

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Subject Name: **PRINCIPLES OF MANAGEMENT**  
Year: **1st Year**

Subject Code: **BBA-101**  
Semester: **1st**

<b>Module Number</b>	<b>Topics</b>	<b>Number of Lectures</b>
1	<b>Chapter 1: Introduction to Management</b>	<b>8L</b>
	1.The evolution of Management , Definition and importance of Management, Different schools of Management thought-classical school, Management Sciences School, Behavioural School, Human Relation School, Operational approach, system approach and contingency approach to Management.	4L
	2. Management Process: An overview of planning, Organizing, Staffing, Directing and controlling.	4L
2	<b>Chapter 2: Planning and Organizing</b>	<b>6L</b>
	1. Meaning, objective, nature and importance of planning, planning process, planning premises, types of plan.	3L
	2. Meaning and importance, organization structure / chart, responsibility and authority, span of control, delegation of authority, centralization-decentralization, line and staff relationships	3L
	3. Types of organizations, formal and informal groups in organizations, Matrix organization.	3L
3	<b>Chapter 3: Staffing and Directing</b>	<b>12L</b>
	1.Staffing function, manpower needs, manpower position, recruitment & selection, training & development, performance appraisal, human resource audit	4L
	2. Meaning of direction, nature of directing; Motivation; basis of motivation- fear, money, satisfaction; importance of behaviour; individual needs, needs of organization.	3L
	3. Factors influencing performance; work environment, group dynamics- formal & informal, Communication principles and practices.	3L
	4. Leadership-nature, styles, attitudes, leader behaviour, leader effectiveness, leadership models.	2L
4	<b>Chapter 4: Social Responsibility of Management</b>	<b>3L</b>
	1.Professional Management as compared to traditional system of owner Management, Impact of political system, government Policy, national economic planning on managerial policy.	3L
5	<b>Chapter 5: Ethics Management</b>	<b>3L</b>
	Top Management, Codes of Ethics, Ethics Committees, Ethics Hotlines. Ethics and Law. Corporate Culture and Ethical Climate.	3L

	Improving Ethical Decision Making- Difficulties in Decision-Making, Suggestions for making Ethical Decisions.	
<b>Total Number Of Hours = 32L</b>		

**Assignment:**

**Module-1.**

1. Answer the following questions:

- (a) What do understand by management?
- (b) Write about different schools of management.

- 2. (a) What do you understand by management process?
- (b) What is controlling?

**Module-2.**

- 1. (i) What is planning?
- (ii) What is organizing?
- (iii) Write about types of plan.
- 2. (a) What is span of control?
- (b) Explain matrix organization.

**Module-3.**

- 1. (a) What is staffing?
- (b) What is directing?
- (c) Describe the differences between recruitment & selection.
- 2. State the different leadership styles.

**Module-4.**

- 1. (a) Write about impact of political system in management.
- (b) Write about national economic planning on managerial policy.

**Module-5.**

- 1. (a) What is corporate culture?
- (b) What is ethical climate?

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## Lecture-wise Plan

Subject Name: English Language Communication  
Year: 1<sup>st</sup>Year

Subject Code-BBH102  
Semester: First

Module Number	Topics	Number of Lectures
1	<b>ENGLISH LANGUAGE GRAMMAR</b>	<b>10L</b>
	1. Correction of Errors in Sentences Building Vocabulary Word formation Single Structures and Transformation	3
	2. Word for a group of Words Fill in the blanks using correct Words Sentence	3
	3. Active &Passive Voice Direct &Indirect Narration (MCQ Practice during classes).	4
2	<b>READING COMPREHENSION</b>	<b>3L</b>
	1. Strategies for Reading Comprehension Practicing	1
	2. Technical & Non Technical Texts for Global/Local/Inferential/Referential comprehension; Précis Writing	2
3.	<b>TECHNICAL COMMUNICATION</b>	<b>8L</b>
	1. The Theory of Communication– Definition & Scope Barriers of Communication	4
	2. Different Communication Models EffectiveCommunication (Verbal/Nonverbal) Presentation/Public Speaking Skills (MCQ Practice during classes)	4
4	<b>MASTERING TECHNICAL COMMUNICATION</b>	<b>6L</b>
	1. Technical Report (formal drafting) Business Letter (formal drafting)	3
	2. Job Application (formal drafting) Organizational Communication (seepage3)	3
5	<b>GROUP DISCUSSION</b>	<b>4L</b>
	1. Principle & Practice	4
<b>Total Number Of Hours = 31</b>		

Faculty In-Charge

HOD, HU Dept.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **Assignment:**

#### **Unit-1(ENGLISH LANGUAGE GRAMMAR):**

1. Exercise of Active & Passive Voice Direct & Indirect Narration (MCQ Practice during classes).

#### **Unit-2 (READING COMPREHENSION):**

1. Précis Writing.

#### **Unit-4(TECHNICAL COMMUNICATION):**

1. Job Application practice

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Subject Name: **Business Accounting**

Subject Code: **BBA-103**

Year: **1<sup>st</sup> Year**

Semester: **1st**

<b>Module Number</b>	<b>Topics</b>	<b>Number of Lectures</b>
1 (Financial Accounting)	<b>I. Introduction to Financial Accounting</b>	<b>15L</b>
	1. Concepts – Conventions – Importance and scope.	5L
	2. Accounting Principles- Double entry system – Brief overview of accounting cycle	5L
	3. Introduction to Balance Sheet and Income Statement	5L
	<b>II. Depreciation</b>	<b>2L</b>
	1. Methods – Accounting – Importance	2L
	<b>III. Revenue Recognition</b>	<b>3L</b>
	1. Introduction -Definitions -Sale of Goods -Rendering of Services -The Use by Others of Enterprise Resources Yielding Interest-Royalties and Dividends	3L
	<b>IV. Provisions</b>	<b>2L</b>
	1. Doubtful Debt – Bad Debt – Importance – Provisions – Reservations.	1L
	2. Accounting Treatments	1L
	<b>V. Financial Statements</b>	<b>4L</b>
	1. Introduction to Corporate Final Accounts as per Schedule VI of Companies Act 1956Income Statement.	3L
2. Interpretation of Annual Reports.	1L	
2 (Cost accounting)	<b>I. Cost Accounting</b>	<b>6L</b>
	1. Key terms, cost concepts, classifications, total cost components.	2L
	2. Cost accounting and management accounting.	2L
	3. Cost accounting and financial accounting.	2L
	<b>II. Elements of Cost</b>	<b>6L</b>
	1. Materials (Purchasing, Storekeeping, Issue, Pricing & Control)	2L

	2. Labour (Costing & Control)	2L
	3. Overheads (Analysis, Distribution and Control, Treatment of Special Items).	2L
<b>Total Number Of Hours = 36L</b>		

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Subject Name: Business Mathematics  
Year: 1<sup>st</sup> Year

Subject Code- BBM104  
Semester: First

<b>Module Number</b>	<b>Topics</b>	<b>Number of Lectures (27)</b>
	Definition, Type of matrices	1
	Addition and Subtraction of matrices	1
	Multiplication of the Matrices	1
	Determinantsofa squarematrix, minor and co-factors	1
	Cramer's rule	1
	Inverse of the Matrices	1
	Rank of the Matrices.	1
	Inequalities	1
	Ratio and Proportion	1
	Indices	1
	Surds	1
	Logarithms	1
	Complex Number	1
	Arithmetic progression, Geometric Progression	3
	ElementaryIdeasofConvergenceand divergence	1
	Equations -Simpleand Quadratic Equations	3
	Permutations&Combinations	2
	TheBinomialTheorem	2
	Thestraightline	1
	CompoundInterestandAnnuities	2

Faculty In-Charge

HOD, Maths Dept.

