algorithm, Kruskal’s Algorithm, Dijkstra's Algorithm. Knapsack Problem, Job Sequencing with deadlines, Huffman Trees  **SLT:** Heap sort  **Textbook 1**: Chapter 9(Section 9.1,9.2,9.3,9.4), Chapter 6( section 6.4)  **Textbook 2:** Ch. 4.3,4.5 | | | **08** |

|  |  |
| --- | --- |
| **Module 4**  **Dynamic Programming:** Multistage Graph**,** Floyd's Algorithm, The Knapsack problem and memory functions, The Travelling Salesperson problem.  **SLT:** Warshall’s Algorithm  **Textbook 1:** Ch. 8.2,8.4  **Textbook 2:** 5.3,5.4,5.9 | **08** |
| **Module 5**  **Backtracking, Branch and Bound solution:** Backtracking, N-Queens Problem, Sum of subsets problem, Assignment problem, The Travelling Salesperson problem, 0/1 Knapsack Problem  **NP-Complete and NP-Hard problems:** Basic concepts, non- deterministic algorithms, P, NP, NP Complete, and NP-Hard classes  **SLT:** Graph Colouring.  **Textbook1:** Ch 11.3, 12.1, 12.2  **Textbook 2**: Ch 7.4 | **08** |
| **PART B** | |
| 1. Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and 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. Implement a merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and 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. 3. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm. 4. Implement Warshall's algorithm to determine the transitive closure of a directed graph or all paths in a directed graph by using the adjacency matrix. 5. From a given source vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra’s algorithm. 6. Implement 0/1 Knapsack problem using Dynamic Programming. 7. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm. 8. Implement a subset concept for a given set S = {sl, s2, ,sn} of n positive integers whose sum is   equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions  {1, 2, 6} and {1,8}. A suitable message is to be displayed if the given problem instance doesn't have a solution. | |
| **Textbooks**   1. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 3rd Edition, 2017, Pearson. 2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2017, Universities Press. | |



|  |  |
| --- | --- |
| **Reference Books**   1. Algorithm Design, John Kleinberg, Eva Tardos, 1st Edition, 2013, Pearson. 2. Algorithms, S. Dasgupta, C.H. Papadimitrou and U. Vazirani, Indian Edition ,2017, McGraw-Hill Education. | |
| **Course Outcomes (COs)**  At the end of the course, students will be able to | |
| **CO1** | **Describe** the computational solutions to well-known problems. |
| **CO2** | **Apply** the appropriate design techniques for problem solving. |
| **CO3** | **Analyze** the computational complexity of various algorithms. |
| **CO4** | **Design and demonstrate** the solution for a given problem using appropriate algorithm. |

**CO – PO – PSO Matrix:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO4** |  |  | 3 |  |  |  |  |  | 3 |  |  | 3 |  |  |
| **Avg.** | **2.5** | **2** | **3** |  |  |  |  |  | **3** |  |  | **2.5** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name** | : Database Management  System | **Course Code:** | BCIDB403\* |
| **Number of LectureHours /**  **Week** | : 03 | **CIE Marks:** | 50 |
| **Number of Tutorial**  **/ Practical Hours /Week** | : 02 | **SEE Marks:** | 50 |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 40+24=64 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 3:0:2 | **Credits:** | 04 |
| **Course Prerequisite**  A Good understanding of basic computer concepts and set theory. | | | |
| **Course Overview**  The course provides fundamental understanding of database management systems, query retrieval, normalization, and transaction management. | | | |
| **Course Learning Objectives (CLOs)**  This course will enable students to,   * Understand basic programming concepts Identify and define the information that is needed todesigna database management system for a given problem. * Understand the core terms, concepts, and tools of relational database management systems. * Build a database management system that satisfies relational theory and provides users withbusinessqueries, forms, and reports. * Learning the language for creating and manipulating the database using. SQL. * Understanding the concept of Transaction Management and concurrency control. | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **Module 1**  **Databases and Database users:** Introduction, Characteristics of database approach, Advantages of using the DBMS approach, **Database system concepts:** Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment, **Data Modeling using Entities and RelationshipsModel:** using high-level conceptual data models for Entity types, Entity sets, attributes and keys, Relationship Types, Relationship sets, roles, and structural constraints, Weak entity types,ER diagrams examples.  **SLT:** ER Diagrams, Naming Conventions and Design Issues  **Textbook-1: Ch 1.1- 1.3,1.6, 2.1 - 2.4, 3.1 - 3.5,3.7** | | | **08** |

