Name: Dipali Yogesh Jadhav

Class: T.Y.B.Sc(Comp.Sci.)

Course Name: Systems Programming

Course code: CS-331

Course outcomes:

CO 1. To understand the design structure of a simple editor.

CO2. To understand the design structure of Assembler and macro processor .

CO 3. To understand the mechanism of **Systems Programming language**.

CO4. To understand Complexity of Operating system as a software.

CO 5.To acknowledge various components of system programms.

Course T.Y.BSc(Comp.Sci.)	Course Specific Outcome CSO	Methodology	Reference Book	No. of Lectures
Introduction Types of program – System program and Application program. Elements of Programming environment Simulation of simple computer	How to write simulation program. get the knowledge of programming environment. To know what is system programming and application programming.	Constructive	Systems Programming and Operating Systems	4
Editors Definition, need/purpose of editor. Structure of editor	The types of editors. knowledge of structure of Editors.	Demonstrative	Systems Programming and Operating Systems	2
Assembler Features of assembly language, advantages Constants and Literals. Design of assembler Pass Structure of Assembler Data structure of 2-pass assembler Intermediate Code	Different pass structure of Assembly. How to write intermediate code. How to use different types of statement in assembly language.	Demontrative	Systems Programming and Operating Systems	10
Macros and Macro Processors Macro definition and call Macro expansion Nested macro calls Advanced macro facilities	get knowledge of Macro expansion. understand Design of macro preprocessor	Lecture	Systems Programming and Operating Systems	10

Design of macro preprocessor				
Compiler Design options Interpreter P-code compiler	understand interpreter, compiler	Lecture	Systems Programming and Operating Systems	2
Linker and Loader Concept of bindings, static and dynamic binding Relocation and linking concept Relocatability Object file formats	understand the concept of binding. get the knowledge of various Object file formats.	Lecture	Systems Programming and Operating Systems	6
Debuggers & Development utilities Debugging functions and capabilities Types of debuggers Development utilities	understand Debugging functions and capabilities. understand different development utilities on Unix environment. learn overview of Debugging Tools.	Constructive	Systems Programming and Operating Systems	6
Operating System as System Software Computer System Architecture Operating System Operations Process Management Memory Management Protection and Security Distributed Systems Computer Environment	understand purpose of operating system with system and user view. understand the concept of Protection and security. study Different Computing Environment.	Demontrative	Systems Programming and Operating Systems	6
System Structure Operating System Services User Operating-System Interface Types of System Calls	understand different operating System services. learn concept of system calls. Study different operating System user interfaces.	Constructive	Systems Programming and Operating Systems	2

References:

- Systems Programming and Operating Systems by D.M.Dhamdhere

Dipali Yogesh jadhav
M.Sc(Comp.Sci.)
Mamasaheb Mohol College

Course Name: Theoretical Computer Science

Class T.Y.B.Sc.(CS)

Course code: CS 332

Course outcome :

CO1. Understand the complex logic using automata theory as well as the theory of language.

CO2. Able to model computation using finite automata, and characterize simple languages with regular expressions and context-free grammars.

CO3.Students able how the Turing machine models everything that can be computed using a real computer and that there are well-defined problems that cannot be solved with any algorithm.

CO4. Understand the relationship between the expressiveness of the models of computation studied during the course.

CO5. Exposed to some basic concepts of computational complexity theory such as reductions between problems.

Course Cs-332 T.Y.B.Sc.(CS)	Course Specific Outcome CSO	Methodology	Reference Book	No. of Lecture
Introduction 1.1 Symbol, Alphabet, String, Prefix&& Suffix of Strings, Formal Language, Operations on Languages. 1.2 Regular Expressions (RE) : Definition & Example 1.3 Regular Expressions Identities.	Understand the concept of set theory and language theory. To get knowledge of formal language and its role in finite state machine.	Constructivism	Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.	3
Finite Automata 2.1 Deterministic finite Automaton – 2.2 Nondeterministic finite automaton – Definition and Examples. 2.3 NFA TO DFA : Method 2.4 NFA with ε- transitions Definition and Examples. 2.5 NFA with ε-Transitions to DFA & Examples 2.6	To know the concept of automata theory. Students can able to use the concept of idealized machine for solving different computational problem	Demonstration	Theory of Computer Science (Automata, Language & Computation) K. L. P. Mishra & N. Chandrasekaran, PHI Second Edition	12