### **Assignment:**

#### **Matrices:**

1. Find the x, y, z and w given that

$$3 \begin{bmatrix} x & y \\ z & w \end{bmatrix} = \begin{bmatrix} x & 6 \\ -1 & 2w \end{bmatrix} + \begin{bmatrix} 4 & x+y \\ z+w & 3 \end{bmatrix}$$

2. Compute the determinant of the matrix  $A = \begin{bmatrix} 1 & 6 & -1 & -2 \\ -2 & 3 & 2 & -2 \\ 3 & 5 & -1 & 2 \\ 1 & -6 & 4 & 7 \end{bmatrix}$  using Laplace expansion.

3. Find the Rank of the Matrix  $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$

4. Find minor, Co-factor and inverse of the Matrix  $A = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 3 & -3 \\ -2 & -4 & -4 \end{bmatrix}$

5. Find its solutions of equation using Cramer's rule  $x + y + z = 6, x - y + 2z = 7, 3x + y + z = 10$ .

6. Find the x, y, z and w given that  $\begin{bmatrix} x & y \\ z & w \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ -1 & 2 \end{bmatrix} * \begin{bmatrix} 4 & 3 \\ 1 & 3 \end{bmatrix}$

### Inequalities:

1. Solve the inequalities  $2x - 17 < 7$ .
2. Solve the inequities  $\frac{x+14}{3} \leq \frac{x+1}{2}$ .
3. Solve the inequities  $x - 8 \geq 4(x+2)$ .
4. Solve the inequities  $\frac{x+9}{2} \leq 10$ .
5. Solve the inequities  $\frac{x-7}{3} \leq x - 12$ .

### Ratio and Proportion:

1. Find the value of x;  $2 : 5 :: x : 20$ .
2. Write this as ratio a centimetre to a meter
3. Two no are in ratio 3:4. If the sum of numbers is 63. Find the numbers.
4. If  $x:y = 1:2$ , then find the ratio of  $(2x+3y) : (x+4y)$ .
5. what must be added to each term of the ratio 2:3, so that it may become equal to 4:5.

### Surds:

1. find the value of x,  $(17)^{3/2} * (17)^x = (17)^5$ .
2. simplify (i)  $[(x)^y]^z$ , (ii)  $(xz)^y$ , (iii)  $(x)^y(x)^z$ , (iv)  $(x^2)^4$ .



# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Subject Name: Business Mathematics  
Year: 1<sup>st</sup> Year

Subject Code- BBM104  
Semester: First

### **A.P. and G.P.**

1. Find the 7<sup>th</sup> term of the GP 1, 5, 25, 125.....
2. Find the 8<sup>th</sup> term of the AP 5, 9, 13, 17....
3. Find the sum of 7<sup>th</sup> term of an AP 15, 22, 29, .....
4. Find the sum of 8<sup>th</sup> term of the GP 4, 8, 16, .....

### **Equations -Simple and Quadratic Equations:**

1. A and B together have Rs. 1210. If  $\frac{4}{15}$  of A's amount is equal to  $\frac{2}{5}$  of B's amount, how much amount does B have?
2. The square of a number is decreased by 15. This value is twice the original number. Find the number(s).
3. 2 times the sum of the number x and 7 is 13.
4. Two small pots and one large pot can hold 8 cups of water. One large pot holds two cups extra cups of water than one small pot. How many cups of water can each pitcher hold?

### **The straight line:**

1. Draw the straight line  $2x + y = 6$ .
2. Find the equation of straight line which passes through (-3, -5) and (1, 9).
3. Find the slope of the equation of straight line which passes through (-6, -7) and (3, 4).

### **Compound Interest and Annuities:**

1. Find simple interest on Rs. 10,000 at the rate of 5% for 5 years. Also find the amount for 5 years.
2. Find the compound amount and compound interest on the principal 20,000 borrowed at 6% compounded annually for 3 years.
3. What is the difference between the compound interests on Rs. 5000 for 2 years at 4% per annum compounded yearly and half-yearly?

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

**Subject Name: Introduction to Computing**  
**Year: 1<sup>st</sup> Year**

**Subject Code: BBC105**  
**Semester: 1<sup>st</sup>**

<b>Module Number</b>	<b>Topics</b>	<b>Number of Lectures</b>
1	<b>Introduction to Computers</b>	<b>2</b>
	1. Generations, Classifications, Applications, Basic Organization, Input and output devices	1
	2. Basic concept of Computer memory, Computer software and networks	1
2	<b>Number system</b>	<b>4</b>
	1. Decimal, Binary, Octal, Hexa-decimal, Conversion of numbers, Addition and subtraction of two numbers, Two's compliment	2
	2. Multiplication and division of binary numbers, Working with fractions, signed number representation in binary form	1
	3. Logic gates	1
3	<b>Introduction to C</b>	<b>4</b>
	1. compiling and executing C programs, using comments, keywords, identifiers, Data type, variables, constants	1
	2. input/output statements in C, operators in C	2
	3. type conversion and type casting.	1
4	<b>Decision Control and looping statements</b>	<b>6</b>
	1. conditional branching statement	2
	2. iterative statements	2
	3. nested loops, break and continue statements, goto statement	2
5	<b>Arrays &amp; Strings</b>	<b>6</b>
	1. Declaration, accessing elements of array, storing values	1
	2. calculating the length of array, two dimensional arrays	1
	3. reading and writing strings, suppressing input, string taxonomy	1
	4. string operations – using and without using library function	2
	5. array of strings	1
6	<b>Functions</b>	<b>5</b>
	1. Declaration, prototype, definition, function call	1
	2. return statement, passing parameters to the function	1
	3. scope of variable, storage classes	1
	4. recursive functions	2
7	<b>Pointers</b>	<b>7</b>
	1. introduction, declaration, Pointer expression and arithmetic	1
	2. null pointer, generic pointer, passing arguments to functions using pointer	1
	3. pointers and arrays, passing an array to function, difference between array name and pointer	2
	4. pointers and strings, array of pointers	1
	5. function pointers, pointers to pointers	1
	6. dynamic memory allocation, drawbacks	1

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

8	<b>Structure-union, Files, Preprocessor directives</b>	<b>4</b>
	1. Structure, nested structure, array of structure	1
	2. union, array of union variable, unions inside structure	1
	3. Files – Reading –writing etc	1
	4. Preprocessor directives	1
Total Lecture Hours – 37 l.h.		

Faculty In-Charge

HOD, CSE Dept

### **Assignments :**

#### **Unit 1:- (Introduction to Computers)**

1. How is OCR technology better than ordinary image scanner?
2. How does MICR technology help to detect fraud in check payment?
3. Which factors will you consider while purchasing a monitor for your personal computer?
4. What is a head crash? How does it occur?
5. What is USB flash drive?
6. What is a BIOS? Which kind of memory is preferred in it, and why?
7. How is application software is different from system software?
8. Classify the operating systems based on their capabilities.
9. In what situation must the network have a gateway?
10. Which device will you prefer to form a network – hub or a switch? Justify your answer.

