LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of CE/IT/CSD/AIML/AIDS/CSE/CST/CS&IT/CEA/MA&CP/CSE(AI)/ CSE(DS)/CSE(CS)

(701,702,703,704,705, 712,713,714,715,716,717,718,719)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**017011291/017021291/017031291/017041291/017051291/017121291/ 017131291/017141291/017151291/017161291/017171291/017181291/ 017191291**
Course Name:	**Mathematics - II**
Category of Course:	Basic Science Course (BSC)
Prerequisite Course:	Mathematics - I (117011191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**2**	**0**	**5**	**50**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Matrices**			**7 (14%)**
	1.1 Elementary row operations of matrices	---	Divide & Conquer (017013591-Unit-3)	
	1.2 Row and reduced row echelon form	---		
	1.3 System of linear equations	---		
	1.4 Homogeneous system of linear equations	---		
	1.5 Non-homogeneous system of linear equations	---		
	1.6 Inverse of Matrix (Using Gauss-Jordan Method)	---	Cryptography(017013791-Unit-2)	
	1.7 Eigen values & vectors	Factorization(017011191-Unit-1)	---	
	1.8 Diagonalization of matrix (Only for Non-symmetric Matrix)		---	
	1.9 Cayley-Hamilton theorem		---	
02	**Fourier Series**			**5 (10%)**
	2.1 Periodic function	Basic integration (017011191-Unit-3)	---	
	2.2 Dirichlet's condition		---	
	2.3 Trigonometric series of sine and cosine function		---	
	2.4 Fourier series of a function of period 2L		---	
	2.5 Fourier series of even and odd function		---	
	2.6 Half range expansions		---	
03	**Some Special Functions**			**4 (8%)**
	3.1Gamma function, Beta function. (And its Properties)	---	---	
	3.2 Bessel function, Dirac's Delta function (Definition only)	---	---	
	3.3 Error function and complementary Error function (Definition only)	---	---	
	3.4 Heaviside's function, pulse unit height and duration function (Definition only)	---	---	
	3.5 Rectangle function, Gate function (Definition only)	---	---	
	3.6 Signum function, Saw tooth wave function (Definition only)	---	---	
	3.7 Triangular wave function, Halfwave rectified sinusoidal function, Full rectified sine wave, Square wavefunction. (Definition only)	---	---	
04	**Fourier Integral and Fourier Transform**			**4 (8%)**
	4.1 Define Fourier integral	Basic integration (017011191-Unit-3)	---	
	4.2 Cosine and sine integral		---	
	4.3 Define Fourier transform		---	
	4.4 Cosine and sine transform		---	
05	**First Order Ordinary Differential Equations**			**5 (10%)**
	5.1 Geometric meaning of $y' = f(x, y)$ direction fields	---	---	
	5.2 Exact differential equations and integrating factor	Basic differentiation & integration(017011191-Unit-3)	---	
	5.3 Linear differential equations		---	
	5.4 Bernoulli equations		---	
06	**Higher Order Ordinary Differential Equations**			**7 (14%)**
	6.1 Linear differential equations of second and higher order	---	---	
	6.2 Homogeneous linear differential equations of higher order	Factorization(017011191-Unit-1)	---	
	6.3 Higher order non-homogeneous equations		---	
	6.4 Solution by undetermined coefficients		---	
	6.5 Solution by variation of parameters		---	
	6.6 Solution by [1/f(D)] r(x) method for finding particular integral.		---	
	6.7 Ordinary differential equations with variable coefficient (Reducible to constant coefficient) (Cauchy-Euler's & Legendre's Equation)	Solution by undetermined coefficients (017011291-Unit-6), Solution by [1/f(D)] r(x) method for finding particular integral (017011291-Unit-6)	---	

07	**Modeling of Ordinary Differential Equations**			**3 (6%)**
	7.1 Orthogonal trajectories of curves (Only Cartesian Curves)	First order ordinary differential equations (017011291-Unit-5)	---	
	7.2 Oscillations and resonance (For undamped Forced Oscillations)	Higher order ordinary differential equations (017011291-Unit-6)	---	
	7.3 Modeling: Electric Circuits (Only RLC-Circuit)		---	
08	**Power Series**			**5 (10%)**
	8.1 Classification of singularities	---	---	
	8.2 Series solution near ordinary points	---	---	
	8.3 Series solution near regular singular points (Frobenius Method)	---	---	
09	**Laplace Transform**			**7 (14%)**
	9.1 Laplace transform of elementary functions	Basic differentiation & integration(017011191-Unit-3)	---	
	9.2 Differentiation of Laplace transform		---	
	9.3 Integration of Laplace transform		---	
	9.4 Laplace transform of derivatives		---	
	9.5 Laplace transform of integrals		---	
	9.6 Unit step function and Dirac's delta function		---	
	9.7 Inverse Laplace transform		---	
	9.8 Convolution theorem		---	
10	**Application of Laplace Transform**			**3 (6%)**
	10.1 Solution of linear ordinary differential equation	Laplace transform of elementary functions, Laplace transform of derivatives, Unit step function and Dirac's delta function, Inverse Laplace transform, Convolution theorem(017011291-Unit-9)	---	
	10.2 Solution of simultaneous equations		---	

Course Outcome	
	Upon completion of the course students will be able to
CO1	Understand and apply matrix operation and properties, solve systems of linear equations using matrices, analyze systems using eigen values and eigne vectors, apply matrices in signal processing tasks, Explain the concept of Fourier series and its properties, Apply Fourier series in digital communications and image processing also in control system analysis.
CO2	Design filter and modulation schemes and Implement algorithms like FFT for efficient computation of Fourier transforms, Apply Bessel functions and other special function to solve engineering problems. Solve first order & first degree ODEs using various methods.
CO3	Solve higher order linear ODEs using various methods such as undetermined coefficients, variation of parameters. Formulate ODEs from real-world engineering problems. Apply knowledge of ODEs to design and analyze systems in computer engineering domains. Apply orthogonal trajectories in edge detection algorithms for image processing and utilize for curve fitting and surface modeling in computer graphics. Understand the significance of ordinary and singular points in ODEs.
CO4	Understand and apply Laplace transforms to solve linear ODEs with constant coefficients. Apply knowledge to real-world engineering problems, especially in signal processing, circuit analysis, control system and system modeling.
Suggested Reference Books	
1	Elementary Linear Algebra, Applications version, Anton and Rorres, Wiley India Edition.
2	Advanced Engineering Mathematics, Erwin Kreysig, Wiley Publication.
3	Advanced Engineering Mathematics, Dennis G. Zill, 4th edition, Jones and Bartlett Publishers.
4	Higher Engineering Mathematics, B.S.Grewal, Khanna Publishers.

