



**3-Year Master of Computer Application (MCA) Curriculum and
Syllabus
First Semester**

Course Code	Course Title	Contact Hrs. / Week			Credit
		L	T	P	
Theory					
TIU-PEN-T101	Career Advancement and Skill Development	1	0	1	2
TIU-PMG-T115	Engineering Economics and Financial Accounting	3	0	0	3
TIU-PMA-T113	Discrete Structures	3	1	0	4
TIU-PCA-T113	Computer Organization	3	1	0	4
TIU-PCA-T103	C programming Concepts & Computational Techniques	3	1	0	4
TIU-PCA-T119	Programming in R Language	3	1	0	4
TIU-PCA-T111	Principles of System Software	2	0	0	2
Practical					
TIU-PCA-L103	C Programming Laboratory	0	0	3	2
TIU-PCA-L117	OS Lab	0	0	3	2
TIU-PCA-L119	R Lab	0	0	3	2
Sessional					
TIU-PES-S199	Entrepreneurship Skill Development	0	0	4	2
Total Credits					31

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)
External Expert-2 (Prof. Amlan Chakraborty, C.U.)
HOD - (Prof. A.B. Chaudhuri)



Detailed Syllabus

Career Advancement and Skill Development

TIU-PEN-T101

L-T-P: 1-0-1

Credit: 2

Topics	Credit
	1
Concept of MOODLE & its handling	
English Language.	
Aptitudes	1
Total	2

Engineering Economics and Financial Accounting

TIU-PMG-T115

L-T-P: 3-0-0

Credit: 3

UNIT I Introduction

Managerial Economics - Relationship with other disciplines - Firms: Types, objectives and goals - Managerial decisions - Decision analysis.

UNIT II Demand & Supply Analysis

Demand - Types of demand - Determinants of demand - Demand function – Demand Elasticity - Demand forecasting - Supply - Determinants of supply - Supply function - Supply elasticity.

UNIT III Production and Cost Analysis

Production function - Returns to scale - Production optimization - Least cost input - Isoquants - Managerial uses of production function. Cost Concepts - Cost function – Types of Cost - Determinants of cost - Short run and Long run cost curves - Cost Output Decision - Estimation of Cost.

UNIT IV Pricing

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



Determinants of Price - Pricing under different objectives and different market structures - Price discrimination - Pricing methods in practice – role of Government in pricing control.

UNIT V Financial Accounting (Elementary Treatment)

Balance sheet and related concepts - Profit & Loss Statement and related concepts - Financial Ratio Analysis - Cash flow analysis - Funds flow analysis – Comparative Financial statements - Analysis & Interpretation of financial statements. Investments - Risks and return evaluation of investment decision - Average rate of return - Payback Period - Net Present Value - Internal rate of return.

Recommended Books:

For Main Reading:

1. McGuigan, Moyer and Harris, 'Managerial Economics; Applications, Strategy and Tactics', Thomson South Western, 10th Edition, 2005.
2. Prasanna Chandra. 'Fundamentals of Financial Management', Tata Mcgraw Hill Publishing Ltd., 4th edition, 2005.

For Supplementary Reading:

1. Samuelson. Paul A and Nordhaus W.D., 'Economics', Tata Mcgraw Hill Publishing Company Limited, New Delhi, 2004.
2. Pares Shah, 'Basic Financial Accounting for Management', Oxford University Press, New Delhi, 2007.
3. Salvatore Dominick, 'Managerial Economics in a global economy'. Thomson South Western, 4th Edition, 2001.

Discrete Structures

TIU-PMA-T113

L-T-P: 3-1-0

Credit: 4

1. Sets, Relations & Functions.

Sets, subsets & operations on sets, finite and infinite sets. Relations & properties of relations, equivalence, compatibility, partial order relation, Poset, lub, glb, maximal & minimal elements of a poset. Functions, inverse functions, composition of functions, recursive functions.

1. Mathematical Logic

Logic operators, Truth tables, Normal Forms, Propositional Calculus, Theory of inference and deduction, Predicate Calculus.

3. Boolean Algebra

Boolean functions, min & max terms, simplification of Boolean function with Karnaugh Map & Quine McClusky method, Lattices.

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



4. Number Theory

Greatest Common Divisors, Euclidean Algorithms, Fibonacci Numbers, Complexity of Euclidean Algorithms, Congruencies and Equivalence Relations, Public Key Encryption Schemes, Dividends.

5. Groups & Subgroups

Group axioms, permutation groups, subgroups, cosets, normal subgroups, semi-groups, free semi-groups, monoids, sequential machines, error correcting codes, modular arithmetic grammars.

6. Combinatorics & Recurrence Relations

Basic Theorems on permutation and combinations. Pigeon hole principle, principle of inclusion and, exclusion. Ordinary & exponential generating functions, recurrence relation, solving recurrence relation by substitution, solving recurrence relation by conversion to linear recurrence relation.

7. Graph theory

Basic definitions trees and graphs, connectivity of a graph, cut points, cycles, Hamiltonian graphs, Trees, different characterization of trees, bipartite graphs, Planar and Dual graphs. Euler theorem. Algorithms on graphs and trees like Breadth first search & Depth first search, Dijkstra's algorithm for shortest path, Floyd's algorithm for all pair shortest paths, Kruskal's and Prim's algorithm for Minimum cost spanning tree. Graph colouring.

8. Finite State Machines and Languages

Languages, Representation of Special languages and Grammars, Finite State Machines, Semigroups, machines and Languages. Machines and Regular Languages, Simplification of Machines.