Finite automaton with output – Mealy and Moore machine.				
Regular Language 3.1 Regular language- Definition and Examples. 3.2 Conversion of RE To FA-Examples. 3.3 Pumping lemma for regular languages and applications. 3.4 Closure properties of regular Languages (Union, Concatenation, Complement, Intersection and Kleene closure)	Concept of regular expression uses in various applications like operating system use regular expression in commands of command line interface.	Constructivism	. Introduction to Automata theory, Languages and computation By John Hopcroft, Rajeev Motwani and Jeffrey Ullman –Third edition Pearson Education	5
Context free grammar and Language 4.1 Grammar - Definition and Examples. 4.2 Derivation-Reduction - Definition and Examples. 4.3 Chomsky Hierarchy. 4.4 CFG : Definition & Examples. LMD, RMD, ,Parse Tree 4.5 Ambiguous Grammar : Concept & Examples. 4.6 Simplification of CFG : 4.7 Normal Forms 4.8 Regular Grammar : Definition. 4.9 Closure Properties of CFL's	Understand the fundamental concepts regarding the grammar and formal language.	Constructivism	Introduction to Automata theory, Languages and computation By John Hopcroft, Rajeev Motwani and Jeffrey Ullman –Third edition Pearson Education	12
Push Down Automata 5.1 Definition of PDA and examples 5.2 Construction of	Students will able to do different computation and mathematical model	Constructivism	Introduction to Automata theory, Languages and	6

PDA using empty stack and final State method : Examples using stack method 5.3 Definition DPDA & NPDA, their correlation and Examples of NPDA 5.4 CFG (in GNF) to PDA : Method and examples	construct using PDA.		computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.	
Turing Machine 6.1 The Turing Machine Model and Definition of TM 6.2 Design of Turing Machines 6.3 Problems on language recognizers. 6.4 Language accepted by TM 6.5 Types of Turing Machines(MultitrackTM,Two way TM, MultitapeTM,Non-deterministic TM) 6.6 Introduction to LBA (Basic Model) &CSG.(Without Problems) 6.7 Computing TM, Enumerating TM, Universal TM 6.8 Recursive Languages 6.9 Turing Machine Limitations 6.10 Decision Problem, Undecidable Problem, Halting Problem of TM	By using Turing machine models everything that can be computed using a real computer.	Constructivism	Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.	10

References: - 1 Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.

2. Introduction to Automata theory, Languages and computation By John Hopcroft, Rajeev Motwani and Jeffrey Ullman –Third edition Pearson Education

 3. Introduction to Computer Theory Daniel I. A. Cohen – 2 nd edition – John Wiley & Sons

 4. Theory of Computer Science (Automata, Language & Computation) K. L. P. Mishra & N. Chandrasekaran, PHI Second Edition

 5. Introduction to Languages and the Theory of Computation John C. Martin TMH, Second Edition

Mrs.Rajnigandha Y. Ingole

M.Sc.(Computer Science) B.Ed.

MamasahebMohol College Pune-38

Course:-Computer Network-I

Name of Teacher:- Prof. D.D Pokhalekar
Class:- T.Y.B.Sc(CS)
Course Code: - 333

Course Outcomes:-

CO1) Understands different types of networks, various topologies and application of networks.
CO2) Learns types of addresses, data communication.
CO3) Acquires knowledge the concept of networking models, protocols, functionality of each
      layer.
CO4) Learn basic networking hardware and tools.
CO5) Get knowledge of overall mechanism of networking.

Course	Course Specific Outcome	Methodology	References	No. of lectures
Introduction to computer Network: Goals and applications – Business Application	Network Hardware - Broadcast and point-to-point. topologies – star, bus, mesh, ring etc Network Types-LAN, MAN, WAN, Wireless Networks, Home network	Constructive	By Andrew Tanenbaum, Pearson Education.[4th Edition]	8
Network models: OSI Reference Model TCP/IP Reference Model	Functionality of each layer Comparison of OSI and TCP/IP Reference Model Physical, Logical and Port addresses	Use of ICT	By Andrew Tanenbaum, Pearson Education.[4th Edition]	5
Transmission Media: Twisted pair cable – UTP Vs STP, categories connectors & applications , Coaxial cable – standards, connectors & applications	Unguided Media – Wireless- Radio Waves,-Infrared waves Light wave transmission Types of cabling and Networking Tool - CAT5 and CAT6 Cable	Constructive	Data Communication and Networking by Behrouz Forouzan,	5
The Physical Layer: Analog and Digital data Transmission Impairments Performance of the Network Bandwidth,	Digital Signals- Bit rate, bit length, baseband attenuation, distortion and noise, Data Rate Limits – Noiseless channel: Throughput, Latency(Delay), Bandwidth –Delay Product, Jitter	Constructive	Data Communication and Networking by Behrouz Forouzan,	14

MAMASAHEB MOHOL COLLEGE

48/1A, Erandwane, Paud Road, Pune - 411038(Maharashtra) India

DEPARTMENT OF COMPUTER SCIENCE

Course	Course Specific Outcome	Methodology	References	No. of lectures
The Data Link Layer: Design Issues Error detection code CRC Data Link Layer Protocols Data Link Layer Devices	Services provided to the Network Layer , Framing – Concept, Methods - Character Count, Flag bytes with Byte Stuffing, Sliding Window Protocols Piggybacking-N Bridges – Filtering, Transparent Bridges	Constructive	By Andrew Tanenbaum, Pearson Education.[4th Edition]	9
The Medium Access Sublayer: Random Access Protocols CSMA Channelization	ALOHA – pure and slotted – 1-persistent, p-persistent and non-persistent CSMA/CD,CSMA/CA FDMA, TDMA and CDMA	Use of ICT	Data Communication and Networking by Behrouz Forouzan,	7

Reference Books: 1) Computer Networks by Andrew Tanenbaum, Pearson Education.[4th Edition]
                  2) Data Communication and Networking by Behrouz Forouzan, TATA McGraw Hill.

