Ujjain Engineering College, Ujjain

Syllabus

Energy & Environmental Engineering

HS3301

B.Tech III sem (CS Branch)

WEF: July 2019

Course objective: The objective of this course is to provide an introduction to energy systems and renewable energy sources with emphasis on alternative energy sources and their technology and applications. The course objective also includes introduction to ecosystems, biodiversity, environmental pollution and social issues.

Unit I : Introduction to energy science

Introduction to energy sources, sustainability and the environment. Fossil fuels (Coal, Oil and Natural gas), Past ,present and future scenario. Remedies and alternatives for fossil fuels. Biomass, Wind, Solar, Tidal and Hydrogen Energies.

Unit II : Environment & Ecosystems

Lithosphere, Hydrosphere and Atmosphere. Concept of an ecosystem, structure and function of an ecosystem-producers, consumers and decomposers. Energy flow in the ecosystems, ecological succession. Food chains, food webs and Ecological pyramids. Examples of Grass land and Aquatic ecosystems.

Unit III: Biodiversity and its conservation

Introduction to biodiversity, genetic ,species and ecosystem diversity. Threats to biodiversity, conservation of biodiversity- In situ and Ex situ conservation of biodiversity.

Unit IV: Pollution

Definition, causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Noise pollution & Marine pollution. Solid waste management. Role of an individual in prevention of pollution.

Unit V : Social issues and Acts

Environmental ethics, Issues and possible solutions. Climate change , Global warming , Acid rains and Ozone layer depletion, Nuclear accidents. Environment Protection Acts- Air (Prevention and control of pollution) act, Water (Prevention and control of pollution) act.

Course outcome: Students after completion of course will have understanding and knowledge of various types of energy sources, ecosystems, biodiversity, pollution and social issues which are essential for sustainability.

References :

1. B.K.Sharma, Environmental Chemistry, Goel publishing house.
2. S.V.S. Rana , Essentials of Ecology and Environment, PHI Publications.
3. S V A Kumar, Energy , Environment, Ethics and Society, TMH Publications.
4. A. K. De, Environmental Chemistry, Wiley Eastern Ltd.
5. R.K.Trivedi, Hand book of environmental laws, rules, guidelines, compliances and standards, Vol I & II, Env. Media
6. Benny Joseph, Environmental Studies, Mc Graw Hill Education India, Pvt. Ltd., New Delhi
7. J. P. Sharma, Environmental Studies, University Science Press, New Delhi
8. Boyl, Godfrey, Bob Everett and Janet Ramage (Eds.2004),Energy Systems and Sustainability: Power for a sustainable future. Oxford University Press.
9. Schaeffer, John (2007) Real Goods Solar living source book: The complete guide to Renewable energy technologies and sustainable living, Gaiam.

B TECH Semester **COMPUTER SCIENCE & Engg.**							
COURSE CONTENTS (UEC SCHEME)							
CS3301	**Data Structures**	**L**	**T**	**P**	**C**	**Max. Marks**	**Min. Marks**
Duration	5 Hours	2	1	2	4	70	22

Course Descriptions:

Study of advanced programming topics focused on logical structures of data. Data Structure Operations. Topics include linked lists, stacks, trees, queues, graphs and analysis of efficiency. Also covers searching, sorting and hashing techniques. Linked Lists; Type of lists; Operations and Applications; Stacks and Queues; Operations and Applications; Trees, Search trees and Heaps; Multiway Trees and Graphs; Searching and Hashing.

Course Objectives:

- To learn the fundamentals of Data Structures, features and applications.
- To learn the Principles of Data Structures.
- To understand and gain knowledge on Linear and Non-linear Data Structures.
- To apply different sorting and hashing techniques.
- To analyze and identify suitable data structure for computational problem solving

Syllabus:

UNIT-I: Linked Lists

Introduction To Data Structures, Pointers, Operations, Linked List definition, Single Linked Lists, Circular Linked List, Doubly Linked List, Circular Doubly Linked List, Application of Linked Lists.

UNIT-II: Stacks and Queues

Stacks: Stack operations, Stack Linked List, Implementation, Stack applications.
Queues: Queue operations, Queue Linked List design, Queue applications.

UNIT-III: Trees, Search Trees And Heaps

Trees: Tree concepts, Binary Trees. Binary Search Trees (BST): Basic concepts, BST operations, BST applications. AVL Search Trees: Basic concepts, AVL Tree implementations. Heaps: Basic concepts, Heap implementation, Heap applications.

UNIT-IV: Multiway Trees and Graphs

Multiway Trees: B-Trees, Simplified B-Trees, B-Tree variations. Graphs: Basic concepts, Operations, Graph storage structures, Graph algorithms - Create graph, Insert vertex, Delete vertex, Retrieve vertex, Depth-first traversal, Breadth-first traversal.