List of Open Source Software/Learning website	
1	https://nptel.ac.in

LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of CE/IT/CSD/AIML/AIDS/CSE/CST/CS&IT/CEA/MA&CP/CSE(AI)/ CSE(DS)/CSE(CS)

(701,702,703,704,705,712,713,714,715,716,717,718,719)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**117013291/117023291/117033291/117043291/117053291/117123291/117133291/117143291/117153291/017163291/017173291/017183291/017193291**
Course Name:	**Database Management System**
Category of Course:	Professional Core Course (PCC)
Prerequisite Course:	Computer Programming using Java-1 (117012191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**0**	**6**	**6**	**30**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Introduction of Database**			**1 (3%)**
	1.1 Database-System Applications, Purpose of Database Systems	---	---	
	1.2 Types of Database Models, Data Independence		---	
	1.3 Database Architecture-Levels, Data Independence, Database Languages, Components of a DBMS Architecture, Database Users and Administrators (DBA)		---	
02	**Entity-Relationship Model**			**2 (8%)**
	2.1 Basic Concepts, Design Process, Entity-Relationship Model	---	---	
	2.2 Constraints, Entity-Relationship Diagrams and its Design Issues	Basic Concepts (117013291 Unit – 2)	---	
	2.3 Extended E-R Features– Generalization, Specialization, Aggregation, Reduction to E-R Database Schema	Basic Concepts (117013291 Unit – 2)	---	
03	**Basics of SQL**			**4 (12%)**
	3.1 Basics of SQL- Rules, Process, Characteristics, Advantages, , SQL Data types (Numeric, Varchar, Char, Integer, Date, Time, Timestamp), DDL, DML, DQL, DCL, Structure – Creation, Alteration	---	SQL Constraints and Functions (117013291 Unit-4), Structured Query Language (SQL) (117013291 Unit - 7), PL/SQL Concepts (117013291 Unit-9), Full Stack Development using JavaScript (017013691 Unit – 10.2), Fundamentals of Computer Science using Python-1(017012491 Unit – 9.1,9.2)	
04	**SQL Constraints and Functions**			**4 (12%)**
	4.1 Defining Constraints – Primary Key, Foreign Key, Unique, Not Null, Check, IN	Basics of SQL (117013291 Unit – 3)	SQL Constraints and Functions (117013291 Unit-5), Structured Query Language (SQL) (117013291 Unit - 7), PL/SQL Concepts (117013291 Unit-9)	
	4.2 Functions – **Aggregate Functions** AVG,COUNT,SUM,MAX,MIN **Built-In Functions –** **1. Numeric/Math** ABS,CEIL,DIV,EXP,FLOOR,MOD, POWER,ROUND,SQRT,TRUNC) **2. Date** AGE,CURRENT_DATE,CURRENT_TIME,CURRENT_TIMESTAMP,EXTRACT [EXTRACT TIMESTAMP -DAY OF MONTH,DAY OF WEEK,DAY OF YEAR,MONTH,WEEK,HOUR,MINUTE,SECOND] **3. String Functions** ASCII,CHR,CONCAT,CONCAT_WS, INITCAP,LEFT,LENGTH,LOWER,LPAD,LTRIM,POSITION, REPEAT,REPLACE, REVERSE,RIGHT,RPAD,RTRIM,SUBSTRING,TRIM,UPPER	---	SQL Constraints and Functions (117013291 Unit-5), Structured Query Language (SQL) (117013291 Unit - 7), PL/SQL Concepts (117013291 Unit-9)	

	Set Operations UNION,UNIONALL,INTERSECT, MINUS			
05	**Relational Model**			**2 (8%)**
	5.1 Structure of Relational Databases, Schema, Keys, Domains, Relations	---	---	
	5.2 Relational Algebra – Fundamental Operators (Projection, Selection, Cross Cartesian, Union, Intersection, Set Difference, Join) and Syntax	Structure of Relational Databases (117013291 Unit – 5)	---	
	5.3 Relational Algebra Queries	Relational Algebra (117013291 Unit – 5)	---	
06	**Relational Database Design**			**2 (9%)**
	6.1 Functional Dependency – Definition, Partial FD, Trivial and Non-Trivial FD, Closure of FD Set, Closure of Attributes, Irreducible Set of FD	**---**	---	
	6.2 Normalization – 1nf, 2nf, 3nf, BCNF	Function Dependency (117013291 Unit – 6), Normalization (117013291 Unit - 6)	---	
	6.3 Decomposition Using FD- Dependency Preservation		---	
07	**Structured Query Language (SQL)**			**5 (15%)**
	7.1 Sub-Queries, Correlated Sub-Queries(Select Statement), Use of Where Clause, Group By, Having, Order By, SQL Conditions/ Operators (AND, OR, NOT, IN, NOT IN, BETWEEN NOT BETWEEN , LIKE)	Basics of SQL (117013291 Unit – 3), SQL Constraints and Functions (117013291 Unit-4)	PL/SQL Concepts (117013291 Unit-9)	
	7.2 Join and its Types, Exist, Not Exist, Any, All, View and its Types(Create, Update, Delete)	Basics of SQL (117013291 Unit – 3), SQL Constraints and Functions (117013291 Unit-4)	PL/SQL Concepts (117013291 Unit-9)	
	7.3 Transaction Control Commands – Commit, Rollback, Save point	Basics of SQL (117013291 Unit – 3), SQL Constraints and Functions (117013291 Unit-4)	PL/SQL Concepts (117013291 Unit-9)	
08	**Transaction & Recovery Management**			**3 (10%)**
	8.1 Transaction Concept, ACID Properties	---	---	
	8.2 Concurrent Executions of Transactions and Related Problems, Serializability, Testing for Serializability, Types of Serializability- Conflict and View	Transaction Concept (117013291 Unit – 8)	---	
	8.3 Solution to Concurrency Related Problems, Locking Mechanism, Two-Phase Locking Protocol, Two- Phase Commit Protocol	Transaction Concept, Serializability (117013291 Unit – 8)	---	
	8.4. System Recovery, Log-Based Recovery	Serializability, Concurrent Execution of Transactions (117013291 Unit – 8)	---	
09	**PL/SQL Concepts**			**5 (15%)**
	9.1 PL/SQL Concepts- Block structure, sub-block, Select Into statements, Control Structures (IF, WHIE LOOP, FOR LOOP) Cursors (Implicit, Explicit), Stored Procedures, Stored Function, Database Triggers	Basics of SQL (117013291 Unit – 3), Constraints and Functions (117013291 Unit-4), Structured Query Language (SQL) (117013291 Unit - 7)	---	
10	**Query Processing & Query Optimization**			**2 (8%)**
	10.1 Overview of Query Processing, Introduction of Query Optimization, Data Security, Audit trail	---	---	
	10.2 Introduction, Access Control Concept, Types of Access Controls Techniques	---	---	

<table>
<tr><th>Sr No.</th><th>Practical Title</th><th>Link to Theory Syllabus</th></tr>
<tr><td>1</td><td>Consider following databases and draw ER diagram and convert entities and relationships
to relation table for a given scenario.
1. COLLEGE DATABASE:
STUDENT (RN, SName, Address, Phone, Gender)
CLASS (RN, EnrolNo)
SUBJECT (Subcode, Title, Sem, Credits)
MARKS (RN, Subcode, EnrolNo, Test1, Test2, Test3, FinalIA)
2. COMPANY DATABASE:
EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo)
DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate)
DLOCATION (DNo,DLoc)
PROJECT (PNo, PName, PLocation, DNo)
WORKS_ON (SSN, PNo, Hours)</td><td>Unit-2</td></tr>
<tr><td>2</td><td>To study DDL-create and DML-insert commands.
From the below given tables, perform the following queries:
CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5) ,CNAME VARCHAR2(18) , BNAME VARCHAR2(18) , AMOUNT NUMBER(8,2) ,ADATE DATE);
CREATE TABLE BRANCH(BNAME VARCHAR2(18),CITY VARCHAR2(18));
CREATE TABLE CUSTOMERS(CNAME VARCHAR2(19) ,CITY VARCHAR2(18));
CREATE TABLE BORROW(LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));

