Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Programming in C

Experiment 1 : Installation, Environment Setup and ~~setting~~ starting with C language.

1. Write a C program to print "Hello World".

```
#include <studio.h>
int main () {
    printf ("Hello World");
    return 0;
}
```

2. Write a C program to print the address in multiple lines (new lines).

```
#include <studio.h>
int main () {
    printf ("Anushka\n");
    printf ("House no. 145 \n");
    printf ("sector 13, Jammu\n");
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. Write a program that prompts the user to enter their name and age.

```
#include <studio.h>
int main() }
    Char name [50];
    int age;
    printf ("Enter your name");
    scanf ("%49s", name);
    printf ("Enter your age");
    Scanf ("%d", &age);
    printf ("name = %s\n", name);
    printf (" age = %d", age);
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. Write a C program to add two numbers, take number from user.

```
#include <studio.h>

int main () {
    int Number1, Number2, Sum ;

    printf ("Enter number 1");
    scanf ("%d", & Number1);

    printf ("Enter number 2");
    scanf ("%d", &Number2);

    Sum=Number1 +Number2;

    printf ("Sum= %d", Sum);

    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 2: Operators:

1. WAP a C program to calculate the area and perimeter of a rectangle based on its length and width.

```
#include <studio.h>
int main ()}
    int length, width, area, perimeter ;
    printf("Enter the length");
    scanf ("%d", &length);

    printf ("Enter the width");
    scanf ("%d", &width);

    area=length * width;
    perimeter = 2*(length+width);

    printf("Area=%d\n", area);
    printf ("Perimeter = %d\n", perimeter);

    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. Write a C program to convert temp. from Celsius to Fahrenheit using the formula: F = (C*9/5)+32.

```
#include <stdio.h>
int main()
{
    int celsius, fahrenheit;
    printf("Enter temperature in celsius \n");
    scanf("%d", &celsius);
    fahrenheit = ((9/5)*celsius)+32);
    printf("Faherheit = %d", faherheit);
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 3.1 : Conditional statements

1. WAP to take the check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle or scalene. Take sides of the triangle as input from a user.

```
#include <studio.h>
int main ()
{
   int side1, side2, side3;
   printf("Enter three sides of a triangle:");
   scanf("%d %d %d", &side1, &side2, &side3);
   if ((side1 + side2 > side3) && (side1 + side3 > side2) && (side2 + side3 > side1))
   {
    if (side1 == side2 && side2 == side3)
    {
       printf("Equilateral triangle");
    }
    else if ((side1 * side1 + side2 * side2 == side3 * side3) ||
          (side1 * side1 + side3 * side3 == side2 * side2) ||
          (side2 * side2 + side3 * side3 == side1 * side1))
    {
     printf("Right-angled triangle");
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
else if (side1 == side2 || side2 == side3 || side1 == side3)
{
    printf ("Isosceles triangle");
}
 else
{
   printf ("Scalene triangle");
}

}
 else
    printf ("Not a valid triangle");
}
 return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges.

```
#include <stdio.h>

int main() {
    float weight, height, bmi;

    printf("Enter weight (in kg): ");
    scanf("%f", &weight);

    printf("Enter height (in meters): ");
    scanf("%f", &height);

bmi = weight / (height * height);
printf("Your BMI is : %.2f\n", bmi);

if (bmi < 15)
    printf("Starvation");
else if (bmi <= 17.5)
    printf("Anorexic");
else if (bmi <= 18.5)
    printf("Underweight");
```

Teacher's Signature _____

Expt. No. _____ Page No. [ ] Date [ ][ ][ ][ ][ ][ ]

```
else if (bmi <= 24.9)
    printf ("Ideal");
else if (bmi <= 39.9)
    printf ("obese");
else if (bmi <= 25.9);
    printf ("Overweigut");
else
    printf ("Morbidly Obese");

 return 0;
}
```

Teacher's Signature

Expt. No. _____ Page No. [ ] Date [ ][ ][ ][ ][ ][ ]

