DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SCHEME OF TEACHING AND EXAMINATION (2017 – 2018)

B.E. (ISE) V SEMESTER

Sl. No	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UIS522C	Software Engineering	04	4	0	0	50	50	100
2	UIS503C	Database Management System	04	4	0	0	50	50	100
3	UIS505C	Java Programming	04	4	0	0	50	50	100
4	UIS506C	Operating Systems	04	4	0	0	50	50	100
5	UIS059E UIS051E	Information Security Advanced Graph Theory	04	4	0	0	50	50	100
6	UIS525L	System Software Laboratory	02	0	0	4	50	50	100
7	UIS510L	Java Programming Lab	1.5	0	0	3	50	50	100
8	UIS511L	Database Application Laboratory	1.5	0	0	3	50	50	100
		Total	**25**	**20**	**2**	**2**	**400**	**400**	**800**

UIS522C: Software Engineering
(4-0-0)(4 Credits, 52 Hours)

Course Objectives :

1) To have knowledge of basic software engineering methods and practices and their appropriate application.
2) To have general understanding of the software process models.
3) To develop and write a Software Requirements Specification.
4) To design and implement a software system and document it.
5) To conduct software testing.
6) To work with a team and understand team dynamics.
7) To be able to write a software project report.

Course Outcomes :

1) Demonstrate an understanding of emergence and fundamentals of software engineering and different software development lifecycle models, and software requirements.
2) Apply the software engineering techniques in the systems specification, design and development stages of software projects
3) Comprehend the importance of software coding and documentation
4) Describe and apply testing and quality management techniques for development of quality software.
5) Exhibit a good knowledge of fundamental principles of software project cost estimation, testing, staffing and management techniques.

Mapping (Co-relations) of CO's with PO's

Sl. No.	Course Outcomes	IS-PO-1	IS-PO-2	IS-PO-3	IS-PO-4	IS-PO-5	IS-PO-6	IS-PO-7	IS-PO-8	IS-PO-9	IS-PO-10	IS-PO-11	IS-PO-12
01	Demonstrate an understanding of emergence and fundamentals of software engineering and different software development lifecycle models, and software requirements.	2	1	1									
02	Apply the software engineering techniques in the systems specification, design	2	2	2		1				2	2		

	and development stages of software projects.												
03	Comprehend the importance of software coding and documentation.	2	2	2		1				2	2		
04	Describe and apply testing and quality management techniques for development of quality software.	2	2	2		1			1	2	2	2	
05	Exhibit a good knowledge of fundamental principles of software project cost estimation, testing,staffing and management techniques.	2	2			1			1	2	2	2	

Syllabus

Academic Year	**:**	**2018 - 19**
Semester	**:**	**V**
Subject	**:**	**Software Engineering**
Subject Code	**:**	**UIS522C**
Credits	:	04
Teaching Hours	:	52

Unit –I	**10 Hours**

INTRODUCTION: Evolution- from an art form to an engineering discipline, software development projects, exploratory style of software development, emergence of software engineering, notable changes in software development practices, computer systems engineering.

SOFTWARE LIFE CYCLE MODELS: A few basic concepts, waterfall model and its extensions, rapid application development, agile development models, spiral model, a comparison of different life cycle models

REQUIREMENT ANALYSIS AND SPECIFICATION: Requirement gathering and analysis, software requirements specification (SRS).

Unit –II	**10 Hours**

SOFTWARE DESIGN: Overview of the design process, how to characterize a good software design, cohesion and coupling, layered arrangement of Modules, approaches to software design

FUNCTION-ORIENTED SOFTWARE DESIGN: Overview of SA/SD methodology, structured analysis, developing the DFD model of the system, structured design, detailed design, design review

OBJECT MODELLING USING UML: Basic Object-orientation concepts, Unified Modeling Language, UML diagrams, Use case model,, Class diagrams, Interaction diagrams, Activity diagram, State chart Diagram

USER INTERFACE DESIGN: Characteristics of a good user interface, basic concepts, types of user interfaces

Unit -III	**10 Hours**

CODING AND TESTING: Introduction to program testing, Coding, code review, software

documentation, testing, unit testing, black – box testing,
White – box testing, debugging, program analysis tools, integration testing, testing object-oriented programs, systems testing
SOFTWARE RELIABILITY AND QUALITY MANAGEMENT: Software reliability, statistical testing, software quality, software quality management system, ISO 9000, SEI capability maturity model
COMPUTER AIDED SOFTWARE ENGINEERING: CASE and its scope, Case Environment, CASE support in software life cycle, other characteristics of CASE tools

Unit-IV	**10 Hours**

SOFTWARE PROJECT MANAGEMENT: software project management complexities, responsibilities of a software project manager, project planning, metrics for project size estimation, project estimation techniques, COCOMO – a heuristic estimation technique, Staffing level estimation, scheduling, organization and team structures, staffing, risk management, software configuration management
EMERGING TRENDS: client- server software, client server architectures, CORBA, COM/DCOM, Service - oriented architecture (SOA), software as a service (SaaS).