Class: TY BSc(CS)

Course Title : Programming in java –I

Course Code : 335

Course Outcome:
CO1: Implement object oriented programming concepts.
CO2: Use and create package and interfaces in a Java program
CO3: Use graphical user interface in Java programs
CO4: Create applets.

Course TYBSc(CS)	Course Specific Outcome CSO	Methodology	Reference Book	No of Lectures
1. An Introduction to Java [4] 1.1 A Short History of Java 1.2 Features or buzzwords of Java 1.3 Comparison of Java and C++ 1.4 Java Environment 1.5 Simple java program 1.6 Java Tools – jdb, javap, javadoc 1.7 Java IDE – Eclipse/NetBeans (Note: Only for Lab Demonstration)	Implement basics of java Use java tools Compatible with java environment	Constructive	Complete reference Java by Herbert Schildt(5th edition)	4
2. An Overview of Java [4] 2.1 Types of Comments 2.2 Data Types 2.3 Final Variable 2.4 Declaring 1D, 2D array 2.5 Accepting input using Command line argument 2.6 Accepting input from console (Using BufferedReader class)	Use Command line arguments Create simple programs	Constructive	Complete reference Java by Herbert Schildt(5th edition	4
3. Objects and Classes [8] 3.1 Defining Your Own Classes 3.2 Access Specifiers (public, protected, private, default) 3.3 Array of Objects 3.4 Constructor, Overloading Constructors and use of 'this' Keyword 3.5 static block, static Fields and methods	Define class with method within it Use constructor Use keywords Create and use packages	Constructive	Complete reference Java by Herbert Schildt(5th edition	8

3.6 Predefined class – Object class methods (equals(), toString(), hashcode(), getClass()) 3.7 Inner class 3.8 Creating, Accessing and using Packages 3.9 Creating jar file and manifest file 3.10 Wrapper Classes 3.11 Garbage Collection (finalize() Method) 3.12 Date and time processing				
4. Inheritance and Interface [7] 4.1 Inheritance Basics (extends Keyword) and Types of Inheritance 4.2 Superclass, Subclass and use of Super Keyword 4.3 Method Overriding and runtime polymorphism 4.4 Use of final keyword related to method and class 4.5 Use of abstract class and abstract methods 4.6 Defining and Implementing Interfaces 4.7 Runtime polymorphism using interface 4.7 Object Cloning	Create and use class hierarchy Create and use Interfaces	Constructive	Complete reference Java by Herbert Schildt(5th edition	7
5. Exception Handling [4] 5.1 Dealing Errors 5.2 Exception class, Checked and Unchecked exception 5.3 Catching exception and exception handling 5.4 Creating user defined exception 5.5 Assertions	Dealing with errors Handle Exception Create Exception	Use of ICT	Complete reference Java by Herbert Schildt(5th edition	4
6. Strings, Streams and Files [7] 6.1 String class and StringBuffer Class 6.2 Formatting string data using format() method 6.2 Using the File class	Handle and Operate Strings Read and write different kind of Data within Files Access File properties	Constructive	Complete reference Java by Herbert Schildt(5th edition	7

6.3 Stream classes Byte Stream classes Character Stream Classes 6.4 Creation of files 6.5 Reading/Writing characters and bytes 6.6 Handling primitive data types 6.7 Random Access files				
7. User Interface Components with AWT and Swing [10] 7.1 What is AWT? What is Swing? Difference between AWT and Swing. 7.2 The MVC Architecture and Swing 7.3 Layout Manager and Layouts, The JComponent class 7.4 Components – JButton, JLabel, JText, JTextArea, JCheckBox and JRadioButton, JList, JComboBox, JMenu and JPopupMenu Class, JMenuItem and JCheckBoxMenuItem, JRadioButtonMenuItem , JScrollBar 7.5 Dialogs (Message, confirmation, input), JFileChooser, JColorChooser 7.6 Event Handling: Event sources, Listeners 7.7 Mouse and Keyboard Event Handling 7.8 Adapters 7.9 Anonymous inner class	Use MVC architecture Create graphical user interface in Java programs	Constructive	Complete reference Java by Herbert Schildt(5th edition	10
8. Applet [4] 8.1 Applet Life Cycle 8.2 appletviewer tool 8.3 Applet HTML Tags 8.4 Passing parameters to Applet 8.5 repaint() and update() method	Create Applets Create GUI within Applet	Use of ICT	Complete reference Java by Herbert Schildt(5th edition	4