DEPOSIT
<table><tr><th>ACT-NO</th><th>CNAME</th><th>BNAME</th><th>AMOUNT</th><th>ADATE</th></tr><tr><td>100</td><td>ANIL</td><td>VRCE</td><td>1000.00</td><td>1-MAR-95</td></tr><tr><td>101</td><td>SUNIL</td><td>AJNI</td><td>5000.00</td><td>4-JAN-96</td></tr><tr><td>102</td><td>MEHUL</td><td>KAROLBAGH</td><td>3500.00</td><td>17-NOV-95</td></tr><tr><td>104</td><td>MADHURI</td><td>CHANDI</td><td>1200.00</td><td>17-DEC-95</td></tr><tr><td>105</td><td>PRMOD</td><td>M.G.ROAD</td><td>3000.00</td><td>27-MAR-96</td></tr><tr><td>106</td><td>SANDIP</td><td>ANDHERI</td><td>2000.00</td><td>31-MAR-96</td></tr><tr><td>107</td><td>SHIVANI</td><td>VIRAR</td><td>1000.00</td><td>5-SEP-95</td></tr><tr><td>108</td><td>KRANTI</td><td>NEHRU PLACE</td><td>5000.00</td><td>2-JUL-95</td></tr><tr><td>109</td><td>MINU</td><td>POWAI</td><td>7000.00</td><td>10-AUG-95</td></tr></table>
BRANCH
<table><tr><td>VRCE</td><td>NAGPUR</td></tr><tr><td>AJNI</td><td>NAGPUR</td></tr><tr><td>KAROLBAGH</td><td>DELHI</td></tr><tr><td>CHANDI</td><td>DELHI</td></tr><tr><td>DHARAMPETH</td><td>NAGPUR</td></tr><tr><td>M.G.ROAD</td><td>BANGLORE</td></tr><tr><td>ANDHERI</td><td>BOMBAY</td></tr><tr><td>VIRAR</td><td>BOMBAY</td></tr><tr><td>NEHRU PLACE</td><td>DELHI</td></tr><tr><td>POWAI</td><td>BOMBAY</td></tr></table>
CUSTOMERS
<table><tr><td>ANIL</td><td>CALCUTTA</td></tr><tr><td>SUNIL</td><td>DELHI</td></tr><tr><td>MEHUL</td><td>BARODA</td></tr><tr><td>MANDAR</td><td>PATNA</td></tr><tr><td>MADHURI</td><td>NAGPUR</td></tr><tr><td>PRAMOD</td><td>NAGPUR</td></tr><tr><td>SANDIP</td><td>SURAT</td></tr><tr><td>SHIVANI</td><td>BOMBAY</td></tr><tr><td>KRANTI</td><td>BOMBAY</td></tr><tr><td>NAREN</td><td>BOMBAY</td></tr></table>
BORROW
<table><tr><th>LOANNO</th><th>CNAME</th><th>BNAME</th><th>AMOUNT</th></tr><tr><td>201</td><td>ANIL</td><td>VRCE</td><td>1000.00</td></tr><tr><td>206</td><td>MEHUL</td><td>AJNI</td><td>5000.00</td></tr><tr><td>311</td><td>SUNIL</td><td>DHARAMPETH</td><td>3000.00</td></tr><tr><td>321</td><td>MADHURI</td><td>ANDHERI</td><td>2000.00</td></tr><tr><td>375</td><td>PRMOD</td><td>VIRAR</td><td>8000.00</td></tr><tr><td>481</td><td>KRANTI</td><td>NEHRU PLACE</td><td>3000.00</td></tr></table>
1. Describe deposit, branch.
2. Describe borrow, customers.
3. List all data from table DEPOSIT.
4. List all data from table BORROW.
5. List all data from table CUSTOMERS.
6. List all data from table BRANCH.
7. Give account no and amount of depositors.
8. Give name of depositors having amount greater than 4000.
9. Give name of customers who opened account after date '1-12-96'.</td><td>Unit-3</td></tr>
<tr><td>3</td><td>Create Table Job (job_id, job_title, min_sal,
<table><tr><th>COLUMN NAME</th><th>DATA TYPE</th></tr><tr><td>job_id</td><td>Varchar2(15)</td></tr><tr><td>job_title</td><td>Varchar2(30)</td></tr><tr><td>min_sal</td><td>Number(7,2)</td></tr><tr><td>max_sal</td><td>Number(7,2)</td></tr></table>
Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no)
<table><tr><th>COLUMN NAME</th><th>DATA TYPE</th></tr><tr><td>emp_no</td><td>Number(3)</td></tr><tr><td>emp_name</td><td>Varchar2(30)</td></tr><tr><td>emp_sal</td><td>Number(8,2)</td></tr><tr><td>emp_comm</td><td>Number(6,1)</td></tr><tr><td>dept_no</td><td>Number(3)</td></tr></table>
Create table deposit(a_no,cname,bname,amount,a_date).
<table><tr><th>COLUMN NAME</th><th>DATA TYPE</th></tr><tr><td>a_no</td><td>Varchar2(5)</td></tr><tr><td>cname</td><td>Varchar2(15)</td></tr><tr><td>bname</td><td>Varchar2(10)</td></tr><tr><td>amount</td><td>Number(7,2)</td></tr><tr><td>a_date</td><td>Date</td></tr></table>
Create table borrow(loanno,cname,bname,amount).
<table><tr><th>COLUMN NAME</th><th>DATA TYPE</th></tr><tr><td>loanno</td><td>Varchar2(5)</td></tr><tr><td>cname</td><td>Varchar2(15)</td></tr><tr><td>bname</td><td>Varchar2(10)</td></tr><tr><td>amount</td><td>Varchar2(7,2)</td></tr></table>
Retrieve all data from employee, jobs and deposit.
1. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.
2. Display all jobs with minimum salary is greater than 4000.
3. Display name and salary of employee whose department no is 20. Give alias name to name of employee.
4. Display employee no,name and department details of those employee whose department lies in(10,20)

To study various options of <u>LIKE</u> predicate
1. Display all employee whose name start with ‘A’ and third character is ‘ ‘a’.
2. Display name, number and salary of those employees whose name is 5 characters long and first three characters are ‘Ani’.</td><td>Unit-7</td></tr>
</table>

	3. Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long. 4. Display the null values of employee and also employee name's third character should be 'a'. What will be output if you are giving LIKE predicate as '%_%' ESCAPE '\'	
4	To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables in Practical 3 1. List total deposit from deposit. 2. List total loan from karolbagh branch 3. Give maximum loan from branch vice. 4. Count total number of customers 5. Count total number of customer's cities. 6. Create table supplier from employee with all the columns. 7. Create table sup1 from employee with first two columns. 8. Create table sup2 from employee with no data 9. Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field. 10. Delete all the rows from sup1. 11. Delete the detail of supplier whose sup_no is 103. 12. Rename the table sup2. 13. Destroy table sup1 with all the data. 14. Update the value dept_no to 10 where second character of emp. name is 'm'. Update the value of employee name whose employee number is 103.	Unit-3, 4 & 7
5	To study Single-row functions: 1. Write a query to display the current date. Label the column Date 2. For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary 3. Modify your query no (2) to add a column that subtracts the old salary from the new salary. Label the column Increase 4. Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names. 5. Write a query that produces the following for each employee: <employee last name> earns <salary> monthly 6. Display the name, hire date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday. 7. Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM. 8. Write a query to calculate the annual compensation of all employees (sal+comm.).	Unit-4 & 7
6	Displaying data from Multiple Tables (join) 1. Give details of customers ANIL. 2. Give name of customer who are borrowers and depositors and having living city nagpur 3. Give city as their city name of customers having same living branch. 4. Write a query to display the last name, department number, and department name for all employees. 5. Create a unique listing of all jobs that are in department 30. Include the location of the department in the output 6. Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK. 7. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Create a query to display the name and hire date of any employee hired after employee SCOTT.	Unit-4 & 7
7	To apply the concept of Aggregating Data using Group functions. 1. List total deposit of customer having account date after 1-jan-96. 2. List total deposit of customers living in city Nagpur. 3. List maximum deposit of customers living in bombay. 4. Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. 5. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE. 6. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998 7. Find the average salaries for each department without displaying the respective department numbers. 8. Write a query to display the total salary being paid to each job title, within each department. 9. Find the average salaries > 2000 for each department without displaying the respective department numbers. 10. Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary. 11. List the branches having sum of deposit more than 5000 and located in city bombay.	Unit-4 & 7
8	To solve queries using the concept of sub query 1. Write a query to display the last name and hire date of any employee in the same department as SCOTT. Exclude SCOTT 2. Give name of customers who are depositors having same branch city of mr. sunil. 3. Give deposit details and loan details of customer in same city where pramod is living. 4. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary. 5. Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000 6. Display the last name and salary of every employee who reports to ford. 7. Display the department number, name, and job for every employee in the Accounting department. 8. List the name of branch having highest number of depositors. 9. Give the name of cities where in which the maximum numbers of branches are located. 10. Give name of customers living in same city where maximum depositors are located.	Unit-4 & 7
9	Manipulating Data: 1. Give 10% interest to all depositors. 2. Give 10% interest to all depositors having branch vrce	Unit-4 & 7