Text Book	:	Rajib. Mall, **"Fundamentals of software engineering"**, 4th edition, pHI
Reference Books	:	1. Ian Somerville **"Software Engineering"**, 7th edition, Pearson Education. 2. Pressman R.S, **"Software Engineering- A Practitioners Approach"**, MGH New Delhi. 3. Jalote P, Narosa **"An integral approach to software Engineering",** New Delhi.

UIS503C: Database Management System
(4-0-0)(4 Credits, 52 Hours)

Course Objectives :

1. Get an idea of defining, constructing, manipulating, and sharing databases among various users and applications
2. Learn about database design models, especially come to know about entity-relationship diagrams
3. Acquire knowledge about relational model, relational model constraint, and relational algebra.
4. Understand structured query language.
5. Know about different normal forms and properties of relational decompositions.
6. Learn about transaction management and crash recovery.

Course Outcomes :

1. Comprehend basics of various database systems
2. Apply high level modeling concepts to the given database problem domain
3. Demonstrate relational modeling concepts and apply ER-to-Relational mapping to model ER diagram into set of tables
4. Design normalized databases
5. Develop database applications
6. Comprehend basics of transaction processing concepts and various concurrency control & database recovery techniques.

Mapping (Co-relations) of CO's with PO's

Sl. No.	Course Outcomes	IS-PO-1	IS-PO-2	IS-PO-3	IS-PO-4	IS-PO-5	IS-PO-6	IS-PO-7	IS-PO-8	IS-PO-9	IS-PO-10	IS-PO-11	IS-PO-12
01	Comprehend basics of various database systems.	**1**	**1**	**1**		**1**		**1**				**1**	**1**
02	Apply high level modeling concepts to the given database problem domain.	**1**	**1**	**1**	**1**	**1**						**1**	**1**
03	Demonstrate relational modeling concepts and apply ER-to-Relational mapping to model ER diagram into set of tables.	**1**	**1**	**1**	**1**	**1**						**1**	**1**
04	Design normalized databases.	**1**	**1**	**1**	**1**	**1**						**1**	**1**
05	Develop database applications.	**3**	**1**	**3**	**1**	**2**		**2**		**1**	**1**	**2**	**3**
06	Comprehend basics of transaction processing concepts and various concurrency control & database recovery techniques	**2**	**1**	**2**	**1**	**2**		**2**		**1**	**1**	**2**	**3**

Syllabus

College Name	:	**Basaveshwar Engineering College (Autonomous), Bagalkot**
Department Name	:	**Information Science and Engineering**
Semester	:	**V**
Subject	:	**DATABASE MANAGEMENT SYSTEMS**
Subject code	:	**UIS503C**
Credits	:	**04**
Teaching Hours		**52**

UNIT - I

INTRODUCTION: Introduction; An example; Characteristics of database approach; Advantages of using DBMS approach; when not to use a DBMS. Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment; Centralized and client-server architectures; Classification of Database Management systems.

ENTITY-RELATIONSHIP MODEL: Using High-Level Conceptual Data Models for Database Design; An example database application; Entity types, Entity sets, Attributes and Keys; Relationship types, Relationship sets, Roles and Structural constraints; Weak entity types; Refining the ER Design; ER Diagrams, Naming conventions and design issues; Relationship types of degree higher than two.

12 hours

UNIT - II
RELATIONAL MODEL AND RELATIONAL DATABASE CONSTRAINTS: Relational model concepts; Relational model constraints and Relational database schemas; Update operations, Transaction and dealing with constraint violations. RELATIONAL ALGEBRA: Unary relational operations: SELECT and PROJECT; Relational algebra operations from set theory; Binary relational operations: JOIN and DIVISION; Additional relational operations; Examples of queries in relational algebra; Relational database design using ER- to-Relational mapping. SQL: data definition and data types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL queries. Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; **14 hours**
UNIT - III
DATABASE DESIGN: Informal design guidelines for relation schemas; Functional dependencies; Normal forms based on primary keys; General definitions of second and third normal forms; Boyce-Codd Normal Form Properties of relational decompositions; Algorithms for relational database Schema design; Multivalued dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form; Inclusion Dependencies; Other Dependencies and Normal forms. **13 hours**
UNIT - IV
TRANSACTION MANAGEMENT : Introduction to transaction processing; Transaction & system concepts; Desirable properties of transactions; Characterizing schedules based on recoverability; Characterizing schedules based on serializability; Transaction support in SQL; CONCURRENCY CONTROL: Two-phase locking techniques for concurrency control; CRASH RECOVERY: Recovery concepts; Recovery techniques based on deferred update; recovery techniques based on immediate update; shadow paging; The ARIES recovery algorithm; **13 hours**