|  |  |
| --- | --- |
| **Module 2**  **Relational Data Model and Relational Database Constraints:** Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations, **Relational Algebra:** Unary relational operations, Relational Algebra operations from set theory, Binary relational operations, Relational Database Design using ER-to-Relational mapping.  **SLT:** Examples of queries in relational algebra  **Textbook-1: Ch 5.1 - 5.3, 8.1 - 8.5,9.1** | **08** |
| **Module 3**  **Basic SQL:** SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, **More SQL:** More complex SQL retrievalqueries, Specifying constraints as assertions and action triggers, Views in SQL, Schema changestatements in SQL  **SLT:** Additional features of SQL  **Textbook-1: Ch 6.1-6.5,7.1 - 7.4** | **08** |
| **Module 4**  **Database design:** Informal Design Guidelines for Relation Schemas; Functional Dependencies; General Definitions of Second and Third Normal Forms; Boyce Codd Normal Form. Further Topics in Functional Dependencies: Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema  **SLT:** Normal Forms Based on Primary Keys  **Textbook-1: Ch 14.1 - 14.5,15.1 - 15.3** | **08** |
| **Module 5**  **Transaction Processing:** Introduction to Transaction Processing, Transaction states and additional operations, Desirable properties of Transactions, characterizing schedules based on Serializability, **Concurrency Control in Databases:** Two-Phase Locking Techniques for Concurrency Control  **SLT:** Guaranteeing Serializability by Two-Phase Locking  **Textbook-1: Ch 20.1, 20.2.1, 20.3, 20.5.1,20.5.2,21.1.1, 21.1.2** | **08** |
| **PRACTICAL MODULE**   1. Consider the following schema for a Library Database:   **Demonstration**  BOOK (BOOK\_ID, TITLE, PUBLISHER\_NAME, PUB\_YEAR) BOOK\_AUTHORS (BOOK\_ID, AUTHOR\_NAME) PUBLISHER (NAME, ADDRESS, PHONE)  BOOK\_COPIES (BOOK\_ID, BRANCH\_ID, NO\_OF\_COPIES)  BOOK\_LENDING (BOOK\_ID, BRANCH\_ID, CARD\_NO, DATE\_OUT, DUE\_DATE) LIBRARY\_BRANCH (BRANCH\_ID, BRANCH\_NAME, ADDRESS)  CARD (CARD\_NO)  Write SQL queries to   * 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch. | **10** |

|  |  |
| --- | --- |
| 1. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017. 2. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query. 3. Create a view of all books and its number of copies that are currently available in the library. 4. Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation. 5. Consider the following schema for Order Database:   **Demonstration**  SALESMAN (SALESMAN\_ID, NAME, CITY, COMMISSION)  CUSTOMER (CUSTOMER\_ID, CUST\_NAME, CITY, GRADE, SALESMAN\_ID)  ORDERS (ORD\_NO, PURCHASE\_AMT, ORD\_DATE, CUSTOMER\_ID, SALESMAN\_ID)  Write SQL queries to   * 1. Count the customers with grades above Bangalore’s average.   2. Find the name and numbers of all salesmen who had more than one customer.   3. List all the salesman and indicate those who have and do not have customers in their cities (Use UNION operation.)   4. Create a view that finds the salesman who has the customer with the highest order of a day.   5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.  1. Consider the following schema for Company Database**: Exercise**   EMPLOYEE(SSN, NAME, ADDRESS, GENDER, SALARY, SUPERSSN, DNO) DEPARTMENT(DNO, DNAME, MGRSSN, MGRSTARTDATE) DLOCATION(DNO,DLOC)  PROJECT(PNO, PNAME, PLOCATION, DNO) WORKS\_ON(SSN, PNO, HOURS)  Write SQL queries to   * 1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that Controls the project.   2. Show the resulting salaries if every employee working on the ‘IoT’ project is Given a 10 percent raise.   3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department   4. Retrieve the name of each employee who works on the entire projects controlled by department number 5. |  |

e) For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

1. Consider the following schema for Airline flight information:

**Exercise**

FLIGHTS (FLNO, SOURCE, DESTINATION, DISTANCE, DEPARTS, ARRIVES, PRICE)AIRCRAFT (AID, ANAME, CRUISINGRANGE)

CERTIFIED (EID, AID)

EMPLOYEES (EID, ENAME, SALARY)

Note that the Employees relation describes pilots and other kinds of employees aswell; every pilot is certified for some aircraft, and only pilots are certified to fly.

Write SQL queries to

* 1. Find the names of aircraft such that all pilots certified to operate them earn more than $80,000.
  2. For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
  3. For all aircraft with cruisingrange over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
  4. Print the enames of pilots who can operate planes with cruising range greater than 3000 miles but are not certified on any Boeing aircraft.
  5. Print the name and salary of every nonpilot whose salary is more than the average salary for pilots

1. Consider the following schema for College database:

**Structured Enquiry**

STUDENT(SNUM, SNAME, MAJOR, LEVEL, AGE) CLASS(NAME, MEETS\_AT, ROOM, FID) ENROLLED(SNUM, CNAME)

FACULTY(FID, FNAME, DEPTID )

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.