UNIT-V: Sorting and Hashing

Internal Sorting: Quick Sort, Shell Sort, Merge Sort, Heap Sort. External Sorting: Introduction, External storage device and sorting, Balanced Merge. Hashing: Introduction, Hash Table structure, Hash functions, Linear Open Addressing, Chaining, Applications.

Course Outcomes:

At the end of the course students will:

- Design solutions for complex engineering problems using linear and non-linear data structures.

- Develop solutions for Complex computational problems by conducting explorative analysis.
- Apply appropriate data structure to provide solutions for real time problems by using the C Language.
- Apply contextual knowledge of data structures to design applications for societal applications like payroll systems, web applications, banking and financial systems.

Text Books:

i. Richard Gileberg and Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, Cengage Learning, Second Edition, 2007.

ii. G.A.V. Pai, "Data Structures and Algorithms", Tata McGraw Hill, Second Edition, 2009.

Reference Books:

i. Debasis Samanta, Classic Data Structures, PHI Learning, Second Edition, 2009.

ii. Aaron M. Tenenbaum, Yedidyah Langsam, and Moshe J. Augenstein, Data Structures Using C, Pearson Education, 2005.

LIST OF EXPERIMENTS:

1. Write a C program that uses functions to perform the following:
   a) Create a singly linked list of integers.
   b) Delete a given integer from the above linked list.
   c) Display the contents of the above list after deletion.
2. Write a C program that uses functions to perform the following:
   a) Create a doubly linked list of integers.
   b) Delete a given integer from the above doubly linked list.
   c) Display the contents of the above list after deletion.
3. Write a C program that uses stack operations to convert a given infix expression into its postfix Equivalent, Implement the stack using an array.
4. Write a C program that uses functions to perform the following:
5. Create a binary search tree of characters, and traverse the above Binary search tree recursively in Postorder and Inorder
6. Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order: a) Insertion sort b) Merge sort
7. Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order: a) Quick sort b) Selection sort
8. Write a C program to implement all the functions of a dictionary (ADT) using hashing.
9. Write C programs for implementing the following graph traversal algorithms:

   a)Depth first traversal b)Breadth first traversal

10. Write a C Program to check whether two given lists are containing the same data.
11. Write a C program to find the largest element in a given doubly linked list.
12. Write a C program to reverse the elements in the stack using recursion.
13. Write a C program to implement stack using linked list.
14. Write a C program to count the number of nodes in the binary search tree.

*These are the General List of Experiments & the list of Experiments may vary as per course Instructor.

BTECH III Semester **COMPUTER SCIENCE & Engg.**							
COURSE CONTENTS (UEC SCHEME)							
CS3302	**Digital systems**	**L**	**T**	**P**	**C**	**Max. Marks**	**Min. Marks**
Duration	3 Hours	2	1	2	4	70	22

Course Description:
This course introduces the basic logic functions, components and methodologies used in the design of digital systems. Digital electronic topics will include the basic logic gates, Boolean algebra, number systems, digital arithmetic, combinational logic circuits, multiplexers, decoders and flip-flops and registers. Digital system applications will include counters, magnitude comparators, Analog-to-Digital and Digital-to-Analog conversions, Clocks and synchronization CMOS, PMOS logic circuits
Combinational logic design, TDM , sampling theorem, PCM, introduction to BPSK and BFSK, shannon's theorem for channel capacity, IC 555 and multivibrators.

Course objective:
Students will try to learn:

- To understand number representation and conversion between different representation in digital electronic circuits.
- To analyze logic processes and implement logical operations using combinational logic circuits.
- To understand characteristics of memory and their classification.
- To understand concepts of sequential circuits and to analyze sequential systems in terms of state machines.
- To understand concept of Programmable Devices, PLA, PAL.

Syllabus:
Unit 1: Review of number systems and number base conversions. Binary codes, Boolean algebra, Boolean functions, Logic gates. Simplification of Boolean functions, Karnaugh map methods, SOP-POS simplification, NAND-NOR implementation.

Unit 2: Combinational Logic: Half adder, Half Subtractor, Full adder, Full Subtractor, look-ahead carry generator, BCD adder, Series and parallel addition, Multiplexer – de-multiplexer, encoder- decoder, arithmetic circuits, ALU

Unit 3 : Sequential logic: flip flops, D,T, S-R, J-K Master- Slave, racing condition, Edge & Level triggered circuits, Shift registers, Asynchronous and synchronous counters, their types and state diagrams. Semiconductor memories, Introduction to digital ICs 2716, 2732 etc. & their address decoding. Modern trends in semiconductor memories such as DRAM, FLASH RAM etc. Designing with ROM and PLA.

Unit 4: Introduction to A/D & D/A convertors & their types, sample and hold circuits, Voltage to Frequency & Frequency to Voltage conversion. Multivabrator :Bistable, Monostable, Astable, Schmitt trigger, IC 555 & Its applications. TTL, PMOS, CMOS and NMOS logic. Interfacing between TTL to MOS.