Text Book	:	1. Remez Elmasri & Shamkant B. Navathe, **"Fundamentals of Database Systems"**, 5th Edition, Pearson Education.
Reference Books	:	1. Ramakrishanan Gehrke, **"Database Management Systems"**, 3rd edition, McGraw-Hill Higher Education. 2. C. J. Date, **"An Introduction to Data base systems"**, Addision Wesley, 4th edition.

UIS505C: Java programming
(4-0-0)(4 Credits, 52 Hours)

Course Objectives :

1) To learn the basic syntax and semantics of the Java language and programming environment

2) To understand the concepts of classes and objects

3) To understand the primitive data types built into the Java language and the difference between variables of primitive types and variables of class types

4) To understand features of a strongly typed language: variable declaration and type compatibility checking

5) To learn about lifetime, scope and the initialization mechanism of variables

6) To be able to implement decisions using if statements

7) To be able to program loops with while, for and do statements

8) To be introduced to classes, constructors and inheritance

9) To be able to design and implement packages , interfaces and applets

10) To understand and learn the concepts of exception handling, multithreading and file handling.

Course Outcomes :

1) Demonstrate an ability to explain fundamental java programming concepts, inheritances, exceptions, applets, packages, interfaces, multithreading and file handling concepts.

2) Demonstrate knowledge and understanding of java programming concepts by writing programming examples and solving problems.

3) Identify and apply the concepts of java programming to solve real life problems.

4) Create, design and use packages, interfaces, multiple threads and files in java programming to solve complex problems.

Mapping (Co-relations) of CO's with PO's

Sl. No.	Course Outcomes	IS-PO-1	IS-PO-2	IS-PO-3	IS-PO-4	IS-PO-5	IS-PO-6	IS-PO-7	IS-PO-8	IS-PO-9	IS-PO-10	IS-PO-11	IS-PO-12
01	Demonstrate an ability to explain fundamental java programming concepts, inheritances, exceptions, applets, packages, interfaces, multithreading and file handling concepts.	1	1	1	1								
02	Demonstrate knowledge and understanding of java programming concepts by writing programming examples and solving problems.	1	2	2	2			1					
03	Identify and apply the concepts of java programming to solve real life problems.	1	2	3	3	2		1					
04	Create, design and use packages, interfaces, multiple threads and files in java programming to solve complex problems.	1	2	3	3	2		1					

Syllabus

Academic Year	**:**	**2018 - 19**
Semester	**:**	**5**
Subject	**:**	**Java Programming**
Subject Code	**:**	**UIS505C**
Credits	:	**04**
Teaching Hours	:	**52**

Unit –I **13 Hours**

Introduction to Java:
History of Java; Java changed the internet: Applets; Byte code: JVM; Java features, JDK, Evolution of Java.

Object Oriented Programming: Abstraction, Encapsulation, Inheritance and Polymorphism; Simple java programs; Lexical issues.

Data types: Integers, Floating point types. Characters, Booleans; **Literals:** Integer literals, Floating point literals, Character literals, Boolean literals, String literals; **Variables; Type**

Conversion and Casting: Java automatic conversion, Casting incompatible casts, Automatic type promotion in expression

Arrays: one dimensional array, Multi dimensional arrays, Strings.

Operators: Arithmetic operators, Bitwise operators, Relational operators, Boolean Logical operators, Assignment operators, The? Operator, Operator precedence;

Control statements: Selection statements, Iteration statements, Jump statements.

Unit –II	**13 Hours**

Classes, Inheritance and Exceptions:

Classes: Defining classes, Declaring Objects, Assigning object reference variables, Defining methods; Constructors: Default constructors, Parameterize constructors, 'this' key word; Overloading: Overloading methods, Overloading constructors; Access control; Understanding static; Introducing final, Nested and inner classes; exploring the string class, using command line arguments.

Inheritance: Super classes, Sub classes, Member access, Using super, Multi level inheritance, Method over riding;

Exceptions: Fundamentals, Exception types, Uncaught exceptions, Using try and catch, Multiple catch clauses, Nested try statements, throw, throws.