	3. Give 10% interest to all depositors living in nagpur and having branch city bombay. 4. Write a query which changes the department number of all employees with empno 7788's job to employee 7844'current department number. 5. Transfer 10 Rs from account of anil to sunil if both are having same branch. 6. Give 100 Rs more to all depositors if they are maximum depositors in their respective branch. 7. Delete depositors of branches having number of customers between 1 to 3. 8. Delete deposit of vijay. 9. Delete borrower of branches having average loan less than 1000.	
10	Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) write a cursor to select the five highest paid employees from the table.	Unit-9

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	MySQL or Oracle or PostgreSQL or SQL Lite

Course Outcome	
	Upon completion of the course students will be able to
CO1	To learn the fundamentals of data models and to represent a database system using ER diagrams with the mapping of Relational model to demonstrate competence with the fundamental tasks involved with modeling, designing
CO2	To Apply the SQL commands for database manipulation, and to identify and solve the redundancy problem in database tables using normalization
CO3	To Analyze transaction processing, concurrency control and database recovery protocols and To Compare and contrast various indexing strategies in different database systems.
CO4	To Apply the triggers, Functions/Procedures, Cursors and views in database schema to automate the real time problems with Optimized Queries.
Suggested Reference Books	
1	Database System Concepts, Abraham Silberschatz, Henry F. Korth & S. Sudarshan, McGraw Hill
2	An introduction to Database Systems, C J Date, Addition-Wesley
3	SQL,PL/SQL the Programming Language of oracle, Ivan Bayross, BPB Publications
4	Fundamentals of Database Systems, R. Elmasri and S.B. Navathe, the Benjamin / Cumming Pub. Co
5	Oracle 9i: PL/SQL Programming ,Scott Urman,Oracle press, Addison Wesley
6	Fundamentals of Database Systems, Ramez Elmasri and Shamkant B Navathe
7	Oracle: The Complete Reference, George Koch, Kevin Loney, TMH /oracle press
8	Mastering SQL, Martin Gruber, B.P.B

List of Open Source Software/Learning website	
1	https://www.w3schools.com/
2	https://www.mysql.com/
3	https://www.tutorialspoint.com/index.htm

Practical Project/Hands on Project		
Sr. No.	**Project List**	**Linked with Unit**
1	**For the following relation schema:** **employee (employee-name, street, city)** **works (employee-name, company-name, salary)** **company (company-name, city)** **manages (employee-name, manager-name)** Give an expression in SQL for each of the following queries: (1) Find the names, street address, and cities of residence for all employees who work for 'City Bank' and earn more than 50,000Rs. (2) Find the names of all employees in the database who live in the same cities as the companies for which they work. (3) Find the names of all employees in the database who live in the same cities and on the same streets as do their managers. (4) Find the names of all employees in the database who do not work for 'City Bank'. Assume that all people work for exactly one company. (5) Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'. Assume that all people work for at most one company. (6) Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.	Unit – 3,4,7,9
2	**EmployeeDetails (EmpId, Fullname, ManagerId, DateOfJoining)** **EmployeeSalary (EmpId,ProjectName,Salary)** (1) SQL query to fetch all employee records from Employee Details table who have a salary record in Employee Salary table. (2) Write a SQL query to fetch project-wise count of employees sorted by project's count in descending order. (3) SQL query to create a new table named Person with data and structure copied from table EmployeeDetails. (4) SQL query to fetch records that exist in EmployeeDetails and Person tables. (5) SQL query to fetch employee names having salary greater than or equal to 5000 and less than or equal 10000. (6) SQL query to find work experience in days till today.	Unit –3,4,7,9

<table>
<tr>
<td>3</td>
<td>(1) Create above table named student by giving id primary key and name with not null constraint and marks must between 1 to 100.
(2) Update name of student whose std_id is 1. Aslo print how many rows updated after update query using cursor in PL/SQL.
(3) Display name of all student with their total mark.
(4) Add one column named subject in student table and drop constraint not null from name column.
(5) Display name of student with stu_id whose marks are greater than average marks of all students.
<table>
<tr><th>id</th><th>std_id</th><th>Name</th><th>Marks</th></tr>
<tr><td>1</td><td>3</td><td>Abhi</td><td>99</td></tr>
<tr><td>2</td><td>5</td><td>Geethasri</td><td>89</td></tr>
<tr><td>3</td><td>6</td><td>Rahim</td><td>49</td></tr>
<tr><td>4</td><td>9</td><td>Ram</td><td>69</td></tr>
<tr><td>5</td><td>1</td><td>Rahul</td><td>87</td></tr>
<tr><td>6</td><td>1</td><td>Rahul</td><td>96</td></tr>
<tr><td>7</td><td>1</td><td>Rahul</td><td>96</td></tr>
<tr><td>8</td><td>9</td><td>Ram</td><td>96</td></tr>
<tr><td>9</td><td>9</td><td>Ram</td><td>96</td></tr>
</table>
</td>
<td>Unit –3,4,7,9</td>
</tr>
<tr>
<td>4</td>
<td>Worker (WORKER_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPARTMENT)
Bonus (WORKER_REF_ID, BONUS_DATE, BONUS_AMOUNT)
Title (WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM)
(1) Write an SQL Query to Print All Worker Details From The Worker Table Order By FIRST_NAME Ascending And DEPARTMENT Descending.
(2) Write an SQL Query to Fetch the Count of Employees Working In The Department ‘Admin’.
(3) Write an SQL Query to Fetch First Name of Work with Salaries >= 50000 And <= 100000.
(4) Write an SQL Query to Print Details for Workers with The First Name As “Vipul” And “Satish” From Worker Table.
(5) Write an SQL Query to Print Details of The Workers Whose FIRST_NAME Contains ‘A’ or ‘a’.
(6) Write an SQL Query to Print Details of The Workers Who Are Also Managers.
(7) Write an SQL Query to Fetch the List of Employees with The Same Salary.</td>
<td>Unit –3,4,7,9</td>
</tr>
<tr>
<td>5</td>
<td>Write queries for the following tables.
Employee(Empno, Ename , Salary, Designation),
Department(Empno, Deptno.)

(1) Display all rows for salary greater than 5000.
(2) Display the deptno for the ename=’Rahul’.
(3) Add a new column deptname in table T2.
(4) Change the designation of ename=’ramesh’ from ‘peon’ to ‘senior clerk’.
(5) Find the total salary of all the rows.
(6) Display designation wise maximum salary given to employee.</td>
<td>Unit –3,4,7,9</td>
</tr>
</table>

LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of CE/IT/CSD/AIML/AIDS/ CSE/CST/CS&IT/CEA/MA&CP/CSE(AI)/ CSE(DS)/CSE(CS)