Write the following queries in SQL. No duplicates should be printed in any of theanswers.

* 1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Rakesh.
  2. Find the age of the oldest student who is either a History major or enrolled in acourse taught by Ravi.
  3. Find the names of all students who are enrolled in two classes that meet at thesame time.
  4. For each faculty member that has taught classes only in room R128, print the faculty member’s name and the total number of classes she or he has taught. For each level, print the level and the average age of students for that level

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Open Ended Enquiry:**  Students must come up with a mini project that involves the database design for the given scenarioand implement the same using relational database management system and a front end tool. | | | | |  |
|  | Type of Experiment | | Program Number | Weightage |
| Demonstration | | 1,2 | 36% |
| Exercise | | 3,4 | 36% |
| Structured Enquiry | | 5 | 18% |
| Open ended | |  | 10% |
|  | | | | |
| **Textbook**   1. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2017. | | | | | |
| **Reference Books**   1. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, Mc-GrawHill, 2013. 2. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, ImplementationManagement,Cengage Learning 2012. 3. Ramakrishnan and Gehrke, Database management systems,3rd Edition, McGraw Hill, 2014 | | | | | |
| **Course Outcomes (COs)**  At the end of the course, students will be able to | | | | | |
| **CO1** | | **Explain** concepts of relational database management systems | | | |
| **CO2** | | **Apply** SQL querying techniques to access database for given specifications | | | |
| **CO3** | | **Design** database system for real world application | | | |
| **CO4** | | **Develop** database applications for a given case study by applying project management in a  team. | | | |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  |  | 2 |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |
| **CO3** |  |  | 2 |  |  |  |  |  |  |  |  |  | 2 |  |
| **CO4** |  |  |  |  | 2 |  |  | 2 |  |  |  |  | 2 |  |
| **Avg.** | **2.5** |  | **2** |  | 2 |  |  | 2 |  |  |  |  | **2.25** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER– IV** | | | |
| **Course Name** | : Principles of Artificial  Intelligence | **Course Code**: | BCIPA404 |
| **No. of Lecture Hours /**  **Week** | : 03 | **CIE Marks**: | 50 |
| **No. of Tutorial / Practical**  **Hours / Week** | : 00 | **SEE Marks**: | 50 |
| **Total No. of Lecture +**  **Tutorial / Practical Hours** | : 40 | **SEE Duration**: | 03 Hrs. |
| **L:T:P** | **:** 3:0:0 | **Credits**: | 03 |
| **Course Prerequisite**  Basic concepts of algorithms, complexity analysis and problem-solving techniques. | | | |
| **Course Overview**  This course provides a strong foundation of fundamental concepts in Artificial Intelligence. To providea basic exposition to the goals and methods and to enable the student to apply these techniques in applications which involve perception, knowledge and reasoning. | | | |
| **Course Learning Outcomes (CLO):**  This course will enable students to,   * Gain a historical perspective of AI and its foundations. * Become familiar with basic principles of AI toward problem solving. * Get to know approaches of inference, perception, Uncertain Knowledge and Reasoning | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1**  **Introduction**: What is AI? Foundations of AI.  **Intelligent Agents:** Agents and environment, Concept of Rationality, The nature of environment, The structure of agents.  **SLT:** History of AI  **Textbook 1:** Chapter 1- 1.1, 1.2, 1.3 Chapter 2- 2.1, 2.2, 2.3, 2.4 | | | **08** |
| **Module 2**  **Problem‐solving:** Problem‐solving agents, Example problems, Searching for Solutions Uninformed Search Strategies: Breadth First search, Depth First Search, Iterative deepening depth first search.  **SLT:** Real-world problems  **Textbook 1:** Chapter 3- 3.1, 3.2, 3.3, 3.4 | | | **08** |