Unit -III	**13 Hours**

Packages, Interfaces and Applets:

Packages: Defining packages, Finding packages and CLASSPATH, Access protection, importing packages;

Interfaces: Defining interfaces, Implementing interfaces, Nested interfaces, Applying interfaces, Variables in interfaces;

The Applet Class: Two types of Applets, Applet basics, Applet Architecture, An Applet skeleton, Simple Applet display methods, Requesting repainting, Using the Status Window, THE HTML APPLET tag, Passing parameters to Applets, getDocumentbase() and getCodebase(), AppletContext and showDocument(), The Audioclip Interface, The AppletStub Interface, Output to the Console.

Unit-IV	**13 Hours**

Multithreaded programming and Managing Files:

Multithreaded programming: Java thread model, Main thread, Creating a thread, Creating multiple threads, Using isArive() and join(), Thread priorities, Synchronization, Interthread communication, Suspending, Resuming and stopping threads, Using multi threading;

Managing input/output files in Java: Java I/O classes and interfaces, Files, Stream classes, Byte Streams, Random access file, Character streams, Serialization, Stream benefits

Text Books	**:**	1) Herbert. Schildt, **"The Complete Reference –Java"**, 10th edition, McGraw Hill Publication. 2) E. Balaguruswamy, **"Programming with Java - A primer"**, 4th edition, McGraw Hill publication.
Reference Books	**:**	1) Paul J. Deitel and Harvey M. Deitel, **"Java for programmers"**, Pearson Education. 2) Y. Daniel Liang, **"Introduction to Java programming"**, 7th edition, Pearson education.

UIS506C: Operating System
(4-0-0)(4 Credits, 52 Hours)

Course Objectives : **1)** Understand system architecture, Operating System structure, operations and services.

2) Understand the concept of process and threads and implement various scheduling algorithms

3) Understand the fundamental principles of synchronization, able to apply banker's algorithm to ensure system safety and resolve deadlocks

4) Understand various issues in Memory Management

5) Understand File structures and gain knowledge about system security and protection.

Mapping (Co-relations) of CO's with PO's

Sl. No.	Course Outcomes	IS-PO-1	IS-PO-2	IS-PO-3	IS-PO-4	IS-PO-5	IS-PO-6	IS-PO-7	IS-PO-8	IS-PO-9	IS-PO-10	IS-PO-11	IS-PO-12
01	Understand the principle of operating system	3	1										2
02	Comprehend the concept of process, thread and scheduling algorithms	3	3	2	2	2							2
03	Demonstrate an understanding of race condition, process synchronization and coordination and ensure the safety of the system by resolving deadlocks	3				3							2
04	Exhibit a good knowledge of fundamental principles of memory management	3	2	3	3		2		2				2
05	Appreciate the implementation of file system, importance of system security, protection and solution used	2	3	2	2								2

Syllabus

Academic Year	**:**	**2018 - 19**
Semester	**:**	**05**
Subject	**:**	**Operating system**
Subject Code	**:**	**UIS506C**
Credits	:	04
Teaching Hours	:	52

Unit –I	**13 Hours**

INTRODUCTION TO OPERATING SYSTEMS, PROCESS MGMT

Role of Operating systems: user view, system view; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines. Process management: Process concept; Operations on processes; Process Scheduling: Basic concepts; scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling.

Unit –II	**13 Hours**

THREADS AND PROCESS SYNCHRONIZATION

Interprocess communication, Threads: concepts, Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Thread scheduling. Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

Unit -III	**13 Hours**

DEADLOCKS AND MEMORY MANAGEMENT

Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Page replacement; Allocation of frames; Basics concepts of Thrashing.

Unit-IV	**13 Hours**

FILE SYSTEM: CONCEPTS AND IMPLEMENTATION, SECONDARY STORAGE STRUCTURES

File system: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management, Protection: Goals, principles and domain of protection, Access Matrix

Text Book	**:**	1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "**Operating System Principles**", 7th edition, Wiley-India, 2006.
Reference Books	**:**	1) Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "**Operating System Principles**", 9th edition, Wiley-India, 2016. 2) P.C.P. Bhatt, "**Operating Systems**", 2nd Edition, PHI, 2006. 3) Harvey M Deital "**Operating Systems**", 3rd Edition, Addison Wesley, 1990. 4) D.M Dhamdhere, "**Operating Systems: A concept based Approach**", 2nd Edition, Tata McGraw- Hill, 2002.