#### **Unit 2 :- (Number system)**

1. How can two numbers be subtracted only through addition? Explain with example.
2. Convert the following –
  - a)  $(10110101)_2 - (?)_{10}$
  - b)  $(5674)_8 - (?)_{10}$
  - c)  $(A1E2)_{16} - (?)_{10}$
  - d)  $(289)_{10} - (?)_8$
  - e)  $(593)_{10} - (?)_2$
  - f)  $(36541)_8 - (?)_{16}$
3.  $10101011 - 01111 = ?$
4.  $11100001 + 11111110111 = ?$
5.  $1111000 / 100 = ?$
6. Why NAND gate is known as universal gate?

#### **Unit 3 :- (Introduction to C)**

1. State true or false –  $a > \$amount$  is a valid identifier in C,

# UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

## Lecture-wise Plan

b> The equality operators have higher precedence than the relational operators.

c> Signed variables can increase the maximum positive range.

d> `printf("%d", scanf("%d",&num));` is valid statement

e> The modulus operators can be used only with integers

2. Write a program to prepare a grocery bill. For that enter the name of the items purchased, quantity in which it is purchased, and its price per unit. Then display the bill in the following format

\*\*\*\*\*BILL\*\*\*\*\*

Item	Quantity	Price	Amount
------	----------	-------	--------

---

Total amount to be paid

---

3. Write a program to read two floating point numbers. Add these numbers and assign the result to an integer. Finally display the value of all the three variables.

4. Find the error(s) –

```
Int n;  
float a b;  
double = a, b;  
complex a b;  
a,b : INTEGER  
long int a;b;
```

5. Find the error(s) –

```
int a = 9;  
float y = 2.0;  
a = b % a;  
printf("%d", a);
```

6. Write the output of the code →

```
int main()  
{  
int a = 2, b = 3, c = 4;  
a = b = c;  
printf("a=%d", a);  
return 0;  
}
```

7. Evaluate the expression –  $(x > y) + ++a \parallel !c$
8. Write a program to count the number of vowel in a given text.
9. Why do we include `<stdio.h>` in our program?
10. Write a program to calculate simple and compound interest .

### **Unit 4 :- (Decision Control and looping statements)**

1. Change the following for loop into do-while loop

```
int i;  
for(i=10 ; i>0 ; i--)
```

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lecture-wise Plan

- ```
printf("%d", i);
```
2. WAP to accept any number and print the number of digits in that program.
  3. WAP to print the sum of all odd numbers from 1 to 100.
  4. WAP that displays all the numbers from 1 to 100 that are not divisible by 2 as well as by 3.
  5. Write down the output

```
#include<stdio.h>
main()
{ int num = 10;
  for(;;--num)
  printf("%d", num);
}
```

6. Find errors  

```
#include<stdio.h>
main() {
int i,j;
for (i=1,j=0 ; i+j <=10 ; i++)
printf("%d", i);
j+=2;
}
```

7. WAP to print the pattern →

```
a> $ * * * *
   *$ * * *
   * * $ * *
   * * * $ *
   * * * * $
```

```
b> @
   @ @
   @ @ @
   @ @ @ @
   @ @ @ @ @
   @ @ @ @
   @ @ @
   @ @
   @
```

8. WAP to print the sum of the following series →

```
a> 1 + (1+2) – (1+2+3) + (1+2+3+4) - .....
```

```
b> -x + x2 – x3 + x4 .....
```

### Unit 5 :- (Arrays)

1. How are multi dimensional arrays useful?
2. What happens when an array is initialized with (a) fewer initializers as compared to its size?  
(b) more initializers as compared to its size?
3. For an array declared as int arr[50], calculate the address of arr[35], if base(arr) = 1000 and w =2.
4. Write a program that reads a square matrix of size n x n. Write a function int isUpperTraingular(int[][],int n) that returns 1 if the matrix is upper triangular.

# UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

## Lecture-wise Plan

- WAP to read two floating point arrays. Merge these arrays and display the resultant array.
- WAP to display the word HELLO in the following format

```
H
H E
H E L
H E L L
H E L L O
```

- Wap to count the number of charecters, words and lines in the given text.
- WAP to find the last instance of occurrence of a given string.
- Write a program to display a list of candidates. Prompt 100 users to cast their vote. Finally display the winner in the election.
- In a class there are 20 students. Each student is supposed to appear in three tests and two quizzes throughout the year. Make an array that stores the names of all these 20 students. Make five arrays that stores marks of three subjects as well as scores of two quizzes for all the students. Calculate the average and total marks of each student. Display the result.

### Unit 6 :- (Functions)

- How many types of storage classes does the C language support? Why do we need different types of such classes?
- What is the difference between formal and actual parameters?
- Write a function to reverse a string using recursion.
- What will happen when the actual parameters are less than the formal parameter in a function?
- WAP to compute  $F(x, y)$  where  $F(x, y) = F(x - y, y) + 1$  if  $y \leq x$  and  $F(x, y) = 0$  if  $x < y$ .
- Write a function to draw the following pattern on the screen

```
* * * * *
!           !
!           !
!           !
* * * * *
```

- Write a function to print a table of binomial coefficients which is given by the formula

$$B(m, x) = m! / (x! (m - x)!) \text{ where } m > x,$$

$$\text{hint : } B(m, 0) = 1, B(0, 0) = 1, \text{ and } B(m, x) = B(m, x-1) * [(m - x + 1) / x]$$

- Write a program to swap two integers using Call by Value method of passing arguments to a function.
- Find the output –

```
#include<stdio.h>
int prod (int x, int y)
```

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lecture-wise Plan

```
{
return (x * y);
}
main()
{
int x = 2, y = 3, z;
z = prod (x, prod(x,y) );
printf(“%d”,z);
return 0;
}
```

10. Find the output

```
#include <stdio.h>
int a;
static int func()
{
return a++;
}
main()
{
a = 10;
printf(“%d”, func() );
a*= 10;
printf(“%d”, func() );
return 0;
}
```

### **Unit 7 :- (Pointers)**

1. Explain the result of the following code

```
int num1 = 2, num2 =3;
int *p = &num1, *q = &num2;
*p++ =*q++;
```

2. What do you understand by null pointer? Discuss with example.
3. Differentiate between ptr++ and \*ptr++.
4. Can we subtract two pointer variables?
5. Can array names appear on the left side of the assignment operator? Why ?
6. WAP to print the lowercase letters into uppercase characters and the vice versa in the given string – “Happy NeW YeaR.”.
7. What is a dangling pointer?
8. What do you mean by wild pointer?
9. With the help of an example, explain how pointers can be used to dynamically allocate space for 2 D and 3D arrays.
10. State true or false—
  - a. Only one call to free() is enough to release the entire array allocated using calloc().
  - b. Ragged arrays consumes less memory space.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

- c. \*ptr++ will add 1 to the value pointed by ptr.
- d. Pointer constants cannot be changed.
- e. Adding 1 to a pointer variable will make it point 1 byte ahead of the memory location to which it is currently pointing.