|  |  |  |
| --- | --- | --- |
| **Module 3**  **Informed Search Strategies:** Heuristic functions, Greedy best first search, A\*search. Heuristic  **Functions Logical Agents:** Knowledge–based agents, The Wumpus world, Propositional logic, Reasoning patterns in Propositional Logic  **SLT:** Logic  **Textbook 1:** Chapter 4 – 4.1, 4.2 Chapter 7- 7.1, 7.2, 7.3, 7.4, 7.5 | | **08** |
| **Module 4**  **First Order Logic:** Representation Revisited, Syntax and Semantics of First Order logic, Using First Order logic.  **Inference in First Order Logic: Propositional** Versus First Order Inference, Unification, Forward Chaining, , Resolution SLT: First order logic-representation revisited  **SLT:** Backward Chaining  **Textbook 1:** Chapter 8- 8.1, 8.2, 8.3 Chapter 9- 9.1, 9.2, 9.3, 9.4, 9.5 | | **08** |
| **Module 5**  **Uncertain Knowledge and Reasoning:** Quantifying Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference using Full Joint Distributions, Independence, Baye’s Rule and its use.  **SLT:** Wumpus World Revisited  **Textbook 1:** Chapter: | | **08** |
| **Textbooks:**   1. Stuart J. Russell and Peter Norvig , Artificial Intelligence, 3rd Edition, Pearson,2015 | | |
| **Reference Books:**   1. Saroj Kaushik, Artificial Intelligence, 3rd Edition, Cengage learning, 2014, ISBN- 13:978-8131510995. 2. Elaine Rich, Kevin Knight: “Artificial Intelligence”, 3rd Edition, Tata McGraw Hill, 2018, ISBN- 13: 9780070087705. | | |
| **Course Outcomes (COs)**  At the end of the course students will be able to | | |
| **CO1** | **Understand** the fundamental knowledge of Artificial Intelligence. | |
| **CO2** | **Apply** basic principles of AI in problem solving and knowledge reasoning. | |
| **CO3** | **Analyze** the key components of intelligent agents and evaluate their performance. | |
| **CO4** | **Present** the solution for a given problem using AI techniques in a team. | |

**CO – PO – PSO Matrix:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  |  |  | 3 |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  |  |  | 2 |
| **CO4** |  |  |  |  |  |  |  | 2 | 2 |  |  |  |  | 2 |
| **Avg.** | **2.5** | **2** |  |  |  |  |  | **2** | **2** |  |  |  |  | **2.25** |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name** | : General Proficiency Enhancement Course – II | **Course Code :** | BGPEK405 |
| **Number of Lecture**  **Hours / Week** | : 00 | **CCE Marks :** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 02 | **SEE Marks :** | 50 |
| **Total Number of Lecture**  **+**  **Tutorial/PracticalHours** | : 30 | **SEE Duration :** | 02 Hrs. |
| **L:T:P** | : 0:2:0 | **Credits :** | 01 |
| **Course Prerequisites:** Knowledge of basic mathematics. | | | |
| **Course Overview:**  Aptitude training is designed to enhance an individual's problem-solving, analytical, and critical  thinking skills. It is often a crucial component of career development, competitive exams, and job recruitment processes. | | | |
| **Course Learning Objectives (CLO) :**   1. Develop Problem Solving Skills. 2. Enhance Quantitative Aptitude, Logical Reasoning and Verbal Ability Skills. 3. Prepare students for job recruitment process and competitive exams. | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1:**  **Quantitative Aptitude:** Time and Work, Time, Speed and Distance, Trains, Boats and Streams, Races. | | | **06** |
| **Module 2:**  **Quantitative Aptitude:** Permutation and Combination, Probability.  **Logical Reasoning:** Data Arrangements. | | | **06** |
| **Module 3:**  **Verbal Ability:** Ordering of Sentences, Reading Comprehension, Critical Reasoning | | | **06** |
| **Module 4:**  **Quantitative Aptitude:** Ages, Data Interpretation.  **Logical Reasoning:** Clocks, Calendars, Syllogisms. | | | **06** |
| **Module 5:**  **Quantitative Aptitude:** Mensuration, Crypt Arithmetic.  **Logical Reasoning:** Data Sufficiency, Cubes, Visual Reasoning | | | **06** |