Name: Prof. Pandit S.P.
Class: T.Y.Bsc(Comp.sci)
Course Name: Object Oriented Software Engineering   **Sem -III**
Course code: CS-336

Course Outcome (CO)

Upon successful completion of this lab course, students will be able to

1) CO1: Develop a working understanding of object-oriented analysis and design processes, Construction.
2) CO2: Understands of the Classes, Attributes and Relationships and develop the basic structure model.
3) CO3: Understands of the Advanced Classes, Interfaces, packages and develop Advanced structural Model i.e. Object Diagram
4) CO4: Understands Use Case, Activity, Sequence, State Chart Interaction Scenarios and Draw a respective diagram.
5) CO5:Develop an understanding of the application of OOAD practices from a software project management perspective

Course Cs-336 T.Y.B.Sc.(CS)	Course Specific Outcome CSO	Methodology	Reference Books	No. of Lectures
1. Object Oriented Concepts and Principles 1.1 What is Object Orientation? - Introduction, Object, Classes and Instance, Polymorphism, Inheritance 1. 2 Object Oriented System Development- Introduction, Function/Data Methods (With Visibility), Object Oriented Analysis, Object Oriented Construction 1.3 Identifying the Elements of an Object Model 1.4 Identifying Classes and Objects 1.5 Specifying the Attributes (With Visibility) 1.6 Defining Operations 1.7 Finalizing the Object Definition	1. Student should learn about object Orientation, Object Oriented Analysis, Design and construction. 2. Identify Classes, objects, attributes and operations. Identify the object model.	Constructivism	Grady Booch, James Rambaugh, The Unified Modeling Language User/Reference Guide, Pearson Education INC	4
2. Introduction to UML 2.1 Concept of UML 2.2 Advantages of UML	1. Learn about UML - Visualizing a Class - Modeling a Class - Modeling	Demonstration	Grady Booch, James Rambaugh, The	02

	the Relations among Classes - Test cases. 2. Understand basic building blocks of UML.		Unified Modeling Language User/Reference Guide, Pearson Education INC	
3. Basic Structural Modeling 3.1 Classes 3.2 Relationship 3.3 Common Mechanism 3.4 Class Diagram (Minimum three examples should be covered)	1) Understand classes and relationship. Develop structural modeling. i.e. Class Diagram	Constructivism	Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC	05
4. **Advanced Structural Modeling** 4.1 Advanced Classes 4.2 Advanced Relationship 4.3 Interface 4.4 Types and Roles 4.5 Packages 4.6 Object Diagram (Minimum three examples should be covered)	1. Understand advanced classes, relationships, Interfaces and packages. 2. Develop Advanced Structural Model. i.e. Object Diagram.	Constructivism	Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC	07
5. **Basic Behavioral Modeling** 5.1 Interactions 5.2 Use Cases and Use Case Diagram with stereo types (Minimum three examples should be covered) 5.3 Interaction Diagram (Minimum two examples should be covered) 5.4 Sequence Diagram (Minimum two examples should be covered) 5.5 Activity Diagram (Minimum two examples should be covered) 5.6 State Chart Diagram (Minimum two examples should be covered	1: Understand Use Case Diagrams, Use Case Scenarios - Use Case Relationships Understanding the Activity Diagrams - Applying Use Cases. 2: Understand Sequence Diagrams - Object Lifeline - Activation Message, Purpose of State Chart Diagram - State and Composite State -Event - Transition. 3: Understand the Interaction Diagrams - Purpose of Collaboration Diagrams - Similarities and Differences between Sequence and Collaboration Diagrams - Notations of a Collaboration Diagram. 4: Draw/develop the use case model, sequence	Constructivism & Use of ICT	Craig Larman, Applying UML and Patterns Pearson Education INC 4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill	09

	diagram, activity diagram, state chart diagram and collaboration diagram etc.			
6. Object Oriented Analysis 6.1 Iterative Development and the Rational Unified Process 6.2 Inception 6.3 Understanding Requirements 6.4 Use Case Model From Inception to Elaboration 6.5 Elaboration	1: Understands Generic Components of the OO Design Model. 2: Apply the system design processes like Partitioning the Analysis Model, Concurrency and Sub System Allocation, Task Management Component, The Data Management Component, The Resource Management Component, Inter Sub System Communication.	Constructivism	Craig Larman, Applying UML and Patterns Pearson Education INC 4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill	06
7. Object Oriented Design 7.1 The Booch Method, The Coad and Yourdon Method and Jacobson Method and Raumbaugh Method 7.2 The Generic Components of the OO Design Model 7.3 The System Design Process - Partitioning the Analysis Model, Concurrency and Sub System Allocation, Task Management Component, The Data Management Component, The Resource Management Component, Inter Sub System Communication 7.4 Object Design Process	1: Understands OODesign Model, Design Processes etc.	Constructivism	Craig Larman, Applying UML and Patterns Pearson Education INC 4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill	04
8. Architectural modeling 8.1 Component 8.2 Components Diagram (Minimum two examples should be covered) 8.3 Deployment Diagram (Minimum two examples	1: Understands and draw Component Diagram, Deployment Diagram.	Constructivism & Use of ICT	Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC	06