### **Unit 8 :- (Structure, union, Files, Preprocessor directives)**

1. What do you mean by nested structure? Discuss.
2. Differentiate between structure and union.
3. WAP to create a structure with information given below. Then read and print the data.  
Employee[10]
  - (a) Emp\_id
  - (b) Name
    - (i) First name
    - (ii) Middle Name
    - (iii) Last Name
  - (c) Address
    - (i) Area
    - (ii) City
    - (iii) State
  - (d) Age
  - (e) Salary
  - (f) Designation.
4. WAP to read data from the keyboard and write it to a file. Read the contents stored in the file and display it on the screen.
5. WAP to copy a file using feof().
6. Can we have a C program that does not use any pre-processor directive?
7. What happens when the argument passed to the macro has multiple white space characters?



# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

**Title of Course:** Language Lab

**Course Code:** BBH192

**L-T-P scheme:** 0-0-6

**Course Credit:** 4

### **Objectives:**

1. This Course has been designed to impart advanced skills of Technical Communication in English through Language Lab. Practice Sessions to 1<sup>ST</sup> Semester UG students of Engineering & Technology.
2. To enable them to communicate confidently and competently in English Language in all spheres.

### **Learning Outcomes:**

1. This course will help the students to learn English very easily. Even the Hindi medium students can translate easily.
2. The technical communication will help the students to improve their speaking skills and drafting skill for engineering students.

### **Course Contents:**

#### **Exercises that must be done in this course are listed below:**

Exercise No.1: Phonetic symbols and transcription.

Exercise No. 2: Honing 'Listening Skill' and its sub skills through Language Lab Audio device;

Exercise No. 3: Honing 'Speaking Skill' and its sub skills;

Exercise No. 4: Master Linguistic/Paralinguistic features (Pronunciation/Phonetics/Voice modulation/Stress/ Intonation/ Pitch Accent) of connected speech;

Exercise No. 5: Honing 'Conversation Skill' using Language Lab Audio –Visual input; Conversational Practice Sessions (Face to Face/ via Telephone, Mobile phone & Role Play Mode);

Exercise No. 6: Introducing 'Group Discussion' through audio –Visual input and acquainting them with key strategies for success;

Exercise No. 7: G D Practice Sessions for helping them internalize basic Principles (turn-taking, creative intervention, by using correct body language, courtesies & other soft skills) of GD;

Exercise No. 8: Honing 'Reading Skills' and its sub skills using Visual/ Graphics/Diagrams /Chart Display/Technical/Non Technical Passages; Learning Global/ Contextual/ Inferential Comprehension;

Exercise No. 9: Honing 'Writing Skill' and its sub skills by using Language Lab Audio –Visual input;

Practice Sessions

Exercise No. 10: Group discussion

### **Text Book:**

1. Phonetic Symbol Guide Book by Geoffrey K. Pullum.
2. Dr.D.Sudharani: Manual for English Language Laboratory Pearson Education (WB edition),2010
3. Board of Editors: Contemporary Communicative English for Technical Communication Pearson Longman, 2010

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

**Title of Course: Computing Lab**

**Course Code: BBC195**

**L-T-P scheme: 0-0-6**

**Course Credit:4**

### **Introduction:**

This course is designed to familiarize students with the basic components of a computer, so as to be able to operate it and be able to interact with it, and carry out simple tasks. In addition, it will initiate the students into the discipline of Programming. It aims to start off the development of problem solving ability using computer programming. This course teaches not only the mechanics of programming, but also how to create programs that are easy to read, maintain, and debug. Students are introduced to the design principles for writing good programs regardless of the hardware and the software platforms.

### **Objective:**

Students will develop their ability to design, develop, test and document structured programs in C language.

**Learning Outcomes:** Students should be able to

1. Understand the basic terminology used in computer programming
2. Write, compile and debug programs in C language.
3. Use different data types in a computer program.
4. Design programs involving decision structures, loops and functions.
5. Explain the difference between call by value and call by reference
6. Understand the dynamics of memory by the use of pointers.
7. Enhance programming skills through problem solving and code development of small-size software applications.
8. Improve self-learning, teamwork and communication skills through project development practices.
9. Engage in continuing professional development under minimal guidance.

### **Course Contents:**

**Exercises that must be done in this course are listed below:**

- 1 Introduction to C programming
- 2 Structured Program Development in C
- 3 Flowchart and Algorithm
- 4 C Program Control
- 5 C Functions
- 6 C Arrays
- 7 C Pointers
- 8 C Characters and Strings
- 9 C Structures, Unions, Bit Manipulations and Enumerations
- 10 C File Processing

**UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**  
**Lab Manual**

**CYCLE – I**

1. Write a program to evaluate area of triangle using the formula  $\sqrt{s(s-a)(s-b)(s-c)}$
2. Write a program to swap two numbers.
3. Write a program to find the greatest of three numbers and print the numbers in ascending order.
4. Write a program to perform the arithmetic expression using switch statement.
5. Write a program to find a factorial of given number using do while statement.
6. Write a program to print all prime numbers upto 'N' numbers.
7. Write a program to print sum of 'N' natural numbers.
8. Write a program to find the total number of even integers and odd integers of 'N' numbers.
9. Write a program to find the sum of odd numbers and even numbers upto 'N' numbers.
10. Write a program to print the product of two matrices of any order.
11. Write a program to read 'N' number of students with 5 subject marks.
12. Write a program to find greatest of 'n' numbers using functions.
13. Write a program to print Fibonacci series using recursion.
14. Write a program to convert all lower case to uppercase characters.
15. Write a program to sort 5 city names in alphabetical order.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **CYCLE II**

16. Write a program to extract a string.
17. Write a program to implement the concept of call by value.
18. Write a program to implement the concept of call by reference.
19. Write a program to implement the concept of structure and union.
20. Write a program to access a variable using pointer.
21. Write a program to print the element of array using pointers.
22. Write a program to print the elements of a structure using pointers.
23. Write a program to display student information by initializing structures.
24. Write a program to pass structure as arguments to function and calculate total marks of 5 subjects.
25. Write a program to write integer data into file and read it from file.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

**Ex. No. 1**

**AREA OF TRIANGLE**

**AIM:**

To write a program for evaluating the area of triangle using the formula  $\sqrt{s(s-a)(s-b)(s-c)}$ .

**ALGORITHM:**

- Step1: Start the program.
- Step2: Get the inputs a, r, t and s.
- Step3: Calculate  $s = (a+b+c) / 2$ .
- Step4: Calculate  $\text{area} = \sqrt{s*(s-a)*(s-b)*(s-c)}$ .
- Step 5: Print the result 'area'.
- Step 6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<math.h>
void main()
{
    int a,b,c;
    float s,area;
    clrscr();
    printf("Enter the values of a,b,c: ");
    scanf("%d%d%d",&a,&b,&c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("The area of a triangle is =%f",area);
    getch();
}
```

**OUTPUT:**

Enter the values of a,b,c: 10 20 30

The area of a triangle is = 0.000000

**RESULT:**

Thus the C program to find the area of triangle using the formula  $\sqrt{s(s-a)(s-b)(s-c)}$  has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **SWAP TWO NUMBERS**

Ex. No. 2

#### **AIM:**

To write a program for swapping of two numbers.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Get the inputs a and b.
- Step3: Find  $a=a+b$ .
- Step4: Find  $b=a-b$ .
- Step 5: Find  $a=a-b$ .
- Step6: Print the result 'a' and 'b'.
- Step7: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    clrscr();
    printf("Enter the values of a and b: ");
    scanf("%d%d",&a,&b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("The values of a and b are: %d %d", a, b);
    getch();
}
```