Recommended Books:

For Main Reading:

1. Kolman, Busby & Ross "Discrete Mathematical Structures"
2. Trembly, J.P. & Manohar. P "Discrete Mathematical Structures with Applications to computer Science"
3. C.L.Liu, "Elements of Discrete Mathematics"

For Supplementary Reading:

1. M.O. Albertson and Joan P. Hutchinson, "Discrete Mathematics with Algorithms"
2. N.Ch. S.N. Iyengar, V.M. Chandrasekaran, K.A. Vanatesh, P.S. Arunachalam, "Discrete mathematics"
3. Peter Linz, "an Introduction to Formal languages and Automata"
4. Narsingh Deo "Graph Theory with Appl. To Engg. & Computer .Sc"
5. M. Sipson & :Lipshutz, "Discrete Mathematics"

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)
External Expert-2 (Prof. Amlan Chakraborty, C.U.)
HOD - (Prof. A.B. Chaudhuri)



Computer Organization

TIU-PCA-T113

L-T-P: 3-1-0

Credit: 4

(1) Digital Computers

1- A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection.

2- Logic gates

3- Adders

4- Flip-Flops (as 1 bit memory device), Encoders, Decoders, Multiplexers, Registers, Shift Registers, Counters, RAM, ROM.

(2) Data Representation & Computer Arithmetic

Number systems, BCD, ASCII & EBCDIC Codes, Two's complement: Addition, subtraction, overflow, Floating point representation.

Addition and Subtraction with Signed Magnitude data, Multiplication Algorithms: Hardware Algorithm and Booth Algorithm, Division Algorithm.

(3) Processing Unit:

Organization of a processor - Registers, ALU and Control unit, Data path in a CPU, Instruction cycle, Organization of a control unit - Operations of a control unit, Hardwired control unit, Micro-programmed control unit. Machine instructions, Operands, addressing modes, Instruction formats, Instruction sets. , Software and Hardware interrupts (only brief introduction), Arithmetic and Instruction Pipelines.

(4) Input-Output Organization

Access of I/O devices, I/O ports, I/O control mechanisms -

Program controlled I/O, Interrupt controlled I/O, and DMA controlled I/O, I/O interfaces- Serial port, Parallel port, PCI bus, SCSI bus, USB bus, Firewall and Infiniband, I/O peripherals - Input devices, Output devices, Secondary storage devices. Instruction level pipelining and Superscalar Processors, Multiple Processor Organizations, Closely and loosely coupled multiprocessors systems, Symmetric Multiprocessors, Clusters, UMA NUMA, Vector Computations, RISC: Instruction execution characteristics, RISC architecture and pipelining. RISC Vs CISC.

(5) Memory Organization

Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM , Internal Organization of a memory chip, Organization of a memory unit, Error correction memories, Interleaved memories, Cache memory unit - Concept of cache memory,

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



Mapping methods, Organization of a cache memory unit, Fetch and write mechanisms, Memory management unit
- Concept of virtual memory, Address translation, Hardware support for memory management.

Recommended Books:

Main Reading:

1. M. M. Mano, Computer System Architecture, Third Edition, 2000, Pearson Education.
2. C. Hamacher, Z. Vranesic and S. Zaky, "Computer Organization", McGraw-Hill, 2002.

Supplementary Reading:

1. J. P. Hayes, Computer Architectures & Organization, Third Edition, 1998, McGraw Hill

C Programming Concepts & Computational Techniques

TIU-PCA-T103

L-T-P: 3-1-0

Credit: 4

(1) Introduction to Programming

The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compilation, Linking and Loading, Steps in Program Development including Testing, Debugging, Documentation.

(2) Algorithms for Problem Solving

Exchanging values of two variables, summation of a set of numbers, Decimal Base to Binary Base conversion, reversing digits of an integer, GCD (Greatest Common Division) and LCM of two numbers; testing whether a number is prime, perfect, Armstrong etc. ; Finding square root of a number, factorial computation, Fibonacci sequence, Evaluation of 'sin x' as sum of a series, Reversing the order of elements of an array, Finding the largest and the smallest number in an array, Removing duplicates from an array, Organizing numbers in ascending order, Printing the elements of upper triangular matrix, multiplication of Two matrices Evaluation of a Polynomial.

(3) Introduction to C Language

Character set, Variables and Identifiers, Built-in Data Types, Variable Definition, Arithmetic operators and Expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple 'C' programs.

(4) Conditional Statements and Loops

Page | 6

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



- 1- Decision making within a program Conditions, Relational Operators, Logical Connectives, if statement, if-else statement.
- 2- Loops: while loop, do while, for loop, Nested loops, Infinite loops, switch statement, Structured Programming.

(5) Arrays

- 1- One dimensional arrays Array manipulation, Searching, Insertion, and Deletion of an element from an array, finding the largest / smallest element in an array.
- 2- Two dimensional arrays Addition/ multiplication of two matrices transpose of a square matrix.
- 3- Null terminated strings as array of characters
- 4- Representation sparse matrices

(6) Functions

- 1- Top-down approach of problem solving
- 2- Modular programming and functions
- 3- Standard Library of C functions
- 4- Prototype of a function Formal parameter list, Return Type, Function call, Block structure.
- 5- Passing arguments to a Function Call by reference, Call by value, Recursive Functions, Arrays as function arguments.