(701,702,703,704,705,712,713,714,715,716,717,718,719)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**117013292/117023292/117033292/117043292/117053292/117123292/117133292/117143292/117153292/017163292/017173292/017183292/017193292**
Course Name:	**Data Structures using Java**
Category of Course:	Professional Core Course (PCC)
Prerequisite Course:	Computer Programming using Java – I (117012191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**0**	**6**	**6**	**30**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Introduction to Data Structure**			**2 (7%)**
	1.1 Data Types – Primitive and Non-Primitive	Computer Programming using Java – I (117012191 Unit-2.4)		
	1.2 Types of Data Structures- Linear & Non-Linear Data Structures	---		
	1.3 Memory Representation of Array: Row Major and Column Major	Arrays (117012191 - Unit-5.1, 5.2)		
02	**Stack-1**			**3 (10%)**
	2.1 Stack-Definitions & Concepts, Operations on Stacks (Push, Pop, Peep, Change)	Arrays (117012191 - Unit-5.1, 5.2)		
	2.2 Applications of Stacks: Tower Hanoi Using recursion, Reverse the String using Stack, Decimal to Binary conversion using Stack	Stack-1(117013292-Unit-2.1)	Recursively Enumerable Languages (017013492-Unit-10.1)	
03	**Stack-2**			**4 (13%)**
	3.1 Applications of Stacks: Expression Conversion (Infix, Polish, Reverse-Polish)	Stack-1(117013292-Unit-2.1)		
	3.2 Applications of Stacks: Expression Evaluation and Implementation (Prefix, Postfix)	Stack-1(117013292-Unit-2.1)		
04	**Queue-1**			**3 (10%)**
	4.1 Queue - Definitions & Concepts, Representation of Queue, Operations on Simple Queue (Enqueue, Dequeue, Display)	Arrays (117012191 - Unit-5.1, 5.2)	Process Management (017013301 -Unit-2.1,2.2,2.3)	
	4.2 Operations on Circular Queue (Enqueue, Dequeue, Display)	Queue-1(117013292 -Unit-4.1)		
05	**Queue-2**			**2 (7%)**
	5.1 Priority Queue - Definition and Introduction and Circular Double Ended Queue – Introduction, Types (Input restricted, Output restricted), Operations on Circular double ended queue – Insert at Front, Insert at Rear, Delete at Front, Delete at Rear	Queue-1(117013292 -Unit-4.1)		
	5.2 Applications of Queue	---		
06	**Singly Linked List**			**3 (10%)**
	6.1 Dynamic Memory Allocation, Memory Representation of Linked List, Availability Stack, Types of Linked List	Object Fundamentals (117012191 - Unit-7.1)	File System (017013301 - Unit-8.2)	
	6.2 Operations on Singly Linked List (Insert, Delete, Traverse)	---		
	6.3 Implementation of Stack and Queue using Singly Linked List	Stack-1(117013292-Unit-2.1), Queue-1(117013292-Unit-4.1), Singly Linked List (117013292-Unit-6.2)		
07	**Doubly and Circular Link List**			**3 (10%)**
	7.1 Operations on Doubly Linked List (Insert, Delete, Traverse)	Singly Linked List (117013292-Unit-6.2)		
	7.2 Operations on Circular Linked List (Insert, Delete, Traverse)	Singly Linked List (117013292-Unit-6.2)		
	7.3 Applications of Linked List	---		
08	**Tree**			**4 (13%)**
	8.1 Tree-Definitions & Concepts - Root, Parent, Child, Leaf, Ancestor of Node, Descendant of Node, Siblings, Level of Node, Internal Node, Height of Node and Tree, Depth of Node and Tree, Subtree, Neighbor of a Nodes, Types of Tree (General Tree, Binary Tree, Balanced Tree, Full Binary Tree, Complete Binary Tree, Perfect Binary Tree, Skewed Tree, Degenerate or pathological Tree	---	Sorting Techniques (017013591 -Unit-2.4)	

	Representation of Binary Tree using arrays and LinkedList, Binary Tree Traversal (Inorder, Postorder, Preorder), Construction of Binary Tree from Traversal	Tree (117013292-Unit-8.1)		
	8.3 Conversion from General Tree to Binary Tree, Threaded Binary Tree and its types - Left Threaded, Right Threaded, Full Threaded Binary Tree, Expression Tree	Tree (117013292-Unit-8.2)		
	8.4 Introduction to Heap and its Types - Min Heap and Max Heap, Operations on heap - Insertion, Deletion	Tree (117013292-Unit-8.2)	Sorting Techniques (017013591 -Unit-2.4)	
09	**Applications of Trees**			**4 (13%)**
	9.1 Binary Search Tree: Definition and Implementation, Operations: Search, Insertion, Min, Max, Successor, Predecessor, Deletion	Tree (117013292-Unit-8.2)		
	9.2 Introduction to Height Balanced BSTs: AVL and Balance Mechanism, Operations on AVL trees - Insertion, Deletions and Rotations	Tree (117013292-Unit-8.1)		
	9.3 Multi-way Search Tree: 2-3 Trees (Order 3 B Tree), B and B+ tree , Insertion Operation	Tree (117013292-Unit-8.1)	Indexing (017013291 -Unit-7.2)	
10	**Hashing**			**2 (7%)**
	10.1 Hashing: Introduction: Hashing, Symbol Table, Hashing Functions by using division method	---	File System (017013301 - Unit-8.2)	
	10.2 Collision Resolution Techniques - Chaining, Linear Probing, Quadratic probing, Double hashing	---		
	10.3 Applications of Hashing	---		

Sr No.	Practical Title	Link to Theory Syllabus
1	Write a program to calculate the address of a 1- D array.	Unit-1
2	Write a program to calculate the address of array in 2 – D array. Where the type of an array – Row major order or column major order – base address and the index where we want to find the address is given as inputs.	Unit-1
3	Write a program to implement stack operations using an array (Operations: push, pop, peep, change, Display)	Unit-2
4	Write a program to reverse the string using stack.	Unit-2
5	Write a program to evaluate the given postfix expression.	Unit-3
6	Write a program to implement simple queue insertion and deletion using an array	Unit-4
7	Write a program to implement circular queue insertion and deletion using an array	Unit-4
8	Write a program to implement DEQUEUE to insert a node using an array	Unit-5
9	Write a program to implement DEQUEUE to delete a node using an array	Unit-5
10	Write a program to create a singly linked list of three nodes where nodes are inserted at the front.	Unit-6
11	Write a program to create a singly linked list of three nodes where nodes are inserted at the last.	Unit-6
12	Write a program to delete a node at particular position from a given singly linked list.	Unit-6
13	Write a program to implement insertion in circular Linked list	Unit-7
14	Write a program to implement deletion in circular Linked list	Unit-7
15	Write a program to implement insertion in doubly Linked list	Unit-7
16	Write a program to implement deletion in doubly Linked list	Unit-7
17	Write a Program to Find the Height of a Binary Tree	Unit-8
18	Write a Program to Implement in – order, pre – order and post – order traversal in a given Binary Tree.	Unit-8

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	Notepad++, VS Code

Course Outcome	
	Upon completion of the course students will be able to
CO1	Understand the concept of memory management, basic data structures with its types and fundamental data structure stack to develop problem-solving skills based on stack that is applicable across various computational domains.
CO2	Understand and utilize queue and LinkedList data structures, including various types and its operations, enhancing their problem-solving skills across diverse computational scenarios. Understand the concept of static versus dynamic data structures.
CO3	Master diverse LinkedList and tree data structures with its operations. Through practical applications, students will understand the versatility of linked lists and trees in different complex problems
CO4	Possess advanced skills in managing different Tree data structure and its operations. Understand the importance of trees in searching. Furthermore, students will grasp the concepts of hashing along with their practical applications
Suggested Reference Books	
1	An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.
2	Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher Thomson Learning.