#### **OUTPUT:**

Enter the values of a and b: 10 20

The values of a and b are: 20 10

#### **RESULT:**

Thus the C program to swap two numbers has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

### **Ex. No. 3 GREATEST OF THREE NUMBERS AND PRINT ASCENDING ORDER AIM:**

To write a program for finding the greatest of three numbers and printing the numbers in ascending order.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Get the inputs a, b and c.
- Step3: Check if((a>b) &&(a>c))
- Step4: Again check if(b>c)
- Step5: Then print the greatest number and display a, b, c.
- Step6: Else print the greatest number and display a, c, b.
- Step7: Check if((b<c) &&(b<a))
- Step8: Again check if(c<a)
- Step9: Then print the greatest number and display b, c, a.
- Step10: Else print the greatest number and display b, a, c.
- Step11: Check if((c<a) && (c<b))
- Step12: Again check if(a<b)
- Step13: Then print the greatest number and display c, a, b.
- Step14: Else print the greatest number and display c, b, a.
- Step15: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Enter the values of a, b and c: ");
scanf("%d%d%d", &a, &b, &c);
if(a<b && a<c)
{
    if(b<c)
```

# UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

## Lecture-wise Plan

```
        {
            printf("The greatest number is: %d", a);
            printf("The ascending order: %d%d%d", a, b, c);
        }
    else
        if(b>c)
        {
            printf("The greatest number is: %d", a);
            printf("The ascending order: %d%d%d", a, c, b);
        }
    }
else if(b<c && b<a)
    {
        if(c<a)
        {
            printf("The greatest number is: %d", b);
            printf("The ascending order: %d%d%d", b, c, a);
        }
        else
        {
            printf("The greatest number is: %d", b);
            printf("The ascending order: %d%d%d", b, a, c);
        }
    }
else
    if(b<a)
    {
        printf("The greatest number is: %d", c);
        printf("The ascending order: %d%d%d", c, b, a);
    }
    else
    {
        printf("The greatest number is: %d", c);
        printf("The ascending order: %d%d%d", c, a, b);
    }
}
```

**OUTPUT:** Enter the values of a, b and c: 6 4 5

The greatest number is: 6

The ascending order: 4 5 6

**RESULT:**

Thus the C program to find greatest of three and to print the numbers in ascending order has been successfully executed and verified.



# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

### Ex. No. 4                      ARITHMETIC EXPRESSION USING SWITCH STATEMENT

#### AIM:

To write a program for performing the arithmetic expression using switch statement.

#### ALGORITHM:

- Step1: Start the program.
- Step2: Display 1. Addition 2. Subtraction 3. Multiplication and 4. Division
- Step3: Get the input a and b.
- Step4: Get the choice.
- Step5: Switch(result)
- Step6: case '+': print the sum of a & b.
- Step7: case '-': print the difference of a & b.
- Step8: case '\*': print the multiplication of a & b.
- Step9: case '/': print the division of a & b.
- Step10: default: invalid option.
- Step11: Stop the program.

#### PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
int op;
clrscr();
printf("Enter the values of a & b: ");
scanf("%d%d", &a, &b);
printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
printf("Enter your choice: ");
scanf("%d", &op);
switch(op)
{
case 1 :printf("Sum of %d and %d=%d", a, b, a+b);
break;
case 2 :printf("Subtraction of %d and %d=%d", a, b, a-b);
break;
case 3 :printf("Multiplication of %d and %d=%d", a, b, a*b);
break;
case 4 :printf("Division of %d and %d=%d", a, b, a/b);
break;
default : printf(" Enter Your Correct Choice.");
break;
}
```

```
}  
getch();  
}
```

**OUTPUT:**

Enter the values of a & b: 10 20

1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice: 1

Sum of 10 and 20 = 30

**RESULT:**

Thus the C program for arithmetic expression using switch statement has been successfully executed and verified.

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

### Ex. No. 5                      FACTORIAL OF A NUMBER USING DO WHILE STATEMENT

#### AIM:

To write a program for finding the factorial of a given number using do while statement.

#### ALGORITHM:

Step1: Start the program.  
Step2: Assign  $f=i=1$ .  
Step3: Get the input n.  
Step4: do .. the following.  
Step5: Find  $f=f*i$   
Step6: Increment  $i=i+1$   
Step7: Repeat from step5 to step6 till while( $i \leq no$ ).  
Step8: Then print f.  
Step9: Stop the program.

#### PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,f;
    f=i=1;
    clrscr();
    printf("Enter a number: ");
    scanf("%d",&n);
    do
    {
        fact*=i;
        i++;
    }while(i<=no);
    printf("Factorial of %d=%d\n", no, fact);
}
```

#### OUTPUT:

```
Enter a number: 5
Factorial of 5 = 120
```

#### RESULT:

Thus the C program for finding the factorial of a given number using do while statement has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Ex. No. 6

**GENERATE PRIME NUMBERS UPTO N NUMBERS**

**AIM:** To write a program for printing all prime numbers upto N numbers.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Get the n value.
- Step3: for(i=1;i<=n;i++)
- Step4: Repeat a, b, c, d & e
  - a) Assign fact=0
  - b) for(j=1;j<=n;j++) repeat c & d
  - c) if i percentage j equal to zero
  - d) fact equal to fact added with one
  - e) if fact equal to 2 print i as prime number
- Step5: Display the prime number till n<sup>th</sup> number.
- Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,fact,j;
    printf("Enter the range: ");
    scanf("%d",&n);
    printf("Prime numbers are: \n");
    for(i=1;i<=n;i++)
    {
        fact=0;
        for(j=1;j<=n;j++)
        {
            if(i%j==0)
                fact++;
            if(f==2)
                printf("%d ",i);
        }
        getch();
    }
}
```

**OUTPUT:**

Enter the range: 10  
Prime numbers are: 3 5 7

**RESULT:**

Thus the C program for printing all prime numbers upto N numbers has been successfully executed and verified.

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

Ex. No. 7

SUM OF N NATURAL NUMBERS

**AIM:**

To write a program for printing the sum of N natural numbers.

**ALGORITHM:**

Step1: Start the program.  
Step2: Get the n value.  
Step3: Initialize i=0 and sum=0.  
Step4: Perform from step 5 to step 6 until i<=n  
Step5: i++  
Step6: sum+=i  
Step7: Print the sum.  
Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i=0,sum=0;
    clrscr();
    printf("Enter the Limit : ");
    scanf("%d",&n);
    while(i<=n)
    {
        i++;
        sum+=i;
    }
    printf("Sum of %d natural numbers = %d",n,sum);
    getch();
}
```

**OUTPUT:**

Enter the Limit : 10

Sum of 10 natural numbers = 55

**RESULT:**

Thus the C program for printing the sum of N natural numbers has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **Ex. No. 8      TOTAL NUMBER OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS**

**AIM:** To write a program for finding the total number of even integers and odd integers of 'N' numbers.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Declare int i, n, odd=0 and even=0;
- Step3: Get the n value
- Step4: for( i=0;i<=n;i++) do the following step.
  - a) Check if(i%2==0)
  - b) even=even+1;
  - c) Else odd=odd+1;
- Step5: Print the odd and even value.
- Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,odd=0,even=0;
    clrscr();
    printf("Enter the n value: ");
    scanf("%d",&n);
    for(i=1;i<n;i++)
    {
        if(i%2==0)
            even=even+1;
        else
            odd=odd+1;
    }
    printf("The total number of odd integers =%d",odd);
    printf("The total number of even integers =%d",even);
    getch();
}
```