(7) Structures and Unions

Structure variables, Initialization, Structure assignment, Nested structure, Structures and Functions, Structures and arrays: Arrays of structures, Structures containing arrays, Unions.

(8) Pointers

Address operators, Pointers type declaration, Pointer assignment, Pointer initialization, Pointer arithmetic, Functions and pointers, Arrays and Pointers, Pointer arrays.

(9) Self Referential Structures and Linked Lists

Creation of a singly connected linked list, traversing a linked list, Insertion into a linked list, Deletion from a linked list.

(10) File Processing

Concept of Files, File opening in various modes and closing of a file, Reading from a file, writing onto a file.

Recommended Books:

Main Reading:

1. B W Kernighan & D.M. Ritchie, The C Programming Language, Prentice Hall of India, 1989
2. Applications Programming in ANSI C, 3rd Edition, Richard Johnsonbaugh, DePaul University Martin Kalin, DePaul University ©1996 |Pearson

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



3. Programming in ANSI - C, Balagurusamy, E, Tata McGraw-Hill
4. R G Dromey, How to solve it by Computer, Prentice Hall in India, 1992.
5. The Art of Programming through Flowcharts & Algorithms, Anil Bikas Chaudhuri, Firewall Media, 01-Dec-2005.

Supplementary Reading:

1. Jones, Robin & Stewart, The Art of C Programming, Narosa Publishing House, New Delhi.
2. Kenneth A, C Problem solving and Programming, Prentice Hall International.
3. Schildt H, C Made easy, MacGraw Hill Book Company, 1987.

PROGRAMMING IN 'C' LANGUAGE
Model Questions
PART ONE(For MTA & ESA)

1. Each question below gives multiple choices of answers. Choose the most appropriate one.

- 1.1 The programming Language C happens to be
- a) An Assembly Level Language.
 - b) A High Level Language with some Assembly Level Language Features.
 - c) A Programming Language used only to write System Software.
 - d) A Programming Language used for developing Application Packages only.
- 1.2 The C declaration `int I_a;` implies
- a) The variable `I_a` is a signed Binary Integer .
 - b) The variable `I_a` is an Unsigned Decimal Integer.
 - c) The variable `I_a` is an signed Hexadecimal Integer.
 - d) The variable `I_a` is a signed Integer that can be expressed in any Base.
- 1.3 The C statement `printf ("The Value =%x",62);` will print
- a) The Value= 62
 - b) The Value = O62
 - c) The Value= OX 3C
 - d) The Value= 3C
- 1.4 In the following C declaration
- ```
float F_C = 12.5;
void VF_A (int);
int main(); { /* begin main */
float F_B; F_C = 13.5;
.....
return (0); }/* end main */
```

B Level Syllabus R4 58

- a) The Variable `F_C` is GLOBAL to both the functions `main ()` as well as `VF_A`.
- b) The Variable `F_C` is LOCAL to the function `main()`;
- c) The Variable `F_C` is LOCAL to the function `VF_A`.
- d) The Variable `F_C` is EXTERNAL.

1.5 Consider the following C Program .

```
define S 10+2
#include <stdio.h>
int main()
{ /* begin main */
```

**Approved by:**

**External Expert-1 (Prof. Subhadip Basu, J.U.)**  
**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**  
**HOD - (Prof. A.B. Chaudhuri)**



```
int Result = S + S ;
printf (“\n\n Result = %d\n\n”, Result); /* Output Line #2 */
return (0);
} /* end main*/
```

The Output generated by the above C Program will be

- a) Result = 10
- b) Result = 12
- c) Result = 24
- d) Result = 20

1.6 What will be the Output generated by the following C Program ?

```
#include <stdio.h>
int main()
{ /* begin main */
int I_C ; float F_D , F_E;
I_C = 5/2 ; F_D = 5/2 ; F_E = 5/2.0;
printf (“\n I_C = %d F_D = %f F_E = %f\n\n”, I_C,F_D,F_E);
return (0);
} /* end main*/
```

- a) I\_C= 1 F\_D = 2.0 F\_E = 2.5
- b) I\_C= 2 F\_D = 2.0 F\_E = 2.5
- c) I\_C= 2 F\_D = 2.5 F\_E = 2.0
- d) I\_C= 2 F\_D = 2.5 F\_E = 2.5

1.7 In C Functions the actual expressions / parameters are passed on to Formal parameters using the method of :

- a) Call by reference.
- b) Call by Value Result.
- c) Call by Value.
- d) Call by Name.

1.8 Consider the following C program segment :

```
typedef struct Point
{ float F_x;
float F_y;
}Point_T;
typedef struct Circle
{ float F_Radius;
Point_T R_Center;
} Circle_T;
int main();
{ // begin main
Point_T R_Point; Circle_T R_Circle;
/* Circle Manipulation Statements */
return(0);
} // end main
```

To manipulate a circle which of the following set of assignment statements will have to be used ?

- a) R\_Circle.F\_Radius = 10.2; R\_Circle.R\_Center.F\_x = 2.0 ;  
R\_Center.F\_y=3.0;
- b) R\_Circle.F\_Radius = 10; R\_Circle.F\_x = 2.0 ; R\_Circle.F\_y=3.0;
- c) R\_Circle.F\_Radius = 10.2; R\_Circle.R\_Center.F\_x = 2.0 ;

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



- d) R\_Circle.R\_Center.F\_y=3.0;  
e) R\_Circle.F\_Radius = 10.2; R\_Circle.F\_x = 2.0 ; R\_Circle.F\_y=3.0;

1.9 In the following C Declaration

```
#define CUI_Size 10
typedef int AI_1D_01_T [CUI_Size];
int main()
{ /* begin main */
AI_1D_01_T AI_1D_A;
```

The variable AI\_1D\_A represents

- a) An array of Integers of any size.  
b) An array of Integers having minimum 10 integers.  
c) An array of Integers having Maximum 10 Integers.  
d) None of the above.