List of Open Source Software/Learning website	
1	www.geeksforgeeks.org
2	www.tutorialspoint.com
3	www.programiz.com
4	https://visualgo.net/en

Practical Project/Hands On Project

Sr. No.	List of Practical Projects	Linked with Unit
1	Hotel management system : Listed below are some of the important functions **dashboard()** – This function displays the menu or welcome screen to perform different hotel booking activities mentioned below. **new_acc()** – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, citizenship number, address and phone number. **room_type()** – This function allows the user to select the categories of the room ie normal or executive with the option of Ac room or non ac room. **check_availability()** – This functionality allows the user to check the number of room vacant prior booking. **book_room()** – This function allows the user to book the selected room. **search_facilities()** – With this function, if the user selects the executive room than user can search for the extra facilities provided like games, swimming, food service in rooms while booking. **payment()** – This function allows making payment of booked room based on number of days the room is occupied via online method option or at the checkout time.	Unit 2,3,4,6
2	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 3 followings: **free**: indicated by integer value 0 **occupied**: indicated by integer value 1 **inaccessible**: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from user and initialize the array. Raise and handle user defined exception if invalid status value given.	Unit 2,3,4,6,8
3	Create an application that performs the following task associated with the files : 1) Eliminating repeated lines from the files. 2) Reverse the content of file and store in another file. 3) Remove the lines starting from any prefix. 4) Obtain the line number where the particular word is present. 5) Obtaining number of words, characters, white spaces and lines present in that particular file.	Unit 2,3,4,5,6
4	Implement calculator functionality.	All
5	Write a program to implement Quadratic equation.	All

LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of CE/IT/CSD/AIML/AIDS/CSE/CST/CS&IT/CEA/MA&CP/CSE(AI)/ CSE(DS)/CSE(CS)

(701,702,703,704,705,712,713,714,715,716,717,718,719)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**117012291/117022291/117032291/117042291/117052291/117122291/117132291/117142291/117152291/017162291/017172291/017182291/017192291**
Course Name:	**Fundamental of Electronics and Electrical Engineering**
Category of Course:	Engineering Science Course (ESC)
Prerequisite Course:	---

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**0**	**2**	**4**	**30**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**DC Circuits**			**3 (10%)**
	1.1 Electrical circuit elements (R, L and C), Voltage and current Sources	---	---	
	1.2 Ohm's law, Series and parallel resistive circuit with voltage & current divider rules	---	---	
	1.3 Kirchhoff's current and voltage laws	Ohm's Law (117012291-Unit-1.2)	---	
	1.4 Charging and discharging of capacitor	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
02	**Network Theorems**			**3 (10%)**
	2.1 Thevenin and Norton Theorems	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
	2.2 Superposition Theorem and Source Transformation	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
	2.3 Nodal and Mesh Analysis	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
03	**Single Phase AC Circuits**			**3 (10%)**
	3.1 Generation of Single Phase, Representation of Sinusoidal Waveforms	---	---	
	3.2 RMS, Average Values and Peak Values, Form Factor and Peak Factor	---	---	
	3.3 Phasor Representation of AC Quantities	Generation of Single Phase (117012291-Unit-3.1)	---	
04	**Analysis of Single-Phase AC Circuits**			**3 (10%)**
	4.1 Analysis of Single-Phase AC Circuits consisting of R, L and C with Power Measurement	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
	4.2 Analysis of Single-Phase Series AC Circuits consisting of RL, RC and RLC with Power Measurement	Ohm's Law (117012291-Unit-1.2), Kirchhoff's current and voltage laws (117012291-Unit-1.3)	---	
	4.3 Series RLC AC Circuit at Resonance	Analysis of Single-Phase Series RLC Circuit (117012291-Unit-4.2)	---	
05	**Three Phase AC Circuits**			**3 (10%)**
	5.1 Generation of three phase E.M. F	Generation of Single Phase (117012291-Unit-3.1)		
	5.2 Voltage and Current Relations in 'STAR' Three Phase AC Circuit	---	---	
	5.3 Voltage and Current Relations in 'DELTA' Three Phase AC Circuit	Three Phase 'STAR'AC Circuit (117012291-Unit-5.1)	---	
	5.4 Power Measurements in Three Phase AC Circuits	Three Phase 'STAR'AC Circuit (117012291-Unit-5.1)	---	
06	**Transformers**			**3 (10%)**
	6.1 Faraday's Law of Electromagnetic Induction	---	---	
	6.2 Construction of transformer, Types, Working Principle of Transformer Operations, E.M.F equation	Faraday's Law (117012291-Unit-6.1)	---	
	6.3 Single Phase Step-Up and Step-Down Transformers	Working Principle of Transformer Operations (117012291-Unit-6.2)	---	
	6.4 Three Phase Transformers	---	---	
07	**Electrical Machines**			**3**

	7.1 Three Phase Induction Motor, Generation of Rotating magnetic field	Faraday's Law (117012291-Unit-6.1)	---	**(10%)**
	7.2 Single Phase Induction Motor	Faraday's Law (117012291-Unit-6.1)	---	
	7.3 DC Motors-Construction, Working & Types	Faraday's Law (117012291-Unit-6.1)	---	
08	**Diode Circuits**			**3 (10%)**
	8.1 Basic idea about forward bias, reverse bias of Diode and VI characteristics	---	---	
	8.2 Half wave rectifier, Full wave rectifier	---	---	
	8.3 Bridge rectifier	---	---	
09	**Bipolar junction transistors**			**3 (10%)**
	9.1 BJT operation, BJT voltages and currents	Basic idea about diodes (117012291-Unit-8.1)	---	
	9.2 CE characteristics	BJT operation (117012291-Unit-9.1)	---	
	9.3 Transistor as a switch	CE characteristics (117012291-Unit-9.2)	---	
10	**Transistor Biasing**			**3 (10%)**
	10.1 DC Load Line Concepts & Q Point Stabilization	CE characteristics (117012291-Unit-9.2)	---	
	10.2 Fixed Bias	CE characteristics (117012291-Unit-9.2)	---	
	10.3 Collector to Base Bias	CE characteristics (117012291-Unit-9.2)	---	
	10.3 Voltage Divider Bias	CE characteristics (117012291-Unit-9.2)	---	

Sr No.	Practical Title	Link to Theory Syllabus
1	Verify KVL and KCL using Development kit.	Unit-1
2	To verify the Thevenin Theorem	Unit-2
3	To verify the Superposition Theorem	Unit-2
4	Measurement of the electric power in a single-phase AC Resistive Circuit.	Unit-4,5
5	To obtain power & power factor of single-phase R – L Series circuits	Unit-4,5
6	To obtain power & power factor of single-phase R – L - C Series circuits	Unit-4,5
7	To plot input and output waveforms of the Half Wave Rectifier.	Unit-8
8	To plot input and output waveforms of the Bridge Rectifier.	Unit-8
9	To study the input and output characteristics of NPN transistor in Common Emitter mode.	Unit-9

Major Components/ Equipment

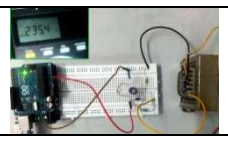
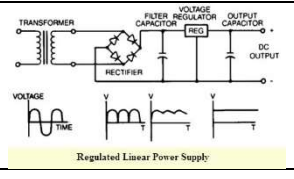
Sr. No.	Component/Equipment
1	DC Network Development Kit, Voltmeter, Ammeter, Connecting Wires
2	DC Network Development Kit, Voltmeter, Ammeter, Connecting Wires
3	DC Network Development Kit, Voltmeter, Ammeter, Connecting Wires
4	Ammeter (0-5 amp), Voltmeter (0-300 volt), Wattmeter (5-amp, 300-volt, 1500 watt), Multimeter, Lamp-bank (non-inductive resistance) (230V, amp), Single-phase variac
5	Ammeter (0-5 amp), Voltmeter (0-300 volt), Wattmeter (5-amp, 300-volt, 1500 watt), Multimeter, Inductive coil (50 Hz, 5 amp), Lamp-bank (non-inductive resistance) (230V,5 amp), Single-phase variac
6	Ammeter (0-5 amp) Voltmeter (0-300 volt), Wattmeter (5-amp, 300-volt, 1500 watt), Multimeter, Single-phase variac, Lamp-bank (non-inductive resistance) (230V, amp), Choke coil, Capacitor bank
7	Trainer Kit, DC Power Supply, Function Generator, Connecting Wires, DSO.
8	Trainer Kit, DC Power Supply, Function Generator, Connecting Wires, DSO.
9	Trainer Kit, DC Power Supply, Function Generator, Connecting Wires, Multimeter.