should be covered) 8.4 Collaboration Diagram (Minimum two examples should be covered)				
9. Object Oriented Testing 9.1 Object Oriented Testing Strategies 9.2 Test Case Design for Object Oriented Software 9.3 Inter Class Test Case Design	1: Implements the Object Oriented Testing Strategies. 2: Generate and use the Test Case Design, Inter class test case design for a system.	Constructivism	Bennett, Simon, Object Oriented Analysis and Design McGraw Hill	05

References.

1. Grady Booch, James Rambaugh, The Unified Modeling Language User/Reference Guide, Pearson Education INC

2. Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC

3. Craig Larman, Applying UML and Patterns Pearson Education INC

4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill

Name: Dipali Yogesh Jadhav
Class: T.Y.Bsc(Comp.sci)
Course Name: Operating system   **Sem -VI**
Course code: CS-341
Course outcomes:

CO 1 To understand an environment of computer system.

CO 2 Learning the working mechanism of operating system.

CO 3 Importance of operating system.

CO 4 Develop architecture for creating new system software.

CO 5 To know about various methods of operating system management.

Course **T.Y.Bsc(Comp.sci)**	Course Specific Outcome CSO	Methodology	Reference Book	No. of Lectures
Introduction Operating System Structure Simple structure, Layered approach, Micro kernels, Modules Virtual Machines – Introduction, Benefits System Boot	Understands services of operating system	Lecture	Operating System Concepts	2
Process Management Process Concept Process Scheduling Operations on Process Process creation with program using fork(), Process termination Interprocess Communication	Recognizes the numerous processes in the operating system. Learns process control architecture.	Constructive	Operating System Concepts	4
Multithreaded Programming [2] Overview Multithreading Models	Identifying an interfaces provides by operating system. Multithreaded model.	Lecture	Operating System Concepts	2
Process Scheduling Basic Concept Scheduling Criteria Scheduling Algorithms Thread Scheduling	Learns the scheduling mechanism of CPU. Process execution system.	Constructive	Operating System Concepts	8

Process Synchronization Background Critical Section Problem Semaphores Classic Problems of Synchronization	Acquire knowledge about synchronization of process. Identifies the problem area of synchronization.	Lecture	Operating System Concepts	6
Deadlocks System model Deadlock Characterization Deadlock Prevention Deadlock Detection	Understands deadlock terminology. Identifies the routes of occurring deadlock. Deadlock prevention and avoidance technique.	Lecture	Operating System Concepts	8
Memory Management Background Swapping Paging Virtual Memory Management	Learns memory management of operating system. Methods of Memory management	Lecture	Operating System Concepts	11
File System File concept Access Methods Allocation Methods Free Space Management	Recognizes concept of file handing. Understands file accessing methods.	Lecture	Operating System Concepts	7

References:

- Operating System Concepts - Siberchatz, Galvin, Gagne (8th Edition).

Dipali Yogesh jadhav
M.Sc(Comp.Sci.)
Mamasaheb Mohol College

Course Name: Compiler Construction Sem - II   **Course code: CS-342**

Course outcome

CO1.Students will be able to understand the principles and techniques used to perform translation and the fundamental concepts of translator construction.

CO2.Understand the various phases of a compiler and to develop skills in designing a compiler.

CO3. Students can use software tools and techniques which are applicable both to compilers and the implementation of system utility routines, command interpreters.

CO4. To understand issues related to memory allocation.

CO5.To constructs compilers.

Course:CS-342 T.Y.B.Sc.CS	Course Specific Outcome CSO	Methodology	Reference Book	No. of Lecture
1. Introduction 1.1 Definition of Compiler, Aspects of compilation. 1.2 The structure of Compiler. 1.3 Phases of Compiler – Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code generation, code optimization, code generation. 1.4 Error Handling 1.5 Introduction to one pass &Multipass compilers, cross compiler, Bootstrapping.	Understand the basic concepts of compiler and aspects of compilation. To know one pass &Multipass compilers, cross compiler, bootstrapping process.	Lecture	Compilers: Principles, Techniques, and Tools	5
2. Lexical Analysis(Scanner) 2.1 Review of Finite automata as a lexical analyzer, 2.2 Applications of Regular Expressions and Finite Automata (lexical analyzer, searching using RE), Input buffering, Recognition of tokens 2.3 LEX: A Lexical analyzer generator (Simple Lex Program)	Use of Finite automata as a lexical analyzer. Create simple Lex Program. Describe the programs related to compilers and the translation process.	Constructivism	Compilers: Principles, Techniques, and Tools	5
3. Syntax Analysis(Parser) 3.1 Definition , Types of Parsers	Apply an algorithm for a top-down or a	Demonstration	Principles of Compiler	20