**OUTPUT:**

```
Enter the n value: 10
The total number of odd integers =5
The total number of even integers = 5
```

**RESULT:**

Thus the above C program for finding the total number of even integers and odd integers of 'N' numbers has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

### **Ex. No. 9 SUM OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS AIM:**

To write a program for finding the sum of even integers and odd integers of 'N' numbers.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Declare int i, n, odd=0 and even=0;
- Step3: Get the n value
- Step4: for( i=0;i<=n;i++) do the following step.
  - a) Check if(i%2==0)
  - b) even=even+i;
  - c) Else odd=odd+i;
- Step5: Print the odd and even value.
- Step6: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,n,sum,even=0,odd=0;
clrscr();
printf("Enter any number: ");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
if(i%2==0)
even=even+i;
else
odd=odd+i;
}
printf("Sum of even integer is: %d",even);
printf("Sum of odd integer is: %d",odd);
getch();
}
```

#### **OUTPUT:**

```
Enter any value: 5
Sum of even integer is: 6
Sum of odd integer is: 9
```

#### **RESULT:**

Thus the C program for finding the sum of even integers and odd integers of 'N' numbers has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

**Ex. No. 10**

**PRODUCT OF TWO MATRICES OF ANY ORDER**

**AIM:**

To write a program for finding the product of two matrices of any order.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Declare int Matrix A[9][9] , MatrixB[9][9] , Matrixproduct [9][9].
- Step3: Declare int n , i , j , k, Row1 , Row2 , Column1 , Column2.
- Step4: Enter the order of Matrix A Row1, Column1.
- Step4: Enter the order of Matrix B Row2, Column2.
- Step5: Check if(Column1 == Row2)
- Step6: Enter the elements of Matrix A and B using for loops.
- Step7: Find Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k] \* Matrix B[k][j] using for loops.
- Step7: Print the resultant matrix Matrixproduct[i][j] using for loop.
- Step8: Else print invalid order so multiplication not possible.
- Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int Matrix A[9][9] , MatrixB[9][9] , Matrixproduct [9][9] ;
    int n , i , j , k; /* 'i' used for rows and 'j' used for columns */ int Row1 , Row2 ,
    Column1 , Column2; clrscr();

    printf(" Enter the order of Matrix A\n");
    scanf("%d * %d " , &Row1 , &Column1);
    printf(" Enter the order of Matrix B\n");
    scanf("%d * %d " , &Row2 , &Column2);
    if(Column1 == Row2)
    {
        printf(" Enter the elements of Matrix A\n");
        for(i=0 ; i<Row1 ; i++)
        {
            for(j=0 ; j<Column1 ; j++)
            {
                scanf("%d" , &Matrix A[i][j] );
            }
        }
        printf(" Enter the elements of Matrix B\n");
```



```
for(i=0 ; i<Row2 ; i++)
{
    for(j=0 ; j<Column2 ; j++)
    {
        scanf("%d" , &Matrix B[i][j] );
    }
}
for(i=0 ; i<Row1 ; i++)
{
    for(j=0 ; j<Column2 ; j++)
    {
        Matrixproduct[i][j] = 0 ;
        for(k=0 ; k<Row2 ; k++)
        {
            Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k] * Matrix B[k][j] );
        }
    }
}
printf(" Product Matrix\n");
for(i=0 ; i< Row1 ; i++)
{
    for(j=0 ;j< Column2;j++)
    {
        printf("%d" , Matrixproduct[i][j] );
    }
    printf("\n");
}
}
else
    printf(" Invalid order so Multiplication not possible\n");
}
```

**OUTPUT:**

```
Enter the order of Matrix A
2 * 2
Enter the order of MatrixB
2 * 2
Enter the elements of Matrix A
1
2
3
4
Enter the elements of Matrix B
5
6
7
8
Product Matrix
19  22
43  50
```

**RESULT:**

Thus the C program for finding the product of two matrices of any order has been successfully executed and verified.

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

Ex. No. 11                      READ 'N' NUMBER OF STUDENTS WITH 5 SUBJECT MARKS

**AIM:**

To write a program for reading 'N' number of students with 5 subject marks.

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize a character array n and integer array r and s.

Step3: Initialize integer i, j and n.

Step3: Read the value of n.

Step4: for(i=0;i<n;i++)

    a) Enter rollno,name,,,,,,

    b) Read these and enter 5 subject marks using for loop and array.

Step5: Display n[i],r[i],s[i][j]

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char n[20][10];
    int i,j,r[20],s[20][6];
    printf("Enter n value: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter name,rollno,....");
        scanf("%s%d",&n[i],&r[i]);
        printf("Enter 5 subject marks:");
        s[i][5]=0;
        for(j=0;j<5;j++)
        {
            scanf("%d",s[i][j]);
            s[i][5]=s[i][5]+s[i][j];
        }
    }
    printf("The data entered is: \n");
    for(i=0;i<n;i++)
    {
        printf("%s\t%d\t",n[i],r[i]);
        for(j=0;j<5;j++)
            printf("%d\t",s[i][j]);
    }
    getch();
}
```

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **OUTPUT:**

```
Enter n value: 1
Enter name,rollno,...Eswar 20
Enter 5 subject marks:
10 50 34 06 42
The data entered is:
Eswar 20          10    50    34    06    42
```

### **RESULT:**

Thus the C program for reading 'N' number of students with 5 subject marks has been successfully executed and verified.

# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

### Ex. No. 12                      GREATEST OF 'N' NUMBERS USING FUNCTION

**AIM:** To write a program for finding greatest of 'n' numbers using function.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Initialize integer a, b and c.
- Step3: Read the value of a,b and c.
- Step4: Call the function large().
  - a) Check if((a>b) && (a>c)) then print a is greater.
  - b) Check elseif (b>c) then print b is greater.
  - c) Check else print c is greater.
- Step5: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    printf(" Enter the value of a,b and c: ");
    scanf("%d, %d, %d", &a, &b, &c);
    large(a,b,c);
    getch();
}

large(int a, int b, int c)
{
    if((a>b) && (a>c))
        print("%d is greater than %d, %d", a, b, c);
    elseif (b>c)
        print("%d is greater than %d, %d", b, a, c);
    else
        print("%d is greater than %d, %d", c, a, b);
}
```

#### **OUTPUT:**

```
Enter the value of a,b and c: 10 30 20
30 is greater than 10, 20
```

#### **RESULT:**

Thus the C program for finding greatest of 'n' numbers using function has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Ex. No. 13

FIBONACCI SERIES USING RECURSION

### **AIM:**

To write a program for finding Fibonacci series using recursion.

### **ALGORITHM:**

Step1: Start the program.

Step2: Initialize a function as int Fibonacci(int).

Step3: Initialize integer i=0, c and n in main function.

Step3: Read the value of n.

Step4: Within for loop call the Fibonacci(int) recursively.

Step5: In Fibonacci(int) function calculate ( Fibonacci(n-1) + Fibonacci(n-2) ) recursively and return the value.