Course Outcome	
	Upon completion of the course students will be able to
CO1	Apply fundamental circuit principles in a methodical way that is appropriate for analysis and design and then use these network theorems to further analyze the electric circuit.
CO2	Identify and analyze the waveforms and phasor diagrams for single phase AC circuits.
CO3	Summarize the working principles and uses of electrical machines, both rotational and static and recognize how rotating magnetic fields are created.
CO4	Comprehend the numerous semiconductor devices, their distinctive features and provide examples of how transistors operate, along with transistor biasing. Also, this will enable students to make contributions to their understanding of system implementation and computer hardware design.

Suggested Reference Books	
1	B. L. Theraja, 'A Textbook of Electrical Technology', S. Chand Publication-Volume I.
2	J.B. Gupta, Basic Electrical Engineering, Kataria & Sons -Volume I.
3	Charles Alexander and Matthew Sadiku, "Fundamentals of Electric Circuits", McGraw Hill.
4	Edward Hughes, Harlow, "electrical & electronic technology", Pearson Education Limited.
5	U. A. Patel, '*Elements of Electrical Engineering*', Atul Prakashan.
6	Albert Malvino & David, "Electronic Principles", Tata McGraw-Hill, Seventh edition.
7	R. L. Boylestad and L. Nashelsky, "Electronic Devices and Circuit Theory", Pearson Education.
8	David A. Bell, "Electronic Devices and Circuits", Oxford University Press, Fifth edition.

List of Open Source Software/Learning website	
1	http://nptel.ac.in
2	https://www.electronicscoach.com
3	https://www.electrical4u.com

Practical Project/Hands on Project

Sr. No.	Project List	Linked with Unit
1	Case study: Verify KVL and KCL for given premises.	Unit 01
2	Identify the values of various passive components (R, L and C) for given circuit board.	Unit 02
3	AC voltage measuring Device using Arduino. 	Unit 04
4	Calculation of a Number of turns and voltage level for a given center tapped Transformer.	Unit 06
5	Design D.C. power supply for mobile.  Regulated Linear Power Supply	Unit 08
6	Dancing LED circuit can be used for any visual sign indication in any highways.	Unit 08
7	Security Alarm system for theft detection.	Unit 09
8	Design automatic ignition electric circuit using bread board.	All unit

LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of CE/IT/CSD/AIML/AIDS/CSE/CST/CS&IT/CEA/MA&CP/CSE(AI)/ CSE(DS)/CSE(CS)

(701,702,703,704,705,712,713,714,715,716,717,718,719)

Bachelor of Engineering (B.E.) – Semester – II

Course Code:	**117012292/117022292/117032292/117042292/117052292/117122292/117132292/117142292/117152292/017162292/017172292/017182292/017192292**
Course Name:	**Computer Programming using Java-2**
Category of Course:	Engineering Science Course (ESC)
Prerequisite Course:	Computer Programming using Java-1 (117012191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
3	**0**	**6**	**6**	**30**

Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Abstraction & Interface with Runtime Polymorphism**			**2 (7%)**
	1.1 Dynamic method dispatch, Object casting and instance of operator	Inheritance (117012191 – Unit - 10)	---	
	1.2 Abstract class, Abstract Method		---	
	1.3 Interface: Implementation of Interface(Partial & Full) Extend Interface		---	
02	**Introduction to Package**			**2 (7%)**
	2.1 Use of Package, Import statement, Access Modifiers	Inheritance (117012191 – Unit - 10)	---	
	2.2 Access control with example		---	
03	**Exception handling**			**3 (10%)**
	3.1 Types of Errors, checked and unchecked Exception	Abstraction & Interface (117012292 – Unit - 01)	---	
	3.2 Exception handling mechanism: Use of try, catch, throw, throws and finally		---	
	3.3 Built in Exception, Custom Exception		---	
04	**Multithreaded Programming**			**4 (13%)**
	4.1 Introduction about Thread	Abstraction & Interface (117012292 – Unit - 01) Exception Handling (117012292 – Unit - 03)	---	
	4.2 Thread life cycle		---	
	4.3 Thread class and Runnable interface		---	
	4.4 Thread methods: start(), run(), getName(), setName(), sleep(), join(), isAlive(), wait(), notify(), currentThread()		---	
	4.5 Synchronized Methods and Synchronized Blocks, Producer – Consumer Problem solution using wait() & notify()		---	
05	**Collection-Part 1**			**3 (10%)**
	5.1 What is collection and need of it. Collection framework Hierarchy, Classes and interfaces in collections, Methods of Collection interface: add(), addAlll(), clear(), contains(), isEmpty(), iterator(), remove(), removeAll(), toArray().	---	---	
	5.2 **List Interface:** **ArrayList**: add(int index, E element), add(E e), clear(), ensureCapacity(int requiredCapacity), get(), set(), isEmpty(), lastIndexOf(Object o), remove(int index), sort(),size(), for each loop to print all elements indexOf(), lastIndexOf() Add Multiple element with Arrays.asList() in Constructor. **Linked List:** Linked list Creation, add(),add(int index, E element), addAll(), addFirst(), addLast(), clear(), contains(), getFirst(), getLast(), remove(), remove(int index), user iterator to print linked list elements **Collections class :** frequency(), reverse(), max(), min(), sort(), Comparator.comparing() to sort class objects by properties.	---	---	
	5.3 **List Interface** **Vector**: create vector, add(), add(index,element), capacity(), clear(), clone(), elementAt(int index), equals(Object o), isEmpty(), remove() **Stack**: Creating a Stack, push(E item), pop(),peek(), search(Object o), empty(), fetch value using iterator()	---	---	

06	**Collection-Part 2**			**3 (10%)**
	6.1 Queue Interface: **Priority Queue** class : add(object), offer(object), remove(), poll(), element(), peek(): Print Queue using iterator Dequeue Interface: **ArrayDeque** class: ArrayDeque creation, add(Element e), addAll(Collection<? extends E> c), addFirst(Element e), addLast(Element e), clear(), getFirst(), getLast(), isEmpty(), offerFirst(Element e), offerLast(Element e), peek(), remove(), removeFirst(), removeLast(), size()	---	---	
	6.2 Set Interface: **HashSet**: add(), clear(), remove(), isEmpty(), size(), removeAll(), addAll(), equals(), print using iterator Map Interface: **HashMap**: create HashMap, size(), isEmpty(), remove(), put(), putAll(), getKey(), getValue(),print using iterator or foreach **HashTable:** put(), remove(), containsKey(), clear(),getKey(), getValue(),print using iterator or foreach	---	---	
07	**IO Programming**			**3 (10%)**
	7.1 Introduction to Stream, Byte Stream, Character stream	Constructor (117012191 – Unit - 08)	---	
	7.2 File Class and its method, constructor of File Class, methods like : canExecute(), canRead(), createNewFile(), equals(), exists(), getAbsolutePath(), getName(), getParent(), getParentFile(), getPath(), isDirectory(), isFile(), length(), listFiles(), mkdir(), list().			
	7.3 File Input Stream, File Output Stream			
08	**Character Stream**			**3 (10%)**
	8.1 Readers and Writers class, FileReader, FileWriter	IO Programming (117012292 – Unit - 07)	---	
	8.2 Buffered Reader, InputStreamReader,			
	8.3 RandomAccessFile with constructor and methods like: close(), readInt(), readUTF(), seek(), writeDouble(), writeFloat(), write(), read(), length(), getFilePointer()			
09	**JDBC Part-1**			**4 (13%)**
	9.1 JDBC Architecture	----	---	
	9.2 JDBC Drivers, Steps to connect to Database, Connectivity with MySQL, DriverManager, Connection	Abstraction & Interface (117012292 – Unit - 01) Exception Handling (117012292 – Unit - 03)	---	
	9.3 Types of JDBC statements: Statement, Prepared statement, Callable statement	Iterator (117012292 – Unit - 05)	---	
10	**JDBC Part-2**			**3 (10%)**
	10.1 Database Metadata, Resultset Metadata	JDBC Part-1 (117012292 – Unit - 09) File, File Handling (117012292 – Unit – 07 & 08)	---	
	10.2 Storing image, Retrieving image, Storing file, Retrieving file, Stored procedures, and functions, Transaction Management(commit(). rollback(), setAutoCommit())		---	