3. WAP to check if three points $(x1, y1)$, $(x2, y2)$ and $(x3, y3)$ are collinear or not.

```
#include <stdio.h>

int main () {
    float x1, y1, x2, y2, x3, y3, area;

    printf ("Enter coordinates of first point: ");
    scanf ("%f %f", &x1, &y1);

    printf ("Enter coordinates of second point: ");
    scanf ("%f %f", &x2, &y2);

    printf ("Enter coordinates of third point: ");
    scanf ("%f %f", &x3, &y3);

    area = x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2);

    if (area == 0)
        printf ("The points are collinear.\n");
    else
        printf ("The points are not collinear.\n");
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. According to the gregorian calendar, it was monday on the date 01/01/01. If any year is input through the keyboard write a program to find out what is the day on 1st january of the year.

```
#include <stdio.h>

int main() {
    int year, day;
    printf("Enter year: ");
    scanf("%d", &year);

    day = (year + (year-1)/4 - (year-1)/100 + (year-1)/400) % 7;

    printf("On 1st january %d, the day was: ", year);
    switch (day) {
        case 0: printf("Sunday\n"); break;
        case 1: printf("Monday\n"); break;
        case 2: printf("Tuesday\n"); break;
        case 3: printf("Wednesday\n"); break;
        case 4: printf("Thursday\n"); break;
        case 5: printf("Friday\n"); break;
        case 6: printf("Saturday\n"); break;
    }
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

5. WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum no. of rect. should be three.

```
#include <stdio.h>

int main () {
    int l1, b1, l2, b2, l3, b3;
    int p1, p2, p3;

    printf ("Enter length and breadth of Rectangle 1: ");
    scanf ("%d %d", &l1, &b1);

    printf ("Enter length and breadth of Rectangle 2: ");
    scanf ("%d %d", &l2, &b2);

    printf ("Enter length and breadth of Rectangle 3: ");
    scanf ("%d %d", &l3, &b3);
    p1 = 2 * (l1 + b1);
    p2 = 2 * (l2 + b2);
    p3 = 2 * (l3 + b3);
    int max = (p1 > p2) ? ((p1 > p3) ? p1 : p3) : ((p2 > p3) ? p2 : p3);
    printf ("The highest perimeter is: %d\n", max);
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 3.2: Loops

1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative and zeroes entered.

```
#include <stdio.h>

int main () {
    int num, pos = 0, neg = 0, zero = 0;
    char choice;

    do {
        printf ("Enter a number : ");
        scanf ("%d", &num);

        if (num > 0)
            pos++;
        else if (num < 0)
            neg++;
        else
            zero++;

        printf("Do you want to enter another number? (y/n): ");
        scanf ("%c", &choice);
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
} while (choice == 'y' || choice == 'Y');

    printf ("\n\Positive numbers : %d \n", pos);
    printf ("Negative numbers : %d \n", neg);
    printf ("zeroes : %d \n", zero);

    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting. Num*1 = Num.

```
#include <stdio.h>

int main () {
    int num, i;
    printf ("Enter a number: ");
    scanf ("%d", &num);

    for (i = 1; i <= 10; i++) {
        printf ("%d * %d = %d \n", num, i, num *i);

    }

    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. WAP to generate the following set of output:

(a) 1
2 3
4 5 6

```
#include <stdio.h>
int main () {
    int i, j, num = 1;

    for (i=1; i <=3 ; i++) {
        for (j =1; j<= i ; j++) {
            printf ("%d", num);
            num++;
        }
        printf ("\n");
    }

    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

```
#include <stdio.h>
int main() {
    int n=5 ; i , j, coef ;

    for (i =0 ; i <n; i++) {
        for (j= 0 ; j <= i ; j++) {
            if (j == 0 || j== i)
                coef =1;
            else
                coef = coef * (i-j+1) /j;

            printf ("%d", coef);
        }
        printf ("\n");
    }
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 yrs. WAP to determine the population at the end of each year in the last decade.