UIS059E: Information Security
(4-0-0)(4 Credits, 52 Hours)

Course Objectives :

1. To understand the basics of information security.
2. To know the aspects of risk management.
3. To understand security standards, policies, blueprint and other models used.
4. Apply the methodology of Security Systems Development Life Cycle.
5. Understanding different types of security attacks, security mechanisms and security services to save computer systems.
6. Analyze and design classical encryption techniques for providing security to information.
7. To understanding security technologies such as firewalls, intrusion detection.
8. Develop an understanding of threats like virus, worm etc.

Course Outcomes :

1. Discuss the basics of information security and demonstrate the aspects of risk management.
2. Student will be able to understand the different policies and standards used in security.
3. To master understanding external and internal threats, attacks to an organization.
4. Understand different types of security attacks, security services and security mechanisms.
5. Analyze various encryption and decryption techniques for providing security to computer system.
6. Comprehend the basic concepts such as firewalls, intrusion detection and different types of threats like virus, worm.

Mapping (Co-relations) of CO's with PO's

Sl. No.	Course Outcomes	IS-PO-1	IS-PO-2	IS-PO-3	IS-PO-4	IS-PO-5	IS-PO-6	IS-PO-7	IS-PO-8	IS-PO-9	IS-PO-10	IS-PO-11	IS-PO-12
01	Discuss the basics of information security and Demonstrate the aspects of risk management.		**2**	**3**	**2**	**3**	**3**						**3**
02	Student will be able to understand the different policies and standards used in security.		**2**	**1**	**1**	**1**	**2**	**3**	**2**				**3**
03	To master understanding external and internal threats, attacks to an organization.		**3**	**1**		**2**	**2**						**3**
04	Understand different types of security attacks, security services and security mechanisms.			**2**	**2**	**1**	**3**	**2**	**1**				**3**
05	Analyze various encryption and decryption techniques for providing security to computer system.	**1**	**3**	**2**	**2**	**3**	**2**	**2**		**1**		**1**	**1**
06	Comprehend the basic concepts such as firewalls, intrusion detection and different types of threats like virus, worm.	**2**	**2**	**2**	**2**	**2**	**2**	**3**	**2**				**1**

Syllabus

Academic Year	**:**	**2018 - 19**
Semester	**:**	**V**
Subject	**:**	**Information Security**
Subject Code	**:**	**UIS059E**
Credits	:	**04**
Teaching Hours	:	**52**

Unit –I	**10 Hours**

INTRODUCTION: What is Security, CNSS Security Model, and Components of an Information System, Balancing Information Security and Access, The Systems Development Life Cycle, The Security Systems Development Life Cycle.

PLANNING FOR SECURITY: Introduction; Information Security Policy, Standards, and Practices; The Information Security Blue Print; Contingency plan and a model for contingency plan.

Unit –II	**10 Hours**

NEED FOR SECURITY: Business Needs, Threats, Attacks.

RISK MANAGEMENT: An Overview of Risk Management, Risk Identification, Risk

Assessment, Risk Control Strategies, Selecting a Risk Control Strategy.		
Unit -III		**10 Hours**
CRYPTOGRAPHY: Security Attacks, Security Services and Security Mechanisms; A model for Internetwork Security. **CLASSICAL ENCRYPTION TECHNIQUES:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steaganography.		
Unit-IV		**10 Hours**
INTRUDERS: Intruders**,** Intrusion Detection, Password Management. **FIREWALLS:** The Need for Firewalls, Firewall Characteristics, Types of Firewalls, Firewall Basing, Firewall configurations. **MALICIOUS SOFTWARE:** Types of Malicious Software, Viruses, Virus Countermeasures, Worms, Distributed Denial of Service Attacks.		
Text Books	**:**	**1)** Michael E. Whitman and Herbert J. Mattord, "**Principles of Information Security**", 4th Edition, Course Technology, Cengage Learning. (Chapters 1, 2, 4, 5) **2)** William Stallings, "**Cryptography and Network Security – Principles and Practices**", Pearson Education, Fifth Edition.(Chapters 1, 2, 20, 21, 22)
Reference Books	**:**	**1)** Micki Krause, Harold F. Tipton, "**Handbook of Information Security Management**", Vol 1-3 CRC Press LLC, 2008. **2)**Atul Kahate, "**Cryptography and Network Security**", Tata McGraw Hill, 2003. **3)**Behrouz A. Forouzan, "**Introduction to Cryptography and Network Security**", 2008, McGraw-Hill.