1.10 Consider the following C Code

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{ /*begin main */
int I_X=6; int *PI_Y;
PI_Y = (int*) malloc (sizeof (int));
*PI_Y = I_X;
printf(" *PI_Y =%d",*PI_Y);
*PI_Y = 7;
printf (" I_X = %d",I_X);
return(0);
} // end main
```

Which, among the following will it produce as output ?

- a) \*PI\_Y = 7 I\_X = 6  
b) \*PI\_Y = 6 I\_X = 7  
c) \*PI\_Y = 7 I\_X = 6  
d) \*PI\_Y = 6 I\_X = 6

**2. Each statement below is either TRUE or FALSE. Identify and mark them accordingly in the answer book**

- 2.1 In C %x format can be used for Inputting signed Octal Integers (FALSE).  
2.2 A Pointer variable content will be the Address of the variable it points to. (TRUE).  
2.3 In C , a SINGLE scanf () can be used to read in the values of any number of predeclared variables (TRUE).  
2.4 Arrays in C are always stored in Column Major fashion (FALSE).  
2.5 ! operator is a BINARY Operator in C. (FALSE).  
2.6 Recursive functions provide an elegant way of representing recurrences (TRUE).  
2.7 Array represents a homogeneous Data Structure (TRUE).  
2.8 A structure cannot be a member of an Union in C (FALSE).  
2.9 In C \*p++ increments the content of the location pointed to by p (TRUE).  
2.10 A C Function can return a whole structure as it's value (TRUE).

**3. Match words and phrases in column X with the nearest in meaning in column Y.**

**X**

- 3.1 Premature exit from within a C Loop  
3.2 Character variable will have a size of

**Y**

- a) 1 Byte  
b) Indentation is essential

Approved by:

**External Expert-1 (Prof. Subhadip Basu, J.U.)**  
**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**  
**HOD - (Prof. A.B. Chaudhuri)**



EM 4, Sector V, Salt Lake, Kolkata-700091, West Bengal, India  
Phone: +91 9836544416/17/18/19, Fax: +91 33 2357 1097

- 3.3 A 'C' Function that do not return a value will be having c) Call by Reference.  
3.4 A string in C is terminated by d) To open a file for writing after discarding it's previous content  
3.5 To understand the Blocks of 'C' e) An Integer type  
3.6 Multiway branching in C can be implemented f) A void type  
3.7 All variables declared inside a function g) Are Local to that function  
3.8 A Pointer Parameter in a C Function simulates h) Opening a file in Read mode, retaining the previous content  
3.9 A Linked List represents i) A white space character.  
3.10 In C fopen "w" mode is used j) 4 Bytes  
k) A '\0' Character  
l) A dynamic Data Structure  
m) Using switch – case statement  
n) Can be achieved by break statement

**6. Fill in the blanks in 4.1 to 4.10 below, by choosing appropriate words and phrases given in the list below:**

(a) Dividing (b) One or ZERO (c) CPU Register (d) extern (e) Optional (f) Randomly (g) At least once (h) At run time (i) Linked List (j) An Array (k) Fields

- 4.1 The Operator  $I\_Value \gg 2$  is equivalent to \_\_\_\_\_ Value by 4 .  
4.2 The Declaration `reg int IReg_C` will allocate a \_\_\_\_\_ for the variable `IReg_C`.  
4.3 On executing `f = ! (K > 10)` f will have a value \_\_\_\_\_  
4.4 The individual Elements of any Array can be accessed \_\_\_\_\_  
4.5 The else portion of an if else statement in c is \_\_\_\_\_  
4.6 In C the body of do-while loop will be executed \_\_\_\_\_  
4.7 Any variable starting with \_\_\_\_\_ in the declaration will be treated as an External variable  
4.8 In C a polynomial of the form  $100M^3 - 20M + 10$  can be efficiently represented by a \_\_\_\_\_  
4.9 The Components of a Records are termed as \_\_\_\_\_  
4.10 In C any dynamic data structure is created \_\_\_\_\_ .

**PART TWO( For ESA)**

5. Consider the following C program Outline that DOES NOT USE any Structured Data Type like ARRAY or STRUCTURE or POINTER whatsoever ANYWHERE

```
:
#include <stdio.h>
#include <math.h>
/* NO OTHER LIBRARY CAN BE USED*/
#define CI_Max 9999
#define CI_Min -9999
/* NO OTHER USER DEFINED CONSTANTS, DATA TYPES OR
GLOBALS CAN BE USED*/
/* User Defined Function Prototypes. NO OTHER FUNCTIONS are used */
void VF_Read_Int (int, int, int*); /* READS and Returns an Integer through
it's pointer parameter provided it lies between a specific range passed as
the other two parameters . If the value read in within the happens to be
OUTSIDE this range, it will continue to loop & print the message Input
```

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)  
External Expert-2 (Prof. Amlan Chakraborty, C.U.)  
HOD - (Prof. A.B. Chaudhuri)



OUT of range ,  
Give Again and wait for a proper value to be inputted by the user. \*/  
int IF\_Test\_Prime (int) ; /\* Used to Test MOST EFFICIENTLY whether the  
Integer passed as it's only parameter happens to be Prime or Not. It  
Returns 1 if the passed Integer is prime returns 0 if it is Non Prime. In  
each case , it prints an appropriate message within it \*/  
int main ()  
{//begin main int I\_Value;  
/\* You May Employ other Simple Variables \*/ VF\_Read\_Int (CI\_Max,  
CI\_Min, &I\_Value);  
/\* Reads in an Integer Value within a Specified Range \*/  
VF\_Print\_NON\_Prime\_Factors (I\_Value);  
/\* Displays all the NON Prime Factors of the value I\_Value\*/  
return(0);  
}//end main

a. Frame the body of the function

VF\_Read\_Int

. The Function heading is as illustrated below :