```
#include <stdio.h>

int main () {
    float population = 100000;
    int year;

    printf ("Initial population = 100000\n");

    for (year = 1; year <= 10; year++) {
        population = population + (population * 0.10);
        printf ("Population at end of year %d = %.0f\n",
                year, population);

    }

    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

5. WAP to print all Ramanujan Numbers up to a limit (sum of two cubes in two ways). Example: $1729 = 12^3 + 1^3 = 10^3 + 9^3$

```
#include <studio.h>

int main () {
    int a, b, c, d; limits;
    printf ("Enter limit: ");
    scanf ("%d", &limit);

    for (int n=1; n <= limit; n++) {
        for (a = 1 ; a*a*a < n; a++) {
            for (b = a; b*b*b < n; b++) {
                for (c = a; c*c*c < n; c++) {
                    for (d = c; d*d*d < n; d++) {
                        if (a*a*a + b*b*b == c*c*c + d*d*d && a!=c && b!=d){
                            if (a*a*a + b*b*b == n)
                                printf ("%d is a Ramanujan number \n", n);
                        }
                    }
                }
            }
        }
    }
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 4: Variable and Scope of Variable

1. Declare a global variable outside all functions and use it inside various functions to understand its accessibility.

```
#include <stdio.h>

int x = 10;

void display () {
    printf ("Inside display (): x = %d \n", x);
}

int main () {
    printf ("Inside main (): x = %d \n", x);
    display ();
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. Declare a local variable inside a function and try to access it outside the function. Compare this with accessing the global variable from within the function.

```
#include <stdio.h>

void test() {
    int num = 20;
    printf("Inside test(): num = %d\n", num);
}

int main() {
    test();

    printf("Local variable 'num' cannot be accessed here.\n");
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. Declare variables within different code blocks (enclosed by curly braces) and test their accessibility within and outside those blocks.

```
#include <stdio.h>

int main () {
    int x = 10;
    {
        int y = 20;
        printf ("Inside block : x = %d , y = %d \n", x, y);
    }

    printf ("Outside block : x = %d \n", x);
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. Declare a static local variable inside a function. Observe how its value persists across function calls.

```
#include <stdio.h>

void counter () {
    static int count = 0;
    count++;
    printf ("function called %d times\n", count);
}

int main () {
    counter ();
    counter ();
    counter ();
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 5 : Array:

1. WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.

```
#include <stdio.h>
int main () {
    int n, i, max1, max2;
    printf ("Enter number of elements : ");
    scanf ("%d", &n);
    int arr [n];
    printf ("Enter %d integers : ", n);
    for (i = 0; i <n; i++)
        scanf ("%d", &arr [i]);

max1 = max 2 = arr [0];
for (i = 1; i < n; i++) {
    if (arr [i] > max 1) {
        max2 = max1;
        max 1 = arr [i];
    } else if (arr[i] > max2 && arr[i] < max 1) {
        max2 = arr [i];
    }
}
Printf ("Second largest number = %d", max2);
return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.

```
#include <stdio.h>
int main() {
    int n, i, pos = 0, neg = 0, even = 0, odd = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        if (arr[i] > 0) pos++;
        else if (arr[i] < 0) neg++;
        if (arr[i] % 2 == 0) even++;
        else odd++;
    }
    printf("Positive: %d\nNegative: %d\n Even: %d\n odd: %d", pos, neg,
        even, odd);
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. Wap to read a list of integers and store it in a single dimensional array. Write a C program to find the frequency of a particular number in a list of integers.

```
#include <stdio.h>
int main() {
    int n, i, num, count = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d integers: ", n);
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter number to find frequency: ");
    scanf("%d", &num);

    for (i = 0; i < n; i++)
        if (arr[i] == num)
            count++;