Unit 5 : Introduction to Digital Communication: Nyquist sampling theorem, time division multiplexing, PCM, quantization error, introduction to BPSK & BFSK modulation schemes. Shannon's theorem for channel capacity.

Course outcomes:

After successful completion of the course student will be able to

- Develop a digital logic and apply it to solve real life problems.
- Analyze, design and implement combinational logic circuits.
- Classify different semiconductor memories.
- Analyze, design and implement sequential logic circuits.

Reference books:

1. Morris Mano, Digital Circuits & Logic Design, PHI
2. Gothman, Digital Electronics, PHI
3. Tocci, Digital Electronics, PHI
4. Mavino& Leach, Digital Principles & Applications, PHI
5. Taub and schilling, Digital Integrated electronics.
6. Simon Haykin, Introductionto Analog& Digital Communication, Wiley.
7. Lathi B.P., Modern analog& digital communication , Oxford University.

List of Experiments:

1 To study & verify different types of logic gates.
2 To verify the De-Morgan's theorem and Boolean rules.
3 To study & verify universal logic gates.
4 To design the logic circuit of Even and Odd Parity Generator and Checker.
5 To study & design logic circuit for Half Adder and Full Adder.
6 To study & design logic circuit for Half Subtractor and Full Subtractor.
7 To study & verify the different types of Flip-Flop.
8 To study & verify the MUX and DE-MUX.
9 To study & verify the different Multivibrator.
10 To study & verify the Full Adder using 4x1 and 8x1 multiplexer.
11 To study & verify IC 555 Timer.

*These are the General List of Experiments & the list of Experiments may vary as per course Instructor.

BTECH III Semester **COMPUTER SCIENCE & Engg.**							
COURSE CONTENTS (UEC SCHEME)							
CS 3303	**Discrete Structure**	**L**	**T**	**P**	**C**	**Max. Marks**	**Min. Marks**
Duration	4 Hours	3	1	0	4	70	22

Course Description

To identify the basic properties of Logic, Relation and Function, graphs and trees to model simple applications and to get familiar and understand the fundamental notions in discrete mathematics. Distinguish between the notion of discrete and continuous mathematical structures.

Course Objectives

- To get familiar and understand the fundamental notions in discrete mathematics
- To understand and demonstrate the basic concept of an algorithm and its application in combinatorial mathematics
- To identify the basic properties of graphs and trees and model simple applications

Syllabus:

Unit – I

Set Theory, Relation, Function, Theorem Proving Techniques : Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Relation: Definition, types of relation, composition of relations, Pictorial representation of relation, Equivalence relation, Partial ordering relation, Job-Scheduling problem Function: Definition, type of functions, one to one, into and onto function, inverse function, composition of functions, recursively defined functions, pigeonhole principle. Theorem proving Techniques: Mathematical induction, Proof by contradiction.

Unit – II

Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Unit – III

Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Normal Forms, Universal and existential quantifiers. Introduction to finite state machine Finite state machines as models of physical system equivalence machines, Finite state machines as language recognizers

Unit – IV

Graph Theory: Introduction and basic terminology of graphs, Planer graphs, Multigraphs and weighted graphs, Isomorphic graphs, Paths, Cycles and connectivity, Shortest path in weighted graph, Introduction to Eulerian paths and circuits, Hamiltonian paths and circuits, Graph coloring, chromatic number, Isomorphism and Homomorphism of graphs.

Unit – V

Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded and complemented lattices. Combinatorics: Introduction, Permutation and combination, Binomial Theorem, Multimonial Coefficients Recurrence Relation and Generating Function: Introduction to Recurrence Relation and Recursive algorithms , Linear recurrence relations with constant coefficients, Homogeneous solutions, Particular solutions, Total solutions, Generating functions, Solution by method of generating functions.

Course Outcomes:

At the end of the course students will:

- Understand sets, relations, functions and discrete mathematical structures
- Apply Propositional logic and first order logic to solve problems
- Formulate and solve tree and graph problems
- Formulate and Count discrete event occurrences and solve recurrence relations

TEXT BOOKS:

- C.L.Liu And D.P.Mohapatra, " Elements Of Discrete Mathematics: A Computer Oriented Approach", Mcgraw Hill, Third Edition, 2012.
- Kenneth H. Rosen, "Discrete Mathematics And Its Applications" Mcgraw Hill, Seventh Edition, 2012 (Indian Adaptation By Kamala Krithivasan, Iit Madras).