```
void VF_Read_Int (int I_High, int I_Low, int *PI_X)
/* READS and Returns an Integer through it's pointer parameter
provided it lies between a specific range passed as the other two
parameters . If the value read in within the happens to be
OUTSIDE this range, it will continue to loop print the message {\bf
Input OUT of range , Give Again and wait for a proper value to
be inputted by the user. NO OTHER PARAMETER CAN BE USED.
*/
```

b. Frame the body of the function

IF\_Test\_Prime

. The Function heading is as illustrated below :

```
int IF_Test_Prime (int I_Num) /* Used to Test MOST EFFICIENTLY
whether the Integer passed as it's only parameter happens to be
prime or Not. It Returns 1 if the passed Integer is prime returns 0 if
it is Non Prime. In each case , it prints an appropriate message
within it */
```

**(6+9)**

6. Consider two integer data files F1 and F2 having following features.

- Number of data values (key) in each file is unknown and the files may be of different sizes.
- The values / Integer Keys in both the files F1 & F2 are Sorted in Descending Order.
- Same data ( key) can appear more than once in F1 or F2.
- F1 and F2 may share common data values i.e. same key item may appear both the files .

Write a C function to merge the two files F1 and F2 to form a third file F3 having the following features.

-Elements in F3 are sorted in ascending order.

- Duplicate entries are not permitted (i. e. ,No element appears more than once).

**(15)**

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



EM 4, Sector V, Salt Lake, Kolkata-700091, West Bengal, India  
Phone: +91 9836544416/17/18/19, Fax: +91 33 2357 1097

7. The following operations are defined on a sorted Doubly linked list of Integers L where elements are arranged in Descending order from left.

INSERT (L,X) : Insert the integer X in the list L if X is not present.

DELETE (L,X) : Delete the integer X from the list L (if it exists).

SHOW-MID (L) : Print the  $n/2$  th element of the list from left where n is the Number of elements in the current list and we use integer

Division where  $5/2 = 2$

Frame C functions to implement each of the above functions INSERT (L,X) , DELETE (L,X) and SHOW\_MID( L)

**(6+6+3)**

**8.**

a. In 2 (two) dimensions, a point can be described by its two coordinates namely X & Y both of which can be real numbers. A line can be described in the following manner :

(i) The co-ordinates of its two end points (X1, Y1) & ( X2, Y2 )

(ii) Its gradient 'm' & intersection 'c' (in the form  $Y = mx + c$ )

(iii) The length of the line is also stored along with.

Specify appropriate data types to store a point as well as a line in C.

**(1+2)**

b. Write a C function Point\_to\_Line (P1, P2) that will accept as parameters the coordinates of two points P1 & P2 and return a line that has the aforesaid 2 points as its end points.

**(5)**

c. A quadrilateral can be described by a sequence of 4(four) lines such that one end point of one line happens to be the starting point of the next line.

Specify a suitable data structure in C to represent a quadrilateral.

**(2)**

d. Write a C function that will accept a quadrilateral as a parameter and classify it whether it is a

[ $2+2+3=7$ ]

A Square.

A Rhombus.

A Rectangle.

in each case it computes the perimeter as well.

**(2+2+3)**

**9.**

a. Write a single Recursive C function to generate the n th Fibonacci number Fib(n) ( n being a +ve non zero integer ) . You cannot use any array, global variables and/or additional parameters/functions. Trace out the Call & Return sequences along with return values clearly by a schematic diagram when your function Fib(n) is invoked from main() with  $n = 6$ . Also mention the TOTAL no. of times any Fib(n) is called for each value of n for invoking Fib(6) from main(), e.g. Fib(2) is called a total of 4 times etc.

**(2+5+2)**

b. What will be the value of A(1, 3) if A(m, n) happens to be defined in the following manner? Specify each computation step in detail .

$A(0, n) = n + 1$  for  $n \geq 0$

$A(m, 0) = A(m - 1, 1)$  for  $m > 0$

$A(m, n) = A(m - 1, A(m, n - 1))$  for  $m, n > 0$  **(6)**