UIS051E: Advanced Graph theory
(4-0-0)(4 Credits, 52 Hours)

Course Objectives :

1. Understand basic definitions and properties of a graph.
2. Understand fundamental theorems of graph theory.
3. Identify different kinds of special graphs and describe the basic properties of each kind.
4. Show certain characteristics of a graph under its given description.
5. Understand basic principles of important graph algorithms such as finding shortest path,minimum spanning tree, maximum flow and minimum cut, etc.
6. Apply basic techniques and strategies of graph theory to solve real world problems.

Course Outcomes :

1. Demonstrate understanding basics of graph theory such as tree, matrix, cutset, k-chromatic etc.
2. Demonstrate ability to illustrate knowledge of graph theory.
3. Demonstrate ability to apply basic concepts of graph theory to model the real life examples.
4. Analyze and apply graph theory concepts to solve societal problems.

Mapping (Co-relations) of CO's with PO's

S.N	Course Outcomes	IS –PO -	IS –PO-	IS –PO-	IS -PO-4	IS -PO-5	IS -PO-6	IS -PO-7	IS -PO-8	IS -PO-9	IS -	IS -	IS -
CO-1	Demonstrate understanding basics of graph theory such as tree, matrix, cutest, k-chromatic	3	3										
CO-2	Demonstrate ability to illustrate knowledge of graph theory.	3	3	3	3	3	3	3					

CO-3	Demonstrate ability to apply basic concepts of graph theory to model the real life examples	3	3	3	3	3	3	3					
CO-4	Analyze and apply graph theory concepts to solve societal problems	3	3	3	3	3	3	3					

Syllabus

Academic Year	**:**	**2018 - 19**
Semester	**:**	**05**
Subject	**:**	**Advanced Graph theory**
Subject Code	**:**	**UIS051E**
Credits	:	**4**
Teaching Hours	:	**52**

Unit –I **13 Hours**

Introduction: What is a graph?, Applications of graphs, Finite and Infinite Graphs, Incidence and degree, Isolated and pendent vertices, null graphs.

Paths and Circuits: Isomorphism, Sub graphs, Walks, Paths, and Circuits, Connected graphs, Disconnected graphs, and components. Euler graphs, Hamiltonian paths and circuits. Traveling salesman problem.

Trees and Fundamental Circuits: Trees, Properties of trees, Pendent vertices in trees, Distance and centers in trees, Rooted and binary trees, Spanning trees, Fundamental Circuits, All spanning trees, spanning trees in a weighted graph.

Unit –II **13 Hours**

Cuts and vertices: Cut-sets, Properties of cut-sets, All cut-sets in a graph, connectivity and seperability, network flows, I & II isomorphism.

Planar and dual graphs: Combinatorial v/s geometric graphs, planar graphs, representations of a planar graphs, detection of planarity, geometric dual, combinatorial dual, criteria of planarity.

Matrix representations of graphs: Incident matrix, submatrices of A(G), Circuit matrix, Fundamental circuit matrix and rank of B, Cut-set matrix, Relationships among A_f, B_f, and C_f. Path matrix, adjacency matrix

Unit -III **13 Hours**

Coloring, Covering and partitioning: Chromatic Number, Chromatic partitioning, Chromatic Polynomial, Matchings, Coverings, four color problem.

Directed Graphs: What is Directed graphs, types of directed graphs, digraphs and binary relations, Euler digraphs, trees with directed edges, Fundamental circuits in directed graphs, Matrices A, B and C of digraphs, Adjacency matrix of digraphs, paired comparison and tournament, acyclic digraphs and decylization.

Unit-IV **13 Hours**

Graph theoretic algorithma and Computer programs: Algorithms, representation of a graphs, some basic algorithms, connectedness and components, a spanning tree, a set of fundamental circuits, cut-vertices and seperability, directed circuits, shortest paths algorithms, depth first

search on a graph, isomorphism. Graphs in computer programming.		
Text Book	**:**	Narsingh. Deo, "**Graph Theory: With applications to Engineering and Computer Science**", PHI Publisher (Ch.1, 2, 3, 4, 5, 7, 8, 9, 11, 15.3)
Reference Books	**:**	Ralph P. Grimaldi, B V. Ramana, "**Descrete and Combinatorial Mathematics**", 5th edition, Pearson Education.

UIS525L: System Software Laboratory
2 Credits (0-0-4)

LEX Programs

1. The number of vowels and consonants in a given string.
2. The number of characters, word, spaces, and of lines in a given input file.
3. the number of:
    a) +ve and –ve integers.
    b) +ve and –ve fractions.
4. The number of comment line in a given C program. Also eliminate them and copy that program into separate file.
5. The number of 'Scanf' and 'Printf' statements in a C program. Replace them with 'readf' and 'writef' statements respectively.
6. Program to recognize whether a given sentence is simple or compound.
7. Program to recognize and count the number of identifiers in a given input file.
8. Counting the number of small letters and capital letters.
9. Counting the number of digits and special characters.
10. Program to find the length of largest word in given sentence.

