Week	Content
1	Review of Object Oriented Programming and Java Fundamentals Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs
2	Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces
3	Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages
4	Exception Handling : Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions
5	File Handling: Byte Stream, Character Stream, File I/O Basics, File Operations
6-9	AWT and Event Handling The AWT class hierarchy ,Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT, Creating GUI applications using AWT
10-15	Swing: Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Creating GUI applications using Swing, Creating GUI applications using Swing

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Objects and classes, interfaces, exceptional handling, file handling

Discrete Structures (BHCS04) Discipline Specific Core Course - (DSC)

Credit: 06

Course Objective

The course aims to introduce the students to Boolean algebra, sets, relations, functions, principles of counting, and growth functions so that these concepts may be used effectively in other courses.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Define mathematical structures (relations, functions, sequences, series, and graphs) and use them to model real life situations.
2. Understand (trace) and construct simple mathematical proofs using logical arguments.
3. Solve class room puzzles based on counting principles.
4. Compare functions and relations with respect to their growth for large values of the input.

Detailed Syllabus

Unit 1

Introduction: Sets - finite and infinite sets, uncountable infinite sets; functions, relations, properties of binary relations, closure, partial ordering relations; counting - Pigeonhole Principle, permutation and combination; mathematical induction, Principle of Inclusion and Exclusion.

Unit 2

Growth of Functions: asymptotic notations, summation formulas and properties, bounding summations, approximation by integrals.

Unit 3

Recurrence: recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, recursion trees, Master Theorem

Unit 4

Graph Theory: basic terminology, models and types, multi-graphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and Circuits, planar graphs, graph coloring, Trees, basic terminology and properties of Trees, introduction to spanning trees.

Unit 5

Propositional Logic: logical connectives, well-formed formulas, tautologies, equivalences, Inference Theory

Practical

1. Write a Program to create a SET **A** and determine the cardinality of SET for an input array of elements (repetition allowed) and perform the following operations on the SET:
   a) ismember (a, A): check whether an element belongs to set or not and return value as true/false.
   b) powerset(A): list all the elements of power set of A.

2. Create a class SET and take two sets as input from user to perform following SET Operations:
   a) Subset: Check whether one set is a subset of other or not.
   b) Union and Intersection of two Sets.
   c) Complement: Assume Universal Set as per the input elements from the user.
   d) Set Difference and Symmetric Difference between two SETS
   e) Cartesian Product of Sets.
3. Create a class RELATION, use Matrix notation to represent a relation. Include functions to check if the relation is Reflexive, Symmetric, Anti-symmetric and Transitive. Write a Program to use this class.
4. Use the functions defined in Ques 3 to check whether the given relation is:
   a) Equivalent, or
   b) Partial Order relation, or
   c) None
5. Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.
6. Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.
7. Write a Program that generates all the permutations of a given set of digits, with or without repetition. (For example, if the given set is {1,2}, the permutations are 12 and 21). (One method is given in Liu)
8. Write a Program to calculate Permutation and Combination for an input value n and r using recursive formula of $^{n}C_{r}$ and $^{n}P_{r}$.
9. For any number n, write a program to list all the solutions of the equation $x_1 + x_2 + x_3 + \ldots + x_n = C$, where C is a constant ($C<=10$) and $x_1, x_2, x_3, \ldots, x_n$ are nonnegative integers using brute force strategy.

10. Write a Program to accept the truth values of variables x and y, and print the truth table of the following logical operations:

   a) Conjunction
   b) Disjunction
   c) Exclusive OR
   d) Conditional
   e) Bi-conditional
   f) Exclusive NOR
   g) Negation
   h) NAND
   i) NOR

11. Write a Program to store a function (polynomial/exponential), and then evaluate the

polynomial. (For example store f(x) = $4n_3 + 2n + 9$ in an array and for a given value of n, say n = 5, evaluate (i.e. compute the value of f(5)).

12. Write a Program to represent Graphs using the Adjacency Matrices and check if it is a complete graph.

13. Write a Program to accept a directed graph G and compute the in-degree and out-degree of each vertex.

14. Given a graph G, write a Program to find the number of paths of length n between the source and destination entered by the user.

15. Given an adjacency matrix of a graph, write a program to check whether a given set of vertices $\{v_1,v_2,v_3.....,v_k\}$ forms an Euler path / Euler Circuit (for circuit assume $v_k=v_1$).

16. Given a full m-ary tree with i internal vertices, Write a Program to find the number of leaf nodes.

References

1. Mohapatra, & Liu, C. L. (2012). *Elements of Discrete mathematics*. 4th edition. McGraw Hill Education.

2. Rosen, K. H. (2011). *Discrete Mathematics and Its Applications*. 7th edition. Tata McGraw Hill Education.

Additional Resources

1. Albertson, M. O., & Hutchinson, J.P., (1988). *Discrete Mathematics with Algorithms.* John Wiley and Sons.

2. Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (2009). *Introduction to algorithms*. 3rd edition. MIT Press.

3. Hein, J. L. (2015). *Discrete Structures, Logic, and Computability.* 4th edition. Jones and Bartlett Learning.

4. Hunter, D. J. (2011). *Essentials of Discrete Mathematics.* 2nd edition. Jones and Bartlett Learning

Course Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Tentative weekly teaching plan is as follows:

Week	Content
1-3	Sets Finite and infinite sets, uncountable infinite sets; functions, relations, properties of binary relations, closure, partial ordering relations, pigeonhole principle, permutation and combination, induction, inclusion exclusion
4-5	Growth of Functions Asymptotic notations, summation formulas and properties, summation formulas and properties (contd.), bounding summations, approx. by integrals
6-8	Recurrences Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, recursion trees, Master's Theorem
9-13	Graph Theory Basic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and Circuits, planar graphs, graph coloring, Trees, basic terminology and properties of Trees, introduction to spanning trees.
14-15	Prepositional Logic Logical connectives, well-formed formulas, tautologies, equivalences, inference theory

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

recurrence, trees and graphs, combinatorics, inductive and deductive reasoning, asymptotic complexity.

Data Structures (BHCS05) Discipline Specific Core Course - (DSC)