REFERENCE:

- R. Balakrishnan and K. Ranganathan, "A Text Book Of Graph Theory", Springer
- Thomas Koshy, "Discrete Mathematics with Applications", Elsevier, 2009.
- Gary Haggard, John Schlipf, and Sue Whitesides, “Discrete Mathematics for Computer Science”, Cengage Learning Publisher, 2005.
- B. Bollobás, “Modern Graph Theory”, Springer, New York 1998

RoyDeep

BTECH III Semester **COMPUTER SCIENCE & Engg.**							
COURSE CONTENTS (UEC SCHEME)							
CS3304	**Objected Oriented Programming Using JAVA**	**L**	**T**	**P**	**C**	**Max. Marks**	**Min. Marks**
Duration	5 Hours	2	1	2	4	70	22

Course Description :
Java is a platform-independent object-oriented programming language used to create stand-alone applications and applets for the World Wide Web. This course gives the student a basic understanding of the Java language and its role in the Object Oriented World. The student creates simple applications and applets.

Course Objectives:
This course aims to :

- Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods, constructor etc.
- Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc and exception handling mechanisms.
- Understand the principles of inheritance, packages and interfaces.

Syllabus:
Unit - I Introduction to Object Oriented Programming:
Object Oriented Concepts, Merits of Object Oriented Technology. Abstraction, Encapsulation, Information Hiding. Object Model: definition, State, Behaviour, Identity and messages. Concepts of object initialization, constructors, constructor overloading. Access modifiers: Class attributes and methods.

Unit - II:
Introduction to Java classes and objects: Java features: Java syntax, dada types, dada type conversions, control statements, operators and their precedence. Introduction to Class: Instance members and member functions. String Handling, Wrapper classes: Arrays and Vectors.

Unite - III Inheritance and Polymorphism: Class relationships: Inheritance and its types, merits and Demerits. Association inheritance, Polymorphism: Dynamic method dispatch, Runtime polymorphism, Abstract classes, Interfaces and packages.

Unite - IV Exception Handling and Multithreading: Exception: Need for exceptions, Checked Unchecked exceptions, creating exceptions. Multithreading: Introduction, Priorities and scheduling, Thread Synchronization and its life cycle.

Unite - V Java I/O, Applets and Event Handling: Basic concept of streams I/O stream & reader-writer classes. File handling. Applet and its Life Cycle, Basic GUI elements, Event Delegation Model and event handling.

Pradeep

Course Outcomes :

After the completion of this course, students will be able to:

After the completion of this course, students will be able to:

- Understand basic concepts of object oriented programming.
- Identify classes, objects, members of a class and relationships among them needed for a specific problem
- Demonstrate the concepts of polymorphism and inheritance
- Understand various error handling techniques using exception handling.
- Apply the knowledge gained for their project work as well as to develop some GUI applications.

Text books :

i. Let us java, by Yashvant kanetkar, 9th Edition

References Book :

i. Core JAVA Vol-1 , by Cay S.Horstmann, Pearson Education, 9th Edition.
ii. The complete Reference , by Harbert Schildt, Tata McGrow Hill, 7th Edition.
iii. The Object Primer , by Scott W Amber, Cambridge , 3rd Edition.
iv. Object Oriented Programming,by Timothy, Budd, Pearson Education,*2nd* Edition ..
v. Head First Java ,by Kathy Sierra, *Bert Bates, 2nd* Edition .

Lab experiments:

1. Lab Setup and Introduction to Java Fundamentals.
2. Create a class named **'Student'** with String variable **'name'** ,integer variable **'roll_no'** to store roll number of student ,integer variable **'Java'** to store marks for Java subject nd **'maths'** to store marks for Maths subject.Assign the value of roll_no as '2' and that f name as 'your own name'(For Eg. Bharti) , marks for java and maths 80,90 respectively by creating **object** of the class Student.
3. Create a class named **'BankAccount'** with String variable **'name'** ,integer variable **'balance'** and **'account number'** .A person can deposit and withdraw money from account and can get the status of balance. Write a java program to initialize data using constructors.
4. Write a program to create a '**Room'** class, the attributes of this class is **roomno, roomtype, roomarea** . In this class the member functions are **setdata** and **displaydata.** Use parametrized constructors to initialize data members .
5. A class named **'Arithmetic'** with a method named '**add'** that takes integers as parameters and returns an integer denoting their sum.A class named **'Adder'** that inherits from a superclass named **Arithmetic.** Your classes should not be be public.
6. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Tri-

angle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape

7. Write a program to implement multiple inheritances using interface in java.
8. Write a program to implement Abstract Class
9. Write a program to implement concept of packages:
    i. Same package - No subclasses
    ii. Same package - Subclasses
    iii. Different package - No subclasses
    iv. Different package - Subclasses
10. Write a program that creates a user interface to perform integer division. The user enters two numbers Num1 and Num2. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Excep- tion Display the exception message.
11. Write a program to show "HELLO JAVA " in Explorer using Applet
12. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

*These are the General List of Experiments & the list of Experiments may vary as per course Instructor.