YACC Programs

1. Program to test the validity of simple expression involving operators +,-,* and /.
2. Program to recognize a valid variable, which starts with a letter followed by any number of letters or digits.
3. Program to evaluate an arithmetic expression involving operating +,-,* and /.
4. Program to recognize strings 'aaab, abbb ',' ab and 'a' using grammar.
5. Program to recognize the grammar (a^nb^n,n >=10).

General Remarks:

1) Lab schedule: 3hrs/week for each student.
2) Student should complete all the lab assignments.
3) Evaluation CIE 50 marks:
   Lab assignments : 30 marks
   Lab CIE : 20 marks

UIS510L: Java Programming Lab
1.5 Credits (0-0-3)

1. Write a Java program to read the two matrices A (m x n) & B (x x y) and find the product matrix C (m x y).

2. Write a Java program to perform arithmetic operations on two complex numbers and print the resultant complex number.

3. The annual examination results of N students are tabulated as follows:

Roll. No	Subject	Subject 2	Subject 3
.	.	.	.
.	.	.	.
.	.	.	.

Write a Java program to read the data and determine

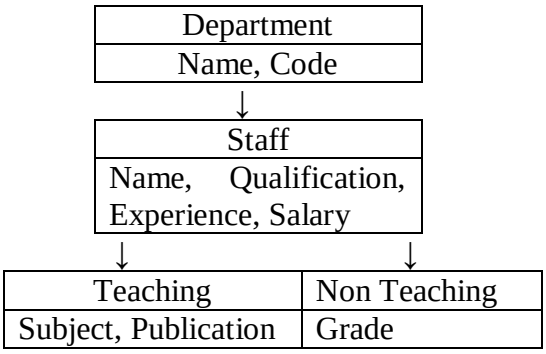
a) Total marks obtained by each student
b) The highest marks in each subject & the roll number of the student who scored it
c) The student who obtained the highest total marks.

4. Write a Java program to sort an array of strings

5. Create a class ACCOUNT that stores customer name, account number, and type of account. From this derive the classes **savings account** and **current account**. The savings account provides compound interest and withdrawal facility but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed. Write a Java program to achieve the following tasks:

a) Accept deposit from a customer & update the balance
b) Display the balance
c) Compute & deposit interest
d) Permit withdrawal & update the balance
e) Check for the minimum balance, impose penalty, if necessary and update the balance

6. An educational institute wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are as shown in below figure. Write a Java program to specify all the classes & define methods to create the database and retrieve individual information & when required.



7. Design a package to contain class EMPLOY which holds the information of employees such as Employ number, Name, Department, and Designation, another package to contain Basic salary, Allowance & Deductions. Write a Java program to input class obtained in these two packages & bring employ number, name, department, designation, basic salary, allowance & deduction along with net salary.

8. Write a Java program to implement the stack operations by defining an interface for stack & a class that implement the stack.

9. Develop an applet that receives three numeric values as input from the user & then displays the largest of the three on the screen. Write a HTML page & test the applet.

10. Write a Java program to print "mythead1" and "mythead2" in parallel by creating thread object.

11. Write a Java program to read string from the keyboard terminated by the symbol $ and write this stream to the filename, r.dat and display the largest character among the string, find total number of characters and total number of words in the stream from the file r.dat.

12. Write a Java program with multiple threads to perform

$P = \sin(x) + \cos(y) + \tan(z)$

General Remarks:

4) Lab schedule: 3hrs/week for each student.
5) Student should complete all the lab assignments.
6) Evaluation CIE 50 marks:
   Lab assignments : 30 marks
   Lab CIE : 20 marks

SYLLABUS

College Name	:	**Basaveshwar Engineering College (Autonomous), Bagalkot**
Department Name	:	**Information Science and Engineering**
Semester	:	**V**
Subject	**:**	**Database Application Laboratory**
Subject code	**:**	**UIS511L**
Credits	:	**1.5**
Teaching Hours	:	**3Hrs/Week**

List of Assignments

I. Consider the following relations:

Student(**snum**: integer, sname: char, subject: char, slevel: char, age: integer)
Class(**cname:**char, roomno: integer, **fid:** integer)
Enrolled(**snum:**integer, **cname:**char)
Faculty(**fid:** integer, fname: char, deptid: integer)

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 the answers.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by faculty name.
2. For each level, print the level and the average age of students for that level.
3. For all levels except JR, print the level and the average age of students for that level.
4. For each faculty member that has taught classes only in roomno 20, print the faculty member's name and the total number of classes she or he has taught.
5. Find the names of students not enrolled in any class.