3.2 Top-Down Parser – 3.3 Recursive Descent Parsing : Definition 3.4 Predictive [LL(1)]Parser(Definition, Model) 3.5 Bottom-Up Parsers 3.6 Operator Precedence Parser - Basic Concepts 3.7 Shift Reduce Parser 3.8 LR Parser 3.9 YACC	bottom-up parser construction; construct a parser for a small context-free grammar.		Design 3.LEX & YACC (O'reilly Publication)	
4. Syntax Directed Definition 4.1Syntax Directed Definitions(SDD) 4.2 Evaluation Orders for SDD's 4.3 Application of SDT 4. 4 Translation Schemes	Understand the role of a semantic analyzer and type checking. Create a syntax-directed definition and an annotated parse tree.	Constructivism	1. Compilers: Principles, Techniques, and Tools	8
5. Memory Allocation 5.1 Memory allocation – static and dynamic memory allocation, 5.2 Memory allocation in block structure languages, Array allocation and access.	Understand the memory allocation methods.	Constructivism	1. Compilers: Principles, Techniques, and Tools	2
6. Code Generation and Optimization 6.1 Compilation of expression 6.2 Code Optimization – Optimizing transformations compile time evaluation, elimination of common sub expressions, dead code elimination, frequency reduction, strength reduction 6.3 Three address code 6.4 Definition of basic block, Basic blocks And flow graphs 6.5 Directed acyclic graph (DAG) representation of basic block	Understand and design code generation schemes. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.	Constructivism	1. Compilers: Principles, Techniques, and Tools. 2. Principles of Compiler Design	8

6.6 Issues in design of code generator				

References:-

1. Compilers: Principles, Techniques, and Tools ,Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman

2. Principles of Compiler Design By : Alfred V. Aho, Jeffrey D. Ullman (Narosa Publication

House)

3. LEX & YACC (O'reilly Publication)

Mrs.Rajnigandha Y. Ingole

M.Sc.(Computer Science)

Mamasaheb Mohol College Pune-38

Course:-Computer Network-II

Name of Teacher:- Prof. D.D Pokhalekar

Class:- T.Y.B.Sc(CS)   **Course Code - 343**

Course Outcomes:-

CO1) Basic Networking concepts .

CO2) Understand wired and wireless networks, its types, functionality of layer

CO3) Understand importance of network security and cryptography.

CO4) Learns how data flows in the network.

CO5) Various social issues of network security.

Course	Course Specific Outcome	Methodology	References	No. of lectures
Wired LANs: IEEE Standards Data Link Layer, Physical Layer Backbone Networks	Frame Format, Frame Length Encoding and Decoding, 10Base5, 10Base2, – Bridged Ethernet, Switched Ethernet	Constructive	Data Communication and Networking by Behrouz Forouzan,	9
Wireless LAN: IEEE 802.11 Architecture Bluetooth Architecture	Basic Service Set, Extended Service Set, Piconet, scatternet	Use of ICT tools	Data Communication and Networking by Behrouz Forouzan,	2
The Network Layer: Design Issues Store-and-forward packet switching Logical Addressing IPV4 Protocol Datagram	Address Space, Notations, Classful Addressing, Subnetting, Supernetting, Classless Addressing, Fragmentation, Checksum, Congestion Control – Definition, Factors of Congestion	Constructive	By Andrew Tanenbaum, Pearson Education.[4th Edition]	10
Address Mapping: Protocol, DHCP Static Address Allocation	Cache Memory, Packet Format, Encapsulation, ARP, RARP , BOOTP, Manual and automatic Configuration	Constructive	Data Communication and Networking by Behrouz Forouzan,	4
The Transport Layer: Process-to-Process Delivery, User Datagram Protocol, Transmission Control Protocol	Multiplexing and De-multiplexing, Connectionless Vs Connection-Oriented, TCP Services – Process toProcess Communication, Stream Delivery Service, sending, TCP Features – Numbering System, Byte Number, Sequence Number, Acknowledgement Number	Constructive	Data Communication and Networking by Behrouz Forouzan,	6