|  |  |
| --- | --- |
| **Textbooks:**   1. Quantitative Aptitude for Competitive Examinations by R.S Aggarwal 2. A Modern Approach to Verbal & Non-Verbal Reasoning by R.S. Aggarwal | |
| **Course Outcomes (COs):**  **Upon completion of the course, students will be able to** | |
| **CO1** | **Solve** complex problems using logical and structured approaches. |
| **CO2** | **Analyze** complex problems using logical and structured approaches. |
| **CO3** | **Develop** critical thinking skills, enabling them to evaluate information and make effective  decisions. |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** |
| **CO1** | 2 |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 2 |  |  |  |  |  |  |  |  |  |
| **CO3** |  |  | 2 |  |  |  |  |  |  |  |  |
| **Avg.** | **2** | **2** | **2** |  |  |  |  |  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER-IV** | | | |
| **Course Name** | : Biology for IT Engineers | **Course Code:** | BITBE406 |
| **Number of Lecture Hours /Week** | : 02 | **CIE Marks:** | 50 |
| **Number of Tutorial / Practical Hours / Week** | : 00 | **SEE Marks:** | 50 |
| **Total Number of Lecture + Tutorial/ PracticalHours** | : 25 |  |  |
|  | **SEE Duration:** | 02 Hrs. |
| **L:T:P** | : 2:0:0 | **Credits:** | 02 |
| **Course Prerequisites:** Fundamentals of Biology. | | | |
| **Course Overview**  This course provides a basic understanding of biological mechanisms of living organisms from the perspective of engineers. In addition, the course is expected to encourage engineering students  to think about solving engineering problems using bioinspired solutions. | | | |
| **Course Learning Objectives (CLOs)**  This course will enable students   * To familiarize with the basic biological concepts and their applications * To involve students in an interdisciplinary vision of biology and engineering. * To gain an appreciation for how biological systems can be designed and engineered tosubstitute naturalsystem. | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **Module 1: Cell Biology & Molecular Biology**  **Cell biology:** Introduction to cell, Basic properties of Cells, Cell overview, Prokaryotic and Eukaryotic Cell, Cell cycle and Cell division, Cell differentiation **Molecular structure:** Chemical composition of Living forms, carbohydrates, aminoacids and proteins.  **Textbook 1**: (2.1 to 2.6), 2.9, 3.1, 3.3, 3.4 | | | **05** |
| **Module 2: Introduction to Human Anatomy**  Overview of human anatomy, Structural organization of the human body – skeletal system, nervous system, respiratory system, circulatory system, endocrine system,digestive system  **Textbook 2:** (1.1 to 1.2), (2.1 to 2.2) ,3.1, (4.1 to 4.3) , (5.1 to 5.2), (7.1 to 7.3) | | | **05** |



|  |  |  |
| --- | --- | --- |
| **Module 3: Bioinformatics**  Introduction to basics of Bio informatics, the Recent Development in the field of Bio informatics, Challenges of Bioinformatics, Applications ofBioinformatics  **Textbook 3:** 1.1, 1.4, 1.5, 1.6 | | **05** |
| **Module 4: Evolutionary Algorithms and Genetics**  History of Evolutionary Computation, Motivation via Biological Evidence, Why Evolutionary Computing? Overview of Genetic Algorithms, Pros and Cons of Genetic Algorithms.  **Textbook 4:** (1.1 to 1.3),2.1, 2.5 | | **05** |
| **Module 5: Bio Inspired Applications**  Ant Colony Optimization: Introduction, Concept of Artificial Ants, Foraging behavior of Ants, Application of ACO in Different Fields, Particle Swarm Optimization: Basic Concepts Particle swarm optimization (PSO), The Artificial Bee Colony Algorithm: Introduction.  **Textbook 4:** (4.1 to 4.3), 4.11, (5.1 to 5.2), 6.1 | | **05** |
| **Textbooks**   1. Biology for Engineers AICTE, Wiley publication, 2018 2. A Textbook to Human Physiology – H.S. Ravi Kumar Patil , H.K. Makari , H. Gurumurthy &   S.V. Sowmya,DreamTech publications , 2021 edition   1. Bioinformatics methods & application by Dev Bukhsh Singh , Rajeshkumar patak, academic press, 2021 2. Advances in Swarm Intelligence for Optimizing Problems in Computer Science, Anand Nayyar,Le Dac-Nhuong, Nguyen Nhu Gia - -Chapman and Hall\_CRC (2019) | | |
| **Reference Books**   1. **“**Biology for Engineers,” Arthur T. Johnson, CRC Press, Taylor and Francis, 2011, Johnson, A. T. 2018. 2. “Biology for Engineers”, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014 | | |
| **Course Outcomes(CO’s):**  Upon Completion of the course, students will be able to | | |
| **CO1** | **Understand** the basic biological principles of living systems at cell, molecular level and  human anatomy | |
| **CO2** | **Examine** the basics of Bio informatics, evolutionary algorithm and bioinspired techniques | |
| **CO3** | **Investigate** the engineering concepts for developing next generation technologies | |
| **CO4** | **Present** engineering problems and solutions using the concepts of Biology | |



**CO-PO-PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** | 3 |  |  |  |  |  |  |  |  |  |  | **3** |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  |  | **2** |  |
| **CO4** |  |  |  |  |  | **2** |  |  | 2 |  |  |  | **2** |  |
| **Avg.** | **3** | **2** |  |  |  | **2** |  |  | **2** |  |  | **3** | **2** |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER-IV** | | | |
| **Course Name** | : Advanced Web  Programming Laboratory | **Course Code:** | BCIAW417 |
| **Number of Lecture Hours**  **/ Week** | : 00 | **CIE Marks:** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 04 | **SEE Marks:** | 50 |
| **Total Number of Lecture**  **+ Tutorial/Practical Hours** | : 48 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 0:0:4 | **Credits:** | 02 |
| **Course Prerequisites:**  Basic knowledge of programming, HTML, CSS and JavaScript are required to learn the course. | | | |
| **Course Overview:** The focus of this course is on the World Wide Web as a platform for interactive applications, content publishing and social services. This course is about the HTTP communication protocol, the markup languages HTML, the CSS for formatting and transforming web content, XML, AJAX,  Node.js and PHP. | | | |
| **Course Learning Objectives (CLO)**  This course will enable students to,   * Learn the syntax and semantics of web technologies. * Demonstrate the use of built-in functions. * To understand the fundamentals of Nodejs, XML, AJAX and PHP. * Interpret the use of procedural statements like conditional statements, loops and function callsin JavaScript | | | |
| **PART A** | | | |
| 1. Design a HTML program to display two paragraphs with following specifications:   P1: display the heading called “ADVANCED WEB PROGRAMMING” and set the text color to white andbackground color to blue.  P2: A paragraph with the basic elements such as bold, italic, subscript, super script, mark and with ahorizontal line and line break   1. Create a webpage to display a table consisting of a customer detail such as customerID, Name,   Experience and Salary. | | | |



