Year I
Semester-1

Course Code: DSC1101	Course Title: Discrete Structures
Course Credits: 04 (Th)	Hour of Teaching/Week: 04

Unit-I

1**. Introduction**
Sets - finite and Infinite sets, uncountably 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.

2. **Growth of Functions**
Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals.

3. **Recurrences**
Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem.

Unit-II

4. **Graph Theory**
Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representaion, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees

5. **Prepositional Logic**
Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory

Suggested Readings:
1. C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition, Tata McGraw Hill, 1985. 2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition, McGraw Hill, 2006.
3. T.H. Coremen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition Prentice Hall on India, 2009.
4. M. 0. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms, John Wiley Publication, 1988.
5. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009.
6. D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

Course Code: DSC1102	Course Title: Computer Fundamentals and Programming in C
Course Credits: 2(Th)+2(P)	Hour of Teaching/Week: 06

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks.
- Understand working of Hardware and Software and the importance of operating systems.
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts.
- Read, understand and trace the execution of programs written in C language.
- Write the C code for a given problem.
- Perform input and output operations using programs in C.
- Write programs that perform operations on arrays.

Course Content:

Unit-I

Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.

Unit-II

Introduction to C Programming: Over View of C; History and Features of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - printf and scanf, control stings and escape sequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions.

C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion.

Control Structures: Decision making Statements - Simple if, if_else, nested if_else, else_if ladder, Switch-case, goto, break & continue statements; Looping Statements - Entry controlled and Exit controlled statements, while, do-while, for loops, Nested loops.

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions -strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc. Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;

User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type.

User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.

Practice Lab

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
   a. Familiarization of Computer Hardware Parts
   b. Basic Computer Operations and Maintenance.
   c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Programming Lab

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series
   S = 1+1/2+1/3+1/4+.....
4. WAP to compute the sum of the first n terms of the following series
   S =1-2+3-4+5.....
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

   *
   **
   * **
   *** *
   *****
   * ****
   *****

   (modify the patterns of the asterisks in different ways)
10. WAP to perform following actions on an array entered by the user:
    a. Print the even-valued elements
    b. Print the odd-valued elements
    c. Calculate and print the sum and average of the elements of array
    d. Print the maximum and minimum element of array
    e. Remove the duplicates from the array
    f. Print the array in reverse order

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays 6 the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions.
16. Write a menu driven program to perform following operations on strings:
    a. Show address of each character in string
    b. Concatenate two strings without using strcat function.
    c. Concatenate two strings using strcat function.
    d. Compare two strings
    e. Calculate length of the string (use pointers)
    f. Convert all lowercase characters to uppercase
    g. Convert all uppercase characters to lowercase
    h. Calculate number of vowels
    i. Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion, (ii) without recursion.
21. Create a matrix. Write a menu-driven program to perform following matrix operations (2-D array implementation):
    a) Sum b) Difference c) Product d) Transpose
22. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
23. Write a program to retrieve the student information from file created in previous question and print it in following format: Roll No. Name Marks
24. Copy the contents of one text file to another file, after removing all whitespaces.
25. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
26. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

(The list of programs above are suggestive and indicative only)

Semester-2

Course Code: DSC1201	Course Title: Data structure
Course Credits: 04 (Th)	Hour of Teaching/Week: 04

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms.

- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs.
- Demonstrate different methods for traversing trees.
- Compare alternative implementations of data structures with respect to performance.
- Describe the concept of recursion, give examples of its use.
- Discuss the computational efficiency of the principal algorithms for sorting and searching.

Course Content:

Unit-I

Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - Fibonacci numbers, GCD, Binomial coefficient nCr, Towers of Hanoi; Comparison between iterative and recursive functions.

Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.

Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and deallocation functions - malloc, calloc, realloc and free. Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header liked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.

Unit-II

Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks, Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.

Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;

Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal; Reconstruction of a binary tree when any two of the traversals are given.

Text Books 1.

Satraj Sahani: Fundamentals of Data Structures

References:

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007))

Course Code: DSC-1202	Course Title: Microprocessor
Course Credits: 2(Th)+2(P)	Hour of Teaching/Week: 06

Course outcome:
On completion of this course the student will be able to:

- Describe the architecture & organization of 8085 & 8086 Microprocessor.
- Understand and classify the instruction set of 8085/8086 microprocessor and distinguish the use of different instructions and apply it in assembly language programming.
- Relate the addressing modes used in the instructions.
- Realize the Interfacing of memory & various I/O devices with 8085/8086 microprocessor.
- Familiarise the architecture and operation of Programmable Interface Devices and realize the programming & interfacing of it with 8085 microprocessor.
- Interface various pheripheral IC's with Intel 8085/8086 microprocessor for its various applications.

Unit-I

Microprocessor architecture: Internal architecture, Programming Model, Addressing modes, Data movement instructions

Microprocessor programming: Register Organization, instruction formats, Program control instructions, assembly language

Unit-II

Interfacing: Bus timings, Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, timer, Interrupt controller, DMA controller, video controllers, communication interfaces. Data transfer schemes: Synchronous data transfer, asynchronous data transfer, interrupt driven data transfer, DMA mode data transfer.

Microprocessor controllers: I/O controllers, interrupt controller, DMA controller, USART controller. Advance microprocessor architecture: CISC architecture, RISC architecture, superscalar architecture, multicore architecture

Practical

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 32-bit binary division and multiplication
2. Write a program for 32-bit BCD addition and subtraction
3. Write a program for Linear search and binary search
4. Write a program to add and subtract two arrays
5. Write a program for binary to ascii conversion
6. Write a program for ascii to binary conversion
7. Write a program in 8085 microprocessor for addition of two 8 bit numbers.
8. Write a program in 8085 microprocessor for addition of two 16 bit numbers.
9. Write a program in 8085 microprocessor for subtraction of two 8 bit numbers.
10. Write a program in 8085 microprocessor for multiplication and division of two 8 bit numbers.

N.B. Students may practice other related programs also.

Reference:

1. An Introduction to Microprocessor 8085, by D.K. Kaushik,
2. Fundamentals of Microprocessors and Microcontrollers by B. Ram.
3. Microprocessor Architecture, Programming and Applications with the 8085 by Ramesh Gaonkar.

TRIPURA UNIVERSITY

(A Central University)
Suryamaninagar-799022

Syllabus

OF

Computer Skills
FNDC-V

Semester – IV

2014

Annexure A

Syllabus

Computer Foundation

(FNDC-V)

Full marks: 100 **Credits: 5**

Introduction: introduction to computer system, Characteristics of Computers, Uses of computers, Types and generations of Computers, Basic Applications of Computer.

Data Representation: Number systems and character representation, decimal, binary, Octal and hexadecimal system, conversion from one number system to another number system, binary arithmetic (addition, subtraction using 1's complement & 2's complement, multiplication).

Representation of data/Information concepts of data processing,

Definition of Information and data, Basic data types, Storage of data/information as files

Human Computer Interface: Types of software, Operating system as user interface, utility programs.

Devices: Input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, bar code reader, web camera, monitor, printer, plotter

Memory: Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks

Use of Computers in Education and Research: Data analysis, Heterogeneous storage, e-Library, Google Scholar, Domain specific packages such as office, SPSS, Mathematica etc.

Reference Books:

1. A. Goel, Computer Fundamentals, Pearson Education, 2010.

2. P. Aksoy, L. DeNardis, Introduction to Information Technology, Cengage Learning, 2006

3. P. K.Sinha, P. Sinha, Fundamentals of Computers, BPB Publishers, 2007