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



**ASSIGNMENTS in 'C' Language**

**Assignment 1.**

Write a program to find sum of all prime numbers between 100 and 500.

**Assignment 2.**

Write a program to obtain sum of the first 10 terms of the following series for any positive integer value of X :

$X + X^3/3! + X^5/5! + X^7/7! + \dots$

**Assignment 3.**

Write a program to reverse the digits of a given number. For example, the number 9876 should be returned as 6789.

**Assignment 4.**

Write a program to compute the wages of a daily laborer as per the following rules :-

Hours Worked Rate Applicable

Upto first 8 hrs Rs 50/-

For next 4 hrs Rs 10/- per hr extra

For next 4 hrs Rs 20/- per hr extra

For next 4 hrs Rs 25/- per hr extra

For rest Rs 40/- per hr extra

Accept the name of the laborer and no. of hours worked. Calculate and display the wages. The program should run for N number of laborers as specified by the user.

**Assignment 5.**

Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number. Print the number and its frequency in a tabular form.

**Assignment 6.**

Define 2 dimensional array a (3,3), b(3,3),sum(3,3),diff(3,3),mult(3,3). Store 9 arbitrary numbers in a(3,3) and 9 arbitrary numbers in b(3,3). Do the following:

a) Calculate sum of a(3,3) and b(3,3) and store in sum(3,3) where

$sum(i,j)=a(i,j)+b(i,j)$

b) Calculate difference of a(3,3) and b(3,3) and store in diff(3,3) where

$diff(i,j)=a(i,j)-b(i,j)$

c) Calculate product of two arrays a(3,3) and b(3,3) and store in mult(3,3) where

$mult(i,j)=\text{summation of } a(i,k)*b(k,j) \text{ over } k \text{ where } k=1 \text{ to } 3.$

Print the result in a tabular form

**Assignment 7.**

Write a function, str\_search(char\* s1,char\* s2, int n) , that takes two strings and an integer, as arguments and returns a pointer to the nth occurrence of 1st string s1 in 2nd string s2, or NULL if it is not present.

**Assignment 8.**

Write a C function to remove duplicates from an ordered array. For example, if input array contains 10,10,10,30,40,40,50,80,80,100 then output should be 10,30,40,50,80,100.

**Assignment 9.**

Apply recursive call to do the following:

(i) Input 'n'(1-200). Calculate sum of 'n' numbers.

(ii) Input 'n'(1-20). Calculate product of 'n' numbers.

(iii) Input 'n'(2-20). Print 'n' number of Fibonacci numbers. In Fibonacci sequence the sum of two successive terms gives the third term. The following are few terms of

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



Fibonacci sequence :-  
1 1 2 3 5 8 13 .....

**Assignment 10.**

Write a program which will arrange the positive and negative numbers in a onedimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers.

Example:

Original array contains: 10,-15,1,3,-2,0,-2,-3,2,-9

Modified array: 10,1,3,0,2,-15,-2,-2,-3,-9

**Assignment 11.**

Write a menu driven program to maintain a Telephone Directory having following file structure:

1. Name : Character type : Length =20 characters.
2. Address : Character type : Length =40 characters.
3. Phone: Character type : Length =12 characters.

Menu

1. Add record(s)
2. Display record(s)
3. Search record(s)
4. Modify record(s)
5. Delete record(s)
6. Backup copy of File
7. Exit

Type your choice= 1,2,3,4,5,6,7— ->

**Assignment 12.**

Write a program to extract words form any text file and store in another file. Sort the words in alphabetical order and store them in the same file. Read the sorted file and print the frequency of each word.

**Assignment 13.**

Write a program to remove all occurrences of word “the” and “The” from an input string. For example

Input : The Dhillon Theatre is now the Fun Republic.

Output : Dhillon atre is now Fun Republic.

**Assignment 14.**

Write a program to display the Following pattern called Floyed’s Triangle.

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

**Assignment 15.**

Write a program that accepts an input integer ‘n’ in the range 3-9 inclusive, and display the following pattern on a cleared screen.

Sample input for n=3 Sample input for n=4

Sample output Sample output