* + 1. Create a webpage to display an online shopping website with the list of items available with the concept of ordered and unordered list.
    2. Design a webpage to create a form for a student to enroll for a course which includes the following fields: Name, USN, Gender Subject interested with the drop-down options, email, Contact Number,Address with submit and reset button.
    3. Design HTML webpage with following CSS specifications: Show any two-font family through text, decorated the text through underline, overline and line through, display the head text with shadow in red color.
    4. Develop javascript code to check whether the given number is prime or not.
    5. Write a JavaScript to design a simple calculator to perform the following operations: sum, product,difference, and quotient.
    6. Develop javascript code to find the largest of three numbers.

NOTE: The above programs will be assessed in CIE ONLY.

**PART B**

1. Develop a javascript to sort and accessing the array elements.Input: n=5

Unsorted array elements: 10,1,- 5,4,15 Sorted Array: -5,1, 4, 10,

15

1. Create a class by the name rectangle with 2 attributes length and breadth. Include a parameterized constructor to assign values to data members and a function to calculate area of the rectangle. Demonstrate creation of object of class rectangle and display its area.

Input: length=5, breadth=6Output: Area=30

1. Develop a javascript to demonstrate the working of callback and async functions.
2. Develop an arrow function in javascript that checks whether a year is leap year, alert the user with true if the year is leap year and false if year is non leap year. Validate centuries also.

Input: 2000, Output: Leap year Input: 2100, Output: Non Leap year

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input: 2004, Output: Leap year Input: 2006, Output:  Non Leap year   1. Develop a javascript that accepts length and breadth of rectangle as parameter of an arrow functions.Call the function using spread and rest operator and alert the user with perimeter of the rectangle. 2. Develop a javascript to demonstrate the usage of optional and default parameters in a function. 3. Create a class by the name box with parameters length, breadth, and height. Create a class boxweightthat extends box and include a new parameter weight. Create another class by the name boxcost that extends boxweight and has a parameter by the name shipmentcost. Include constructors in all the classes. Create an object of boxcost and display values of all parameters that represent multilevel inheritance. 4. Cretae a database with studentname and usn. Develop a javascript that accesses this database using get methods of REST API to display the contents of database in webpage. | | | | | |
| **Open Ended Experiments**  Students shall solve a problem (either given by the staff or student may come up with their own problem) using the design techniques.   1. Weather forecast website. 2. Javascript Music Player. 3. Hospital Management System. 4. Online Shopping Webpage. | | | | | |
| **Experimental Weightage:** | | | | | |
|  | **Type of Experiment** | | **Program -No** | **Weightage** |  |
| Exercise | | 5, 7, 8 | 33.3% |
| Structure Enquiry | | 1, 2, 4 | 33.3% |
| Demonstration | | 3, 6 | 22.2% |
| Open ended | |  | 11.1% |
| **Course Outcomes:** | | | | | |
| **CO1** | | **Apply** and demonstrate the usage of various programming constructs | | | |
| **CO2** | | **Design** and **Develop** applications to solve problems across various technical and real-  world domains | | | |
| **CO3** | | **Explore** various programming tools and techniques in a team | | | |