Sr No.	Practical Title	Link to Theory Syllabus
1	Write an application that generates custom exception if any value from its command line arguments is negative.	Unit – 3
2	Write a method for computing xy by doing repetitive multiplication. x and y are of type integer and are to be given as command line arguments. Raise and handle exception(s) for invalid values of x and y. Also define method main. Use finally in above program and explain its usage.	Unit – 3
3	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 2followings:free: indicated by integer value 0 occupied: indicated by integer value 1 inaccessible: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from command line arguments and initialize the array. Raise and handle user defined exception if invalid status value given	Unit – 3
4	Write a complete program to accept N integer numbers from the command line. Raise and handle exceptions for following cases : - when a number is –ve - when a number is evenly divisible by 10 - when a number is greater than 1000 and less than 2000 - when a number is greater than 7000 Skip the number if an exception is raised for it, otherwise add it to find total sum	Unit – 3
5	Declare a class called book having author_name as private data member. Extend book class to have two sub classes called book_publication&paper_publication. Each of these classes have private member called title. Write a complete program to show usage of dynamic method dispatch (dynamic polymorphism) to display book or paper publications of given author. Use command line arguments for inputting data. .	Unit – 1
6	Write a program that reads file name from user, through command line argument and displays/reads content of the text file on console.	Unit – 6
7	Write a program that reads file name from user, through command line argument and displays/reads content of the text file on console.	Unit – 6
8	Write a program to replace all “word1” by “word2” from a file1, and output is written to file2 file and display the no. of replacement.	Unit – 6
9	Write a program that counts the no. of words in a text file. The file name is passed as a command line argument. The program should check whether the file exists or not. The words in the file are separated by white space characters.	Unit – 7
10	Write a program to read the content of a file into a character array and write it into another file. Get names of the files from command line	Unit – 7
11	Read employee salary and calculate the income tax based on 10% of income and store it in tax.txt file for five different employees	Unit – 7
12	The abstract Vegetable class has three subclasses named Potato, Brinjal and Tomato. Write an application that demonstrates how to establish this class hierarchy. Declare one instance variable of type String that indicates the color of a vegetable. Create and display instances of these objects. Override the toString() method of Object to return a string with the name of the vegetable and its color	Unit – 1
13	Write a program that illustrates interface inheritance. Interface P is extended by P1 And P2. Interface P12 inherits from both P1 and P2.Each interface declares one constant and one method. Class Q implements P12.Instantiate Q and invokes each of its methods. Each method displays one of the constants	Unit – 1
14	The Transport interface declares a deliver() method. The abstract class Animal is the superclass of the Tiger, Camel, Deer and Donkey classes. The Transport interface is implemented by the Camel and Donkey classes. Write a test program that initialize an array of four Animal objects. If the object implements the Transport interface, the deliver() method is invoked. .	Unit – 1
15	Write a abstract class named Person and its two subclasses named student and Employee. A person has a name, address, phone number and email address. A student has enrollment course. An Employee has an office, salary, and designation. Define constructors and methods for input and display for both classes. Define constructor and methods for input and display for both classes. Write a main program to give demonstration of all.	Unit – 1
16	Write a complete multi-threaded program to meet following requirements: - Read matrix [A] m x n - Create m number of threads - Each thread computes summation of elements of one row, i.e. ith row of the matrix is processed by ith thread. Where 0 <= i< m. - Print the results	Unit – 4
17	Write an application that executes two threads. One thread displays "Good Morning" every 1000 milliseconds & another thread displays "Good Afternoon" every 3000 milliseconds. Create the threads by implementing the Runnable interface.	Unit – 4
18	Write a complete multi-threaded program to meet following requirements: o Two threads of same type are to be instantiated in the method main. o Each thread acts as a producer as well as a consumer. o A shared buffer can store only one integer information along with the source & destination of the information at a time. o The information produced is to be consumed by appropriate consumer. o Both producers produce information for both consumers. o Each thread produces 5 information	Unit – 4

19	Write a multithreaded program to print all odd positive numbers in ascending order up to n, where n is a positive integer number given as a command line argument. Instantiate requited number of threads, where each thread except the last, examines next 50 numbers and the last thread examines remaining numbers up to n.	Unit – 4
20	Write a complete multi threaded program to meet following requirements for producerconsumer threads: - Three threads – one producer and two consumers to be instantiated in the method main. - At a time, the producer produces one integer information along with consumer_id to represent id of a consumer that will consume produced information. - Information and consumer_id are stored in a shared buffer. - The information produced is to be consumed by appropriate consumer only, as specified by the producer. - The producer thread produces total 6 information	Unit – 4
21	Write a complete multi threaded program to meet following requirements for producerconsumer threads: - Three threads – one producer and two consumers to be instantiated in the method main. - At a time, the producer produces one integer information along with consumer_id to represent id of a consumer that will consume produced information. - Information and consumer_id are stored in a shared buffer. - The information produced is to be consumed by appropriate consumer only, as specified by the producer. - The producer thread produces total 6 information	Unit – 4
22	Consider Bank table with attributes AccountNo,CustomerName, Balance, Phone and Address. Write a database application which allows insertion, updation and deletion of records in Bank table. Print values of all customers whose balance is greater than 20,000.	Unit – 10
23	Write a program using JDBC for getting personal information – name, birthdate, sex, address, phone no, email-id & store it in database. Also provide list of all records, all male, all female & all minors (age below 18).	Unit – 10
24	Write Java application program to change the basic = basic + 500 of all the employees whose age is greater then 40 from employee table then display how many record updated	Unit – 10
25	Give the use of Statement, PreparedStatement and CallableStatement object and Write code to insert three records into student table using PreparedStatement (assume student table with Name, RollNo, and Branch field).	Unit – 9

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	JDK, JRE, VS CODE, PhpMyAdmin, My SQL

Course Outcome	
	Upon completion of the course students will be able to
CO1	Apply concept of abstraction, interface, packages and Exception handling to create error free code.
CO2	Implement multithreading in object programs, understand use of collection (ArrayList, LinkedList, Vector and Stack) in programs.
CO3	Understand different kind of file I/O programming, use of collection (Queue, set, Hash Map and Hash Table) in programs.
CO4	Apply the concepts of JDBC, Transection processing, statement objects and Resultset to perform operations on Database.
Suggested Reference Books	
1	Java: The Complete Reference, Tenth Edition (Complete Reference Series), Herbert Schildt – McGrawHill
2	Java Server Programming Java EE 7 (J2EE 1.7), Black Book Kindle Edition, Kogent Learning Solutions Inc – Dreamtech
3	Core Java Volume I--Fundamentals, 11th edition, Cay Horstman – Pearson
4	Core Java - An Integrated Approach Includes All Version Upto Java 8, Dr. R. Nageswara Rao – Dreamtech
5	Programming with Java by, E Balagurusamy – McGrawHill

List of Open Source Software/Learning website	
1	https://www.javatpoint.com/java-tutorial
2	https://www.tutorialspoint.com/java/index.htm
3	https://www.geeksforgeeks.org/java/
4	https://www.oracle.com/java/technologies/downloads/#jdk17-windows & https://notepad-plus-plus.org/downloads/
5	https://www.programiz.com/java-programming/