```
3 4
3 2 3 4 3 4
3 2 1 2 3 4 3 2 3 4
3 2 3 4 3 2 1 2 3 4
```

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



3 4 3 2 3 4

4 3 4

4

**Assignment 16.**

Write a program to count the vowels in free text given as standard input. Read text one character at a time until you encounter end-of-data. Then print out the number of occurrences of each of these vowels.

**Assignment 17.**

Write a program to copy one file to another such that every word is reversed before being written to the target file. Assume the maximum size of each word is 10 characters and each word is separated either by new line(s), tab(s) or space(s). For example, if source file contains "I am an Indian", the target file should contain "I ma na naidnI".

**Assignment 18.**

Define a structure for an Employee having EmployeeName, EmployeeCode, BasicPay, DearnessAllowance, HRA, PF, GrossPay, NetPay Take an array of 10 Employees. Write 'C' functions to :-

- Accept data for EmployeeName, EmployeeCode, BasicPay for all the employees.
- Compute :-
  - DearnessAllowance = 50% of BasicPay
  - HRA = 20% of BasicPay + DearnessAllowance
  - PF = 12% of BasicPay + DearnessAllowance
  - GrossPay = BasicPay + DearnessAllowance + HRA
  - NetPay = GrossPay – PF
- Display the name of employee who has highest GrossPay.
- Compute and display average net pay.
- Display list of all employees in the alphabetical order of employee name.

**Assignment 19.**

Write a program to convert a given decimal number to its binary equivalent and vice versa.

**Assignment 20.**

Input any positive integer number ( $n \leq 9999999$ ). Convert the number into words.

**Assignment 21.**

- Define a structure of a node of a linked list having an integer data member x.
- Use the above structure in (a) and write the functions for the following parts;
  - a function which takes a pointer to the head of linked list, which is in ascending order and an integer, x to be inserted in the linked list, as arguments. The node must be inserted in such a way that the linked list remains in ascending order after insertion.
  - a function which takes a pointer to the head of a linked list and an integer, x to be removed from the linked list, as arguments. If x is not found in the linked list, then it should display an appropriate message.

**Assignment 22.**

Write a program to replace 'a' with 'b', 'b' with 'c', ..., 'z' with 'a' and similarly for 'A' with 'B', 'B' with 'C', ..., 'Z' with 'A' in a file. The other characters should remain unchanged.

**Assignment 23.**

Write a function `char* stuff(char* s1, char* s2, int sp, int rp)` to stuff string s2 in string s1 at position sp, replacing rp number of characters (rp may be zero).

**Assignment 24.**

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



EM 4, Sector V, Salt Lake, Kolkata-700091, West Bengal, India  
Phone: +91 9836544416/17/18/19, Fax: +91 33 2357 1097

Write a program to display the content of a Text file which means it will behave like TYPE command of MSDOS. Suppose the name of your program file: FILETYPE.C and FILETYPE.EXE and the name of the source file is MYFILE.TXT. The following command should work: C: \PROGRAM> FILETYPE MYFILE.TXT

**Assignment 25.**

Write a program to input name, address and telephone number of 'n' persons (n<=20). Sort according to the name as a primary key and address as the secondary key. Print the sorted telephone directory.

**Introduction to OS**  
**TIU-PCA-T109**

**L-T-P: 3-1-0**

**Credit: 4**

**Introduction to OS:**

Definition. Services offered. OS as a Resource Manager of a computer.

**Disk Operating Systems:**

Definition. DOS Commands, Simple File Operations, Directory Related Commands. Bath files.

**Microsoft Windows:**

An overview of different versions of Widows, Basic Windows elements, File management through Windows, Using essential accessories: Systems tools - Disk cleanup, Disk defragmenter, Entertainment, Games, Calculator, Imaging - Fax, Notepad, Paint, WordPad.

**UNIX:**

**Basics of UNIX:**

**UNIX Architecture:** Multi-user, Multitasking, Kernel and Shell, The file system, Relation between process and file, The building block approach, The command line, Command syntax, Internal and external commands, The two schools (AT & T and Berkeley), The graphical user interface (GUI), Seeking help (man), Open source software and Linux. Types of files in Unix, Structure of the system, File System type: Parent-child relationship, Directory handling and navigation (mkdir, rmdir, pwd and cd), The PATH variable, Absolute and relative pathnames, and Directories, Creating, viewing (cat), copying (cp), renaming (mv) and deleting (rm) files, Listing files (ls), viewing through pg, tail and head command.

**File Attributes Structure** of the inode, Brief discussion on partitions and file systems, Analyzing the ls -l output, File type and permissions (chmod), Significance of directory permissions, Hard and soft Links, Concept of ownership, The /etc/passwd and /etc/group files, Changing ownership (chown and chgrp), Modification and access times, Default file and directory permissions (umask). The vi Editor. The three modes, Basic navigation (h, j, k, l), Moving to a specific line number (G), The repeat factor, The input mode commands (i, a, r, s and o), Saving and quitting (:w, :x and :q), Text deletion (x and X), Using operators in deleting and copying text (d, y and p), Undoing and repeating commands (u and .), Pattern search (/ and n) and substitution (:s), Moving text from one file to the

**Approved by:**

**External Expert-1 (Prof. Subhadip Basu, J.U.)**

**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**

**HOD - (Prof. A.B. Chaudhuri)**



other, Customization features: Abbreviation (:abb), Key mapping (:map), Setting vi parameters (:set), The file .exrc.

**The Shell as an Interpreter.** The major shells: bourne Shell, C shell, Korn and Bash, The shell's interpretive cycle, Wild-cards, Escaping and quoting, Difference between single and double quotes, The three standard files and redirection (>, < and >>), Connecting commands with pipes (|), Command substitution, Shell variables and how they determine system behaviour, Aliases and Command history, Environment variables, setting terminal (Sty).

### **Process Management:**

Processing basics, Process States, Process State Transition, Process Control Block (PCB), Parent-child relationship, The different segments of a process, Internal and external commands, Process creation basics (fork() and wait()), Role of init in process creation and in spawning user shells, Exporting variables (export) and consequences, Exit status of a process (%?), Displaying process attributes (ps), Killing processes (kill), Running commands in background (& and nohup), Job control (fg, bg and [ctrl-z]), Scheduling processes (cron).