Course	Course Specific Outcome	Methodology	References	No. of lectures
The Application Layer: Domain Name System E-MAIL Architecture, File Transfer Protocol WWW Architecture	Domain, Name Space, Distribution of Name Space, DNS in the Internet, Resolution, User Agent, Message Transfer Agent-SMTP, WEB Documents, HTTP Transaction, Persistent and Non persistent	Constructive	Data Communication and Networking by Behrouz Forouzan,	7
Network Security: Introduction, Message confidentiality, Cryptography Encryption Model, Communication Security Firewalls	Security Services- Message-Confidentiality, Integrity, Cryptographic Principles, Web Security Threats, Secure Naming, DNS Spoofing, Secure DNS, Social Issues Privacy, Anonymous Remailers	Constructive	By Andrew Tanenbaum, Pearson Education.[4th Edition]	10

Reference Books: 1) Computer Networks by Andrew Tanenbaum, Pearson Education.[4th Edition]
2) Data Communication and Networking by Behrouz Forouzan, TATA McGraw Hill.
.[4th

Class: TY BSc(CS)

Course Title : Programming in java –II

Course Code : 345

Course Outcome

CO1: Demonstrate approaches for performance and effective coding

CO2: Develop Java client/server applications

CO3: Write multi-threaded Java applications.

CO4: Update and retrieve the data from the databases using SQL.

CO5: Develop server side programs in the form of Servlets and JSP.

Course TYBSc(CS)	Course Specific Outcome CSO	Methodology	Reference Book	No of Lectures
1. Collection [6] 1.1 Introduction to the Collection framework 1.2 List – ArrayList, LinkedList and Vector,Stack,Queue 1.3 Set - HashSet, TreeSet, and LinkedHashSet 1.4 Map – HashMap, LinkedHashMap, Hashtable and TreeMap 1.5 Interfaces such as Comparator, Iterator, ListIterator, Enumeration	Use different collection classes Implement collection interfaces	Constructive	Complete reference Java by Herbert Schildt(5th edition)	6
2. Database Programming [10] 2.1 The design of jdbc, jdbc configuration 2.2 Types of drivers 2.3 Executing sql statements, query execution 2.4 Scrollable and updatable result sets 2.5 Metadata – DatabaseMetadata, ResultSetMetadata 2.6 Transactions – commit(), rollback(), SavePoint (Database : PostgreSQL)	Understand basic design of JDBC drivers Execute SQL statements	Constructive	Core Java Volume-II- Advanced Features, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press	10

3. Servlet [12] 3.1 Introduction to Servlet and Hierarchy of Servlet 3.2 Life cycle of servlet 3.3 Tomcat configuration (Note: Only for Lab Demonstration) 3.4 Handing get and post request (HTTP) 3.5 Handling a data from HTML to servlet 3.6 Retriving a data from database to servlet 3.7 Session tracking – User Authorization, URL rewriting, Hidden form fields, Cookies and HttpSession	Implement life cycle of servlet Dealing with cookies Track session	Constructive	Servlets And JSP by O'Rally	12
4. JSP [10] 4.1 Simple first JSP program 4.2 Life cycle of JSP 4.2 Implicit Objects 4.3 Scripting elements – Declarations, Expressions, Scriplets, Comments 4.4 JSP Directives – Page Directive, include directive 4.5 Mixing Scriplets and HTML 4.6 Example of forwarding contents from database to servlet, servlet to JSP and displaying it using JSP scriplet tag	Add dynamic contents in web page Handle different tags and directive Create simple web application	Constructive	Servlets And JSP by O'Rally	10
5. Multithreading [6] 5.1 What are threads? 5.2 Life cycle of thread 5.3 Running and starting thread using Thread class 5.4 Thread priorities 5.5 Running multiple threads 5.6 The Runnable interface 5.7 Synchronization and interthread communication	Dealing with errors Handle Exception Create Exception	Use of ICT	Complete reference Java by Herbert Schildt(5th edition	4
6. Networking [4] 6.1 Networking basics – Protocol, Addressing, DNS, URL, Socket,	Understand java.net package Implement TCP/IP and	Constructive	Complete reference Java by Herbert	6

Port	Datagram protocol		Schildt(5th edition	
6.2 The java.net package – InetAddress, URL, URLConnection class				
6.3 SocketServer and Socket class				
6.4 Creating a Socket to a remote host on a port (creating TCP client and server)				
6.5 Simple Socket Program Example				

Course Name: Computer Graphics   **Course code: CS-346**

Course outcome

CO1. Identify and explain the core concepts of computer graphics.

CO2. Understand how graphics objects are represented in Computer.

CO3.Understand how graphics system in a computer supports presentation of graphics information.

CO4. Understand how to manipulate graphics object by applying different transformations.

CO5. Apply graphics programming techniques to design and create computer graphics scenes.

CO6.To provides the programmer's perspective of working of computer graphics.

CO7. Analyze the fundamentals of animation, virtual reality, underlying technologies and principles.