**CO-PO-PSO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** |  |  | 3 |  |  |  |  |  |  |  |  |  | 2 |  |
| **CO3** |  |  |  |  | 2 |  |  | 2 |  |  |  |  | 2 |  |
| **Avg.** | **3** |  | **3** |  | **2** |  |  | **2** |  |  |  | **2** | **2** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name** | : Python Programming for Machine Learning  Laboratory | **Course Code:** | BCIPM417 |
| **Number of Lecture Hours /**  **Week** | : 00 | **CIE Marks:** | 50 |
| **Number of Tutorial /**  **Practical Hours / Week** | : 04 | **SEE Marks:** | 50 |
| **Total Number of Lecture + Tutorial/Practical Hours** | : 48 | **SEE Duration:** | 03 Hrs. |
| **L:T:P** | : 0:0:4 | **Credits:** | 02 |
| **Course Prerequisite:** Basic knowledge of programming like python and C. | | | |
| **Course Overview**  This course provides hands-on coding exercises on programming concepts like classes, custom functions  and libraries such as numpy, pandas and Data Visualization. | | | |
| **Course Learning Objectives (CLO)**  The course should enable the students to,   * Understand and implement the basic constructs of object-oriented programming. * Develop programs for data manipulation and data operations using numpy and pandas. * Perform pattern matching using regular expressions | | | |
| **PRACTICAL MODULE PART-A**   1. Write a python program to check whether the number is prime or not.   *(Concepts Explored : Looping and Conditional Statements )*   1. Write a Python program to calculate the length of a string without using built in function and checkwhether it is a palindrome or not.   *(Concepts Explored : Strings, Slicing)*   1. Write a Python function that accepts a sentence from the user and counts the number of words inthe sentence.   *(Concepts Explored : String splitting, List)*   1. Write a Python program to sum all the numbers in a list.   *(Concepts Explored : List )*   1. Write a Python program to store phone numbers of persons in a dictionary and display the phonenumber of the requested person.   *(Concepts Explored : Dictionary)*   1. Develop a python program to copy the contents of only odd lines from one file to another file.   *(Concepts Explored : File Handling)*   1. Write a python program to check if the number is even or odd using a user defined function. *(Concepts Explored : User defined function)* | | | |

8. Develop a python program to count occurrence of all characters present in a string.

*(Concepts Explored : Dictionary get() method)*

**NOTE:** The above programs will be assessed in CIE-1 ONLY.

**PART-B**

# A- Demonstration

**A1.** Write a program to build a simple Student Management System using Python which can perform thefollowing operations: Accept, Display, Search, Delete, Update.

**A2. a)** Write a Pandas program to display a summary of the basic information about a specified Data Frame and its data.

**b)** Read the given data “churn.csv” and save it as a data frame called churn\_data. Perform followingoperations on the data frame i) Count total number of duplicate records in the data frame ii) Count the no. of duplicate records in the churn data frame based on the cutomerID column iii) Count number of missing values in each columns iv) Count the total no. of missing values for the variable TotalCharges v) Average monthly charge paid by a customer for the services he/she has signed up for vi) Display the records having “1@#” under the variable Dependents vii) Replace null values in churn data frame by median value or by max count class category (Demonstration)

*(*[*https://drive.google.com/file/d/1JSYGIIkIZr4jyheDEH0X1\_TMSnQ2CzXc/view*](https://drive.google.com/file/d/1JSYGIIkIZr4jyheDEH0X1_TMSnQ2CzXc/view)

**A3.** Explore the automobile dataset and visualize the i) Distribution of the two and four door cars with respect to the type of fuel they use ii) Distribution of cars of different body styles with respect to the type of fuel they use ii) Total number of each type of body style cars categorized by fuel type iii) Horsepowerof each of the fuel type with reference to the type of drive wheel present in cars. (Demonstration)

*(https://*[*www.kaggle.com/datasets/toramky/automobile-*](http://www.kaggle.com/datasets/toramky/automobile-dataset?resource=download))[*dataset?resource=download)***.**](http://www.kaggle.com/datasets/toramky/automobile-dataset?resource=download))

# B- Exercise

**B1.** a) Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

1. Write a NumPy program to save a given array to a text file and load it.

**B2.** Design a Python program using Regular expressions to

1. Extract Email IDs from a given text.
2. Validate the user password with minimum length=6 and maximum length=16 and must have atleast one lower-case letter, upper-case letter, number and special symbol (#, @, $, \_).

**B3.** Write a NumPy program to create a structured array from given student name, height, class and theirdata types. Finally sort the array on height.

# C- Structured Inquiry

**C1.** Develop a pandas program to read CSV file (items.csv, given below) and replace the null values in *quantity* by mean, *price* by median, *bought* by standard deviation, *forenoon* by min and *afternoon* by max.