### **Filters and Regular Expressions:**

Using wc, head, tail and cut commands, Translating characters (tr), Specifying file and searching for a pattern (grep), Performing substitution (sed), Enhancing power of grep and sed with regular expressions. The awk Filter.

### **Shell Programming:**

Shell scripts, and execution methods, User's initialization file (. Profile and rc, etc.), The dot command, Interactive execution and command line arguments (\$1, \$2 etc), Meta Characters - syntactic (&&, (), & ||, ;;<, > etc), pattern matching, substitute shell variables, Quoting , Test Command. Control flow: For, If, While, Case, The Here document, String handling and computation using expr, Setting positional parameters (set command), and shift, Shell functions, arrays.

### **C Programming Tools:**

Compiling a C program (cc), Multi module programs, Header files, System calls and Library functions, The I/O functions.

### **Recommended Books:**

#### **Main Reading:**

- 1- Maurice J. Bach, Design of the Unix Operating System, Third Edition, 2000, PHI.
- 2- Sumitabha Das, Unix : Concepts and Applications, Third Edition, 1998, Tata McGraw Hill.

#### **Supplementary Reading:**

1. A User guide to unix system, Thomas Rebecca yate, Second Edition, 2002,. Tata McGrw Hill.
- 2.Sptehen Prata Advanced Unix - A programmer's Guide.

#### **Approved by:**

**External Expert-1 (Prof. Subhadip Basu, J.U.)**

**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**

**HOD - (Prof. A.B. Chaudhuri)**



## **Principles of System Software**

### **TIU-PCA-111**

**L-T-P: 2-0-0**

**Credit: 2**

#### **1. Overview of System Software**

Introduction, Software, Software Hierarchy, Systems Programming, Machine Structure, Interfaces, Address Space, Computer Languages, Tools, Life Cycle of a Source Program, Different Views on the Meaning of a Program, System Software Development, Recent Trends in Software Development, Levels of System Software

#### **2. Overview of Language Processors**

Programming Languages and Language Processors, Language Processing Activities, Program Execution, Fundamental of Language Processing, Symbol Tables Data Structures for Language Processing: Search Data structures, Allocation Data Structures.

#### **3. Assemblers**

Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers, Single pass Assembler for Intel x86 Algorithm of Single Pass Assembler, Multi-Pass Assemblers, Advanced Assembly Process, Variants of Assemblers Design of two pass assembler.

#### **4. Macro and Macro Processors**

Introduction, Macro Definition and Call, Macro Expansion, Nested Macro calls, Advanced Macro Facilities, Design Of a Macro Pre-processor, design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor, Design Issues of Macro Processors, Features, Macro Processor Design Options, Two-Pass Macro Processors, One-Pass Macro Processors

#### **5. Linkers and Loaders**

Introduction, Relocation of Linking Concept, Design of a Linker, Self-Relocating Programs, Linking in MSDOS, Linking of Overlay Structured Programs, Dynamic Linking, Loaders, Different Loading Schemes, Sequential and Direct Loaders, Compile-and-Go Loaders, General Loader Schemes, Absolute Loaders, Relocating Loaders, Practical Relocating Loaders, Linking Loaders, Relocating Linking Loaders, Linkers v/s Loaders

**Approved by:**

**External Expert-1 (Prof. Subhadip Basu, J.U.)**

**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**

**HOD - (Prof. A.B. Chaudhuri)**



## 6. Scanning and Parsing

Programming Language Grammars, Classification of Grammar, Ambiguity in Grammatical Specification, Scanning, Parsing, Top Down Parsing, Bottom up Parsing, Language Processor Development Tools, LEX, YACC.

## 7. Compilers

Causes of Large Semantic Gap, Binding and Binding Times, Data Structure used in Compiling, Scope Rules, Memory Allocation, Compilation of Expression, Compilation of Control Structure, Code Optimization

## 8. Interpreters & Debuggers

Benefits of Interpretation, Overview of Interpretation, The Java Language Environment, Java Virtual Machine, Types of Errors, Debugging Procedures, Classification of Debuggers, Dynamic/Interactive Debugger

### Text Books.

For Main Reading:

1. System Software: An Introduction to Systems Programming, 3e Paperback – 2002 by BECK

Books for Supplementary Reading:

1. System Programming , D M Dhamdhare McGraw Hill Publication ,
2. System Programming, Srimanta Pal , OXFORD Publication
3. System Programming and Compiler Construction, R.K. Maurya & A. Godbole
4. System Software, Santanu Chattopadhyay , Prentice - Hall India,2007

## Programming in R Language

### TIU-PCA-T119

L-T-P: 3-1-0

Credit: 4

**Unit-1:** Introduction to R, R character set, R words, constants, operators, precedence and associativity of the operators, R working environment as a displayer, R as a calculator, R as a data manipulator, R objects and their data types.

**Unit 2:** R as a programming environment, programming in R using, Sequence, Selection iteration and Case logic structures.

**Unit 3:** User-defined functions in R, Recursion, Basic data structures in R(vector, factor, list, data frame, matrix, array).

**Unit 4:** Data file handling in R.

Page | 20

Approved by:

External Expert-1 (Prof. Subhadip Basu, J.U.)

External Expert-2 (Prof. Amlan Chakraborty, C.U.)

HOD - (Prof. A.B. Chaudhuri)



**Unit 5:** Charts and Graphs in R

**Unit 6:** Statistical Applications of R.

Recommended Books:

1. The art of Programming through R by Anil Bikash Chowdhury
2. The art of R programming by Norman Matloff, , No Starch Press, Sanfrancisco.
3. Statistical Programming in R by Srinivasa,Siddesh,Shetty and Sowmya, Oxford Higher Education.

### **C PROGRAMMING LAB**

**TIU-PCA-L103**

**L-T-P: 0-0-3**

**Credit: 2**

1. Familiarization of a computer and the environment and execution of sample programs
2. Expression evaluation
3. Conditionals and branching
4. Iteration
5. Functions
6. Recursion
7. Arrays
8. Structures and Unions
9. File handling

### **Operating System Laboratory**

**TIU-PCA-L109**

**L-T-P: 0-0-3**

**Credit: 2**

In this lab students are exposed to DOS, WINDOWS and Linux Operating System. Initially they will be learning some of the frequently used commands. Then they will learn how to write and execute batch files and shell scripts.

**Approved by:**

**External Expert-1 (Prof. Subhadip Basu, J.U.)**

**External Expert-2 (Prof. Amlan Chakraborty, C.U.)**

**HOD - (Prof. A.B. Chaudhuri)**