Step6: Print the result.

Step7: Stop the program.

### **PROGRAM:**

```
#include<stdio.h>
int Fibonacci(int);
int main()
{
    int n, i = 0, c;
    printf("Enter the n value: ");
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    }
    return 0;
}

int Fibonacci(int n)
{
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

**UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**  
**Lab Manual**

**OUTPUT:**

Enter the n value: 9

Fibonacci series: 0 1 1 2 3 5 8 13 21

**RESULT:**

Thus the C program for finding Fibonacci series using recursion has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **Ex. No. 14                      LOWER CASE TO UPPERCASE CHARACTERS**

**AIM:** To write a program for converting all lower case to uppercase characters.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Take a string a function of return value data type is void str upper.
- Step3: Read a string.
- Step4: While (s[i]! ='\0') the do the following
  - a)if((s[i]>='a') &&(s[i]<='z'))
  - b)s[i]=s[i]-32;
  - c)i++;
- Step5: Display changed string.
- Step6: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str;
    printf("Enter a string: ");
    scanf("%s",str);
    to_str_upper(char[]);
    printf("Changed to: %s",str);
}
void to_str_upper(char[])
{
    int i=0;
    while(s[i]!='\0')
    {
        if((s[i]>='a') && (s[i]>='z'))
        s[i]=s[i]-32;
        i++;
    }
}
```

#### **OUTPUT:**

```
Enter a string : g nec
changed to: GNEC
```

#### **RESULT:**

Thus the C program for converting all lower case to uppercase characters has been successfully executed and verified.



# UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR

## Lab Manual

Ex. No. 15                      SORT 5 CITY NAMES IN ALPHABETICAL ORDER

**AIM:**

To write a program for sorting 5 city names in alphabetical order.

**ALGORITHM:**

Step1: Start the program.

Step2: Using for loop and array get the city name.

Step3: Using loop for(i=65;i<122;i++) and for(j=0;j<5;j++)

a) Check if(city[j][0]==i)

b) Display the sorted list of cities.

Step4: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    ch city[5][20];
    int i,j;
    clrscr();
    printf("Enter the names of cities...\n\n");
    for(i=0;i<5;i++)
        scanf("%s",&city[i]);
    printf("Sorted list of cities...\n\n");
    for(i=65;i<122;i++)
    {
        for(j=0;j<5;j++)
        {
            if(city[j][0]==i)
                printf("\n%s",city[j]);
        }
    }
}
```

**OUTPUT:**

```
Enter the names of cities: Hyderabad Chennai Bombay Goa Vizag
Sorted list of cities:
Bombay
Chennai
Goa
Hyderabad
Vizag
```

**RESULT:** Thus the C program for sorting 5 city names in alphabetical order has been successfully executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

**Ex. No. 16**

**EXTRACTS THE PART OF A STRING**

**AIM:**

To write a program for extracting the part of a string.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Declare the character array s[30] and r[30].
- Step3: Declare the integer variables i, j, m & n.
- Step4: Get the input string using gets().
- Step5: Get the value of m and n for extracting from the input string.
- Step6: Initialize j=0.
- Step7: Using a loop for(i=n-1;i<m+n-1;i++)
  - a) Assign r[j]=s[i];
  - b) Increment J by 1.
- Step8: Print the extracted part of the string.
- Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[30],r[30];
    int i,j,m,n;
    clrscr();
    printf("Enter a string: ");
    gets(s);
    printf("Enter the values of m & n: ");
    scanf("%d%d",&m,&n);
    j=0;
    for(i=n-1;i<m+n-1;i++)
    {
        r[j]=s[i];
        j++;
    }
    printf("The extracted part of string %s: ",r);
    getch();
}
```

**UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**  
**Lab Manual**

**OUTPUT:**

Enter a string: Gurunanak  
Enter the values of m & n: 3 5  
The extracted part of string: run

**RESULT:**

Thus the C program for extracting a part from the given string was executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Ex. No. 17

CALL BY VALUE

**AIM:**

To write a program to increment the value of an argument using call by value.

**ALGORITHM:**

Step1: Start the program.

Step2: Declare the integer variable x and a integer function incr()

Step3: Initialize x=7.

Step4: Pass the x value to the function incr(x).

a) Within the function increment the x value by 1.

b) Return the value.

Step5: Print the original value and incremented value of x.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<string.h>
main()
{
    int x;
    int incr(int n);
    printf("***Call by Value***\n");
    x = 7;
    printf("Original value of x is: %d/n: ", x);
    printf("Value of incr(x) is: %d/n ", incr(x));
    printf("The value of x is: %d/n: ", x);
}

/* Function increments n */
int incr(int n)
{
    n = n + 1;
    return n;
}
```

**OUTPUT:**

Original value of x is: 7

Value of incr(x) is : 8

The value of x is: 7

**RESULT:**

Thus the C program to increment the value of an argument using call by value was executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

**Ex. No. 18**

**CALL BY REFERENCE**

**AIM:**

To write a program for swapping two values using call by reference method.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Assign the integer variable a=10 and b=20.
- Step3: Call the swap() function.
- Step4: Swap the values using pointer.
- Step5: Print the original value and swapped value of a & b.
- Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void swap( int *x, int *y )
{
    int t ;
    t = *x ;
    *x = *y ;
    *y = t ;
    printf( "\nx = %d y = %d", *x,*y);
}

int main()
{
    int a = 10, b = 20 ;
    swap ( &a, &b ) ;
    printf ( "\na = %d b = %d", a, b ) ;
    getch();
}
```

**OUTPUT:**

```
a=10 b=20
x=20 y=10
```

**RESULT:**

Thus the C program to swap two values using call by reference method was executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan** **STRUCTURE**

**Ex. No. 19(a)**

### **AIM:**

To write a program for displaying student information by initializing structures.

### **ALGORITHM:**

Step1: Start the program.

Step2: Initialize a structure student with name as character array and roll number and age as integer.

Step3: In the main program create a object s1 for the structure student.

Step4: Using the object s1 print the student name, roll number and age.

Step6: Stop the program.

### **PROGRAM:**

```
#include<stdio.h>
struct student
{
    char name[10];
    int rollno;
    int age;
};
main()
{
    static struct student s1;
    clrscr();
    printf("Enter the name, rollno & age");
    scanf("%s%d%d\n",&s1.name,&s1.rollno,&s1.age);
    printf("%s %d %d",s1.name,s1.rollno,s1.age); getch();
}
```

### **OUTPUT:**

```
Enter name, rollno & age
Ravi 11 25
Ravi 11 25
```

### **RESULT:**

Thus the C program to display student information by initializing structures was executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

Ex. No. 19(b)

UNION

**AIM:**

To write a program for implementing the concept of union data type.

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize a union Data with Str as character array, i as integer and f as float.

Step3: In the main program create a variable name data for the union Data.

Step4: Using the variable and member access operator print all the members of the union Data.

Step5: Stop the program.

**PROGRAM:**

```
#include <stdio.h>
#include <string.h>
union Data
{
    int i;
    float f;
    char str[20];
};

int main()
{
    union Data data;
    data.i = 10;
    printf( "data.i : %d\n", data.i);
    data.f = 220.5;
    printf( "data.f : %f\n", data.f);
    strcpy( data.str, "C Programming");
    printf( "data.str : %s\n", data.str);
    return 0;
}
```

**OUTPUT:**

```
data.i : 10
data.f : 220.500000
data.str : C Programming
```

**RESULT:**

Thus the C program to implement the concept of union data type was executed and verified.

# UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

## Lecture-wise Plan

Ex. No. 20

ACCESS THE VALUE OF VARIABLES USING POINTER

### AIM:

To write a program for accessing the value of variables using pointer.

### ALGORITHM:

- Step1: Start the program.
- Step2: Declare integer as a, b, c and two pointer variables \*p1 & \*p2.
- Step3: Initialize a=12 and b=4.
- Step4: Assign the a & b values to the pointer variables p1 & p2.
- Step5: Perform arithmetic operations.
- Step6: Print the address of a & b and print the a, b, c, x & y values.
- Step7: Stop the program.

### PROGRAM:

```
#include<stdio.h>
main()
{
    int a,b,*p1,*p2,x,y,z;
    clrscr();
    a=12,b=4;
    p1=&a; p2=&b;
    x=*p1**p2-6;
    y=(4-*p2)**p1+10;
    printf("Address of a=%d\n",p1);
    printf("Address of b=%d\n",p2);
    printf("a=%d,b=%d\n",a,b);
    printf("x=%d,y=%d\n",x,y);
    *p2=*p2+3; *p1=*p2-5;
    z=*p1**p2-6;
    printf("a=%d,b=%d\n",a,b);
    printf("z=%d\n",z);
    getch();
}
```

**OUTPUT:** Address of a = 65543

Address of b = 64455

a = 12 b = 4

x =     y =

z=42

### RESULT:

Thus the C program to access the value of variables using pointer was executed and verified.



# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

**Ex. No. 21                      PRINT THE ELEMENT OF ARRAY USING POINTERS**

**AIM:**

To write a program for printing the element of array using pointers.

**ALGORITHM:**

Step1: Start the program.  
Step2: Declare integer array a[5] and a pointer variable \*p=&a[0]  
Step3: Initialize i as integer.  
Step4: Using the for loop for(i=0;i<5;i++)  
Step5: print the value of \*(p+i).  
Step6: Then using the for loop for(i=0;i<5;i++)  
Step7: Print the value of (p+1).  
Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
main()
{
    int a[5]={5,4,6,8,9};
    int *p=&a[0];
    int i;
    clrscr();
    for(i=0;i<5;i++)
        printf("%d",*(p+i));
    for(i=0;i<5;i++)
        printf(" %u\n",*(p+i));
    getch();
}
```

**OUTPUT:**

```
1 2 3 4 5
1 2 3 4 5
```

**RESULT:**

Thus the C program to print the element of array using pointers was executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

Ex. No. 22

PRINT THE ELEMENTS OF A STRUCTURE USING POINTERS

**AIM:**

To write a program printing the elements of a structure using pointers.

**ALGORITHM:**

Step1: Start the program.  
step2: Take a character array name, a number and price in structure  
step3: In main take a struct variable product and a pointer  
Step4: Using a loop for(\*ptr=product;ptr<product+3;ptr++)  
Step5: Read the value by using array operator  
ptr->name,ptr->no,ptr->price  
step6: Display name,no,price.  
Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
struct invest
{
    char name[20];
    int number;
    float price;
};
main()
{
    struct invest product[3],*ptr;
    clrscr();
    printf("input\n\n");
    for(*ptr=product[3];ptr<product+3;ptr++)
        scanf("%s%d%f",&ptr->name,&ptr->number,&ptr->price);
    printf("\nResult: \n\n");
    ptr=product;
    while(ptr<product+3)
    {
        printf("%20s%5d%10.2f\n",ptr->name,ptr->number,ptr->price);
        ptr++;
    }
    getch();
}
```

**UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**  
**Lab Manual**

**OUTPUT:**

Raja

11

120

Result:

Raja

11

120

**RESULT:**

Thus the C program to print the elements of a structure using pointers was executed and verified.

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **Ex. No. 23 DISPLAY COLLEGE ADDRESS USING STRUCTURES AND POINTERS AIM:**

To write a program for displaying college address using structures and pointers.

#### **ALGORITHM:**

- Step1: Start the program.
- Step2: Take name, location and city inside the collegeaddress structure.
- Step3: Enter the required data.
- Step4: Print the result.
- Step5: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
struct collegeaddress
{
    char name[20],location[20],city[20];
};
main()
{
    struct collegeaddress add,*ptr;
    p=&add;
    p->name={"Annamalai University"};
    p->location={"Annamalainagar"};
    p->city={"Chidambaram"};
    printf("%s%s%s",p->name,p->location,p->city);
}
```

#### **OUTPUT:**

Annamalai University Annamalainagar Chidambaram

#### **RESULT:**

Thus the C program to display college address using structures and pointers was executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

**Ex. No. 24**                      **PASS STRUCTURE AS ARGUMENT TO FUNCTION**

**AIM:**

To write a program for passing structure as argument to function and calculate total marks of 5 subjects.

**ALGORITHM:**

- Step1: Start the program.
- Step2: Inside the structure ex2 declare 6 integers.
- Step3: Declare structure ex2 as s1.
- Step4: Declare structure ex2 as s2,ex2 as fun().
- Step5: Display the message as enter the marks.
- Step6: Take value of the subjects from the user.
- Step7: Store the return value in s2.total.
- Step8: Print the value of s2.total.
- Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
struct ex2
{
    int m1,m2,m3,m4,m5,total;
};
main()
{
    struct ex2 s1;
    struct ex2 s2;
    struct ex2 fun();
    printf("enter the marks");
    scanf("%d%d%d%d%d",&s1.m1,&s1.m2,&s1.m3,&s1.m4,&s1.m5);
    s2=fun(s1);
    printf("%d",s1.total);
}
struct ex2 fun(s1)
struct ex2 s3;
{
    s3.total=s1.m1+s1.m2+s1.m3+s1.m4+s1.m5;
    return(s3);
}
```

# **UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR**

## **Lecture-wise Plan**

### **OUTPUT:**

Enter the marks  
10 20 30 40 50  
150

### **RESULT:**

Thus the C program to pass structure as argument to function and calculate total marks of 5 subjects was executed and verified.

# **UNIVERSITY OF ENGINEERING AND MANAGEMENT, JAIPUR**

## **Lab Manual**

**Ex. No. 25**                      **WRITE INTEGER DATA INTO FILE AND READ IT FROM FILE**

**AIM:**

To write a program for writing integer data into file and read it from file.

**ALGORITHM:**

Step1: Start the program.  
Step2: Initialize integer num.  
Step3: Declare FILE \*f2.  
Step4: Open the file f2 using fopen() in write mode.  
Step5: Get the integer from user and write it into the file using putw().  
Step6: Close the file.  
Step7: Open the file f2 using fopen() in read mode.  
Step8: Read the integer using getw().  
Step9: Print the integer.  
Step10: Close the file.  
Step11: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
main()
{
    int num;
    FILE *f2;
    f2=fopen("data.int","w");
    scanf("%d",&num);
    putw(num,f2);
    fclose(f2);
    f2=fopen("data.int","r");
    num=getw(f2);
    printf("%d",num);
    fclose(f2);
}
```

**OUTPUT:**

12  
12

**RESULT:**

Thus the C program to write integer data into file and read it from file was executed and verified.