II. The following relations keep track of airline flight information:

Flights(flno: varchar, source: char, destination: char, distance: float, departs: float, arrives: float, price: real)
Aircraft(aid: integer, aname: char, cruising_range: integer)
Certified(eid: integer, aid: integer)
Employees (eid: integer, ename: char, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL.

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than 30,000.
2. For each pilot who is certified for more than one aircraft, find the eid and the maximum cruising range of the aircraft for which she or he is certified.

3. Find the names of pilots whose salary is less than the price of the cheapest route from Bagalkot to Bangalore.
4. Find the names of pilots certified for aircraft name starting with Air.
5. Find the flight Number of all the flights that can be used on routes from Bagalkot to Bangalore.
6. Print the names of employees who are certified only on aircrafts with cruising range longer than 400 miles.
7. Print the names of employees who are certified only on aircrafts with cruising range longer than 100 miles and who are certified on some aircraft.

III. Consider the Insurance database given below. The primary keys are underlined and the data types are specified:

PERSON (Driver – id #: string, Name: string, Address: string)
CAR (Regno: string, Model: string, Year: int)
ACCIDENT (Report-number: int, Accd-Date: date, Location: string)
OWNS (Driver-id #: string, Regno:string)
PARTICIPATED (Driver-id: string, Regno:string, Report-Number: int, Damage Amount: int)

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Demonstrate how you
   a. Update the damage amount to 25000 for the car with a specific Regno in the ACCIDENT table with report number 12.
   b. Add a new accident to the database.
3. Find the total number of people who owned cars that were involved in accidents in 2008.
4. Find the number of accidents in which cars belonging to a specific model were involved.

IV. Consider the following database of student enrollment in courses & books adopted for each course:

STUDENT (Regno: string, Name: string, Bdate: date)
COURSE (Course #: int, Cname: string, Dept: string)
ENROLL (Regno: string, Course#: int, Sem: int, Marks: int)
BOOK–ADOPTION (Course#: int, Sem: int, Book-ISBN: int)
TEXT (Book–ISBN: int, Book–Title: string, Publisher: string, Author: string)

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Demonstrate how you add a new textbook to the database and make this book be adopted by some department.
3. Produce a list of textbooks (include Course#, Book–ISBN, Book–Title) in the alphabetical order for courses offered by the ‘IS’ department that use more than two books.
4. List any department that has all its adopted books published by a specific publisher

V. Consider the following relations for a order processing database application in a company.

CUSTOMER(**custno:int ,**cname:string , city:string)

ORDER(**orderno:int** , odate:date , custno:int , ord_amt:int)

ORDER_ITEM(orderno:int , itemno:int , quantity:int)

ITEM(**itemno:int** , unitprice:int)

SHIPMENT(orderno:int , warehouseno:int , ship_date:date)

WAREHOUSE(**warehouseno:int** , city:string)

1. Create the above tables by properly specifying the primary keys and foreign keys.
2. Enter at least five tuples for each relation.
3. Produce a listing: custname ,No_of_orders , Avg_order_amount , where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.
4. List the orderno for orders that were shipped from all the warehouses that the company has in a specific city.
5. Demonstrate the deletion of an item from the ITEM table

VI. Consider the following database for a banking enterprise:

BRANCH(**Branch-name**:string, Branch-City:string, Assets:real)

ACCOUNT(**Accno**:int, Branch-Name:string, Balance:real)

DEPOSITOR(Customer-Name:string, Accno:int)

CUSTOMER(**Customer-Name**:string, Customer-Street:string, Customercity: string)

LOAN(**Loan-Number**:int, Branch-Name:string, Amount:real)

BORROWER(Customer-Name:string, Loan-Number:int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys

(ii) Find all the customers who have at least two accounts at the Main branch.

(iii) Find all the customers who have an account at all the branches located in a specific city.

(iv) Demonstrate how you delete tuples in ACCOUNT relation at every branch located in a specific city.

(v) Find all loan numbers for loans made at the specific branch with loan amounts greater than Rs1200.

(vi) Find all loan numbers for loans with loan amounts between Rs 90,000 and Rs100000.

General Remarks:

1) Lab schedule: 3hrs/week for each student.
2) Student should complete all the lab assignments.
3) Evaluation CIE 50 marks:

Lab assignments : 30 marks

Lab CIE : 20 marks