Course:CS-342 T.Y.B.Sc.CS	Course Specific Outcome CSO	Methodology	Reference Books	No. of Lecture
1.Introduction to Computer graphics 1.1 Introduction to computer graphics & graphics systems 1.2 Components of Computer Graphics Representation, Presentation , Interaction and Transformations 1.3 Applications of Computer Graphics 1.3 Pixel/Point ,Raster v/s Vector ,RGB color model, intensity 1.4 Programming essentials – event driven programming. OpenGL library	Understand the basic concept of computer graphics. Applications of computer graphics Event driven programming and working with OpenGL library .	Chalk and talk	Computer Graphics principles (2nd Ed.) – Pearson Education.	4
2. Input devices and Interaction tasks 2.1Logical Interaction – Locator, valuator , pick and choice; 2.2 Physical devices used for interaction – keyboard, mouse, trackball,spaceball, tablets, light pen, joy stick, touch panel, data glove; 2.4 Keyboard , Mouse	Understand working of physical devices used for interaction like keyboard, mouse, trackball, spaceball, tablets, light pen, joy stick, touch panel, data glove. Implementing GUI in Open GL.	Constructivism	Computer Graphics principles (2nd Ed.) – Pearson Education.	4

interaction in OpenGL 2.5 Graphical User Interfaces-cursors , radio buttons, scroll bars, menus, icons 2.6 Implementing GUI in open GL				
3. Presentation and Output devices 3.1 Presentation Graphics frame buffer, display file, lookup table; 3.2 Display devices, Random and Raster scan display devices; CRT, 3.3 Hardcopy devices - Plotters and Printers	Presentation of graphics using frame buffer, display file, lookup table. Understand the working of various display devices, Random and Raster scan display devices, CRT, and hardcopy devices - Plotters and Printers	Constructivism	Computer Graphics principles (2nd Ed.) – Pearson Education.	4
4. Raster Scan Graphics 4.1 Line drawing algorithms; DDA algorithm, Bresenham's line drawing algorithm, Circle generation algorithm; 4.2 Scan conversions- Generation of the Display, Image compression 4.3 Displaying Lines and characters 4.3 Polygon filling -Scan converting polygons, fill algorithms, Boundary fill algorithm, flood fill algorithm	Understand how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition. Implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.	Demonstrative	Computer Graphics (C version 2nd Ed.)– Pearson education	10
5. Transformations 5.1 Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, Reflection, shear 5.2 Transformation of points, lines, parallel lines, intersecting lines. Viewing pipeline 5.3 Window to viewport co-	Understand the concept of basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, Reflection, shear. Provide an understanding of mapping from a world	Demonstrative	Mathematical Elements for Computer Graphics (2nd Ed.)" – TMH Computer Graphics (C version 2nd Ed.) – Pearson education	7

ordinate transformation. Setting window and viewport in OpenGL.	coordinates to device coordinates.			
6 Clipping 6.1 clipping operations , point clipping , 6.2 Line clipping; Cohen Sutherland algorithm, Midpoint subdivision algorithm, Cyrus beck algorithm; 6.3 Polygon clipping , Sutherland Hodgman algorithm, Weiler-Atherton Algorithm	Understand clipping operations like point clipping and Line clipping. Implement different clipping algorithm in graphics application.	Constructivism	Computer Graphics using OpenGL, PHI Learning	7
7 3D transformation & viewing 7.1 3D transformations: translation, rotation, scaling & other transformations; 7.2 Three dimensional viewing, Parallel and Perspective projections, 7.3 View Volumes and General Projection Transformations. 7.4 3 D clipping	Create effective OpenGL programs to solve graphics programming issues, including 3D transformation. Describe the importance of viewing and projections. Use of three dimensional viewing, Parallel and Perspective projections.	Demonstrative	Computer Graphics using OpenGL, PHI Learning	6
8 Hidden surfaces Elimination 8.1 Depth comparison, A-buffer algorithm, Back face detection; Depth -Buffer 8.2 Scan-line Method - BSP tree method, the Painter's algorithm, Area-subdivision algorithm	Understand depth comparison and depth buffer Use of scan-line method - BSP tree method, the Painter's algorithm, Area-subdivision algorithm.	Consturctivism		4

Text Books: 1. Hearn, Baker – "Computer Graphics (C version 2nd Ed.)" – Pearson education

2. F. S. Hill, Stephen Kelly, Computer Graphics using OpenGL, PHI Learning

3. David F. Rogers - Procedural Elements of Computer Graphics, Tata McGRAw Hill

Reference Books:

4. Foley, Vandam, Feiner, Hughes – "Computer Graphics principles (2nd Ed.) – Pearson Education.

5. W. M. Newman, R. F. Sproull – "Principles of Interactive computer Graphics" – TMH.

6. D. F. Rogers, J. A. Adams – " Mathematical Elements for Computer Graphics (2nd Ed.)" – TMH

7. Z. Xiang, R. Plastock – " Schaum's outlines Computer Graphics (2nd Ed.)" – TMH

Mrs.Rajnigandha Y. Ingole

M.Sc.(Computer Science)

MamasahebMohol College Pune-38