|  |  |
| --- | --- |
| **C2.** Using the data on births in the United States, provided by the Centers for Disease Control (CDC), Find   1. Total number of US births by year and gender ii) Average daily births by day of week and decade iii) Average daily births by date. (Demonstration)   *(https://raw.githubusercontent.com/jakevdp/data-CDCbirths/ master/births.csv)*  ***C3.*** Visualize and understand finishing results from a marathon race with respect to distribution of split fractions, distribution of split fractions by gender, distribution of split fractions by gender and age. (Exercise)  *(*[*https://raw.githubusercontent.com/jakevdp/marathon-data/master/marathon-data.csv*](https://raw.githubusercontent.com/jakevdp/marathon-data/master/marathon-data.csv)*)*  ***D- Open Ended Experiments***  The student can choose to solve any one open-ended problem to illustrate python application in thedomains specified below (but not restricted to) using various python packages.   * 1. Excel file handling   2. PDF/word file manipulation 3.CSV file analysis using pandas.   4.Simple Games 5.Demonstrate polymorphism | |
| **Course Outcomes (COs)**  At the end of the course students will be able to | |
| **CO1** | **Apply** the concepts of object-oriented programming for modelling and solving real world problems |
| **CO2** | **Demonstrate and Develop** the use of regular expression for pattern matching |
| **CO3** | **Apply and Explore** various python packages for data exploration and manipulation techniques using suitable tools. |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO2** |  |  | 3 |  |  |  |  |  |  |  |  | 2 |  |  |
| **CO3** |  |  |  |  | 2 |  |  | 2 |  |  |  | 2 |  |  |
| **Avg.** | **3** |  | **3** |  | 2 |  |  | **2** |  |  |  | **2** |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – IV** | | | |
| **Course Name** | : Technical Proficiency  Enhancement Course-II | **Course Code:** BITTP408 | |
| **Number of Lecture Hours /**  **Week** | : 00 | **CIE Marks :** 50 | |
| **Number of Tutorial /**  **Practical Hours / Week** | : 02 | **SEE Marks :** 50 | |
| **Total Number of Lecture +**  **Tutorial/Practical Hours** | : 30 | **SEE Duration :** 02 Hrs. | |
| **L:T:P** | : 0:0:2 | **Credits :** 01 | |
| **Course Prerequisite:** Working knowledge of Data Structures and C++ Programming. | | | |
| **Course Overview:** This course is designed to enhance an individual’s problem-solving, analytical, and critical thinking skills using data structures and C++ programming language. It also helps the students to acquire skills for career development, competitive exams, and job recruitment processes. | | | |
| **Course Learning Objectives (CLO)**  The course should enable the students to,   * To enhance problem-solving skills * To develop technical Skill such as Data structures and OOPs Concepts * To Prepare students for job recruitment process and competitive exams | | | |
| **MODULES** | | | **TEACHING**  **HOURS** |
| **Module 1: Problem Solving Techniques**  **Recursion:** Introduction to recursion, Principle of mathematical induction, Fibonacci numbers, Recursion using arrays, Recursion using strings, Recursion using 2D arrays.  **Time and Space Complexity:** Order complexity analysis, Theoretical complexity analysis, Time complexity analysis of searching and recursive algorithms, Theoretical space complexity, Space complexity analysis of merge sort.  **Backtracking:** Introduction to Backtracking, Rat In a Maze, N-queen, Word Search. | | | **06** |
| **Module 2: Object-oriented Programming**  **Basics of OOP:** Introduction to oops, Creating objects, Getters, and setters, Constructors and related concepts, Inbuilt constructor and destructor, Example classes. **Advance Concepts of OOP:** Static members, Function overloading and related concepts, Abstraction, Encapsulation, Inheritance,  Polymorphism, Virtual functions, Abstract classes, Exception handling. | | | **06** |
| **Module 3: Linear Data Structures 1**  **Linked Lists:** Introduction to linked list, Inserting node in linked list, Deleting node fromlinked list, Midpoint of linked list, Merge two sorted linked lists, merge sort  of a linkedlist, Reversing a linked list. | | | **06** |
| **Module 4: Linear Data Structures 2**  **Stacks and Queues:** Introduction to stacks, Stack using arrays, Dynamic Stack  class, Stack using linked list, Inbuilt stack, Queue using arrays, Dynamic queue class, Queue using linked list, Inbuilt queue. | | | **06** |



|  |  |
| --- | --- |
| **Module 5: Trees**  **Generic Trees:** Introduction to Trees, Making a tree node class, Taking a tree as inputand printing, Tree traversals, Destructor for tree node class.  **Binary Trees**: Introduction to Binary Trees, Taking a binary tree as input and printing,Binary Tree traversals, Diameter of binary tree.  **Binary Search Trees**: Introduction to Binary Search Trees, Searching a node in BST, BST class, Inserting and Deleting nodes in BST, Types of balanced BSTs. | **06** |

|  |  |
| --- | --- |
| **Course Outcomes (COs):**  **Upon completion of the course, students will be able to** | |
| **CO1** | **Apply** various Data structure and OOPs concepts such as Backtracking, trees, linked list, exceptionhandling |
| **CO2** | **Develop** critical thinking skills, enabling them for job interviews and assessment tests |

**CO – PO – PSO Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO** | **PO** | | | | | | | | | | | **PSO** | | |
| **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| **Avg.** | 3 | **3** |  |  |  |  |  |  |  |  |  |  |  |  |