    printf("Frequency of %d = %d", num, count);
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. WAP that reads two matrices A(m×n) and B(p×q) ----- Matrix Multiplication (check compatibility).

```
#include <stdio.h>
int main() {
    int a[10][10], b[10][10], c[10][10];
    int m, n, p, q, i, j, k;

    printf("Enter rows and columns of matrix A: ");
    scanf("%d %d", &m, &n);
    printf("Enter rows and columns of matrix B: ");
    scanf("%d %d", &p, &q);

    if (n != p) {
        printf("Matrix multiplication not possible!\n");
        return 0;
    }

    printf("Enter elements of matrix A:\n");
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
for (i=0; i<m; i++)
    for (j=0; j<q; j++) {
        c[i][j] = 0;
        for (k=0; k<n; k++)
            c[i][j] += a[i][k] * b[k][j];
    }

printf("\nResultant Matrix: \n");
for (i=0; i<m; i++) {
    for (j=0; j<q; j++)
        printf("%d\t", c[i][j]);
    printf("\n");
}
return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 6 : Functions

1. Develop a recursive and non-recursive function FACT(num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n=0. Otherwise, FACT(n) = n * FACT(n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for diff^n values of n and r with suitable messages.

⇒ Recursive Factorial + Binomial Coefficient

```
#include <stdio.h>

int FACT(int n) {
    if (n==0)
        return 1;
    else
        return n * FACT(n-1);
}
int main() {
    int n, r, nCr;
    printf("Enter n and r: ");
    scanf("%d %d", &n, &r);

    nCr = FACT(n) / FACT(r) * FACT(n-r));
    printf("Binomial Coefficient (nCr) = %d", nCr);
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

⇒ Non-Recursive factorial

```
#include <stdio.h>

int FACT(int n) {
    int i, fact = 1;
    for (i = 1; i <= n; i++)
        fact *= i;
    return fact;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Factorial of %d = %d", num, FACT(num));
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. Develop a recursive function GCD (num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.

```
#include <stdio.h>

int GCD(int a, int b) {
    if (b == 0)
        return a;
    else
        return GCD (b, a%b);
}

int main () {
    int num1, num2;
    printf ("Enter two numbers: ");
    scanf ("%d %d", &num1, &num2);
    printf ("GCD of %d and %d = %d", num1, num2, GCD
            (num1, num2));
    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. Develop a recursive function FIBO (num) that accepts an integer argument. Write a C program that invokes this function to generate the fibonacci sequence up to num.

```
#include <stdio.h>

int FIBO (int n) {
    if (n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return FIBO (n-1) + FIBO (n-2);
}
int main () {
    int n, i;
    printf ("Enter number of terms: ");
    scanf ("%d", &n);
    printf ("Fibonacci series: ");
    for (i=0; i<n; i++)
        printf ("%d", FIBO(i));
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

4. Develop a C function ISPRIME (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

```
#include <stdio.h>

int ISPRIME (int num) {
    if (num <= 1)
        return 0;
    for (int i=2; i <=num /2; i++)
        if (num % i ==0)
            return 0;
    return 1;
}

int main () {
    int start, end;
    printf ("Enter start and end of range: ");
    scanf ("%d %d", &start, &end);
    printf ("Prime numbers between %d and %d are:\n",
            start, end);
    for (int i= start; i <= end; i++)
        if (ISPRIME (i))
            printf ("%d", i);
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

5. Develop a function REVERSE (str) that accepts a string argument. Write a C program that invokes this function to find the reverse of a given string.

```
#include <stdio.h>
#include <string.h>

void REVERSE (char str[]) {
    int len = strlen (str);
    printf ("Reversed string: ");
    for (int i = len - 1; i >= 0; i--)
        printf ("%c", str[i]);

}

int main () {
    char str[100];
    printf ("Enter a string: ");
    gets (str);
    REVERSE (str);
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

Experiment 7: structures and Union

1. Write a C program that uses functions to perform the following operations: (a) Reading a complex number.
(b) Writing a complex no.
(c) Addition and subtraction of two complex no.

```
#include <stdio.h>

struct complex {
    float real;
    float imag;
};

struct complex readcomplex() {
    struct complex c;
    printf("Enter real part: ");
    scanf("%f", &c.real);
    printf("Enter imaginary part: ");
    scanf("%f", &c.imag);
    return c;
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
void displayComplex(struct Complex c) {
    printf("%.2f + %2.f i\n", c.real, c.imag);
}

struct complex add (struct complex c1, struct Complex c2) {
    struct Complex sum;
    sum.real = c1.real + c2.real;
    Sum.imag = C1.imag + c2.imag;
    return sum;
}

Struct Complex subtract (struct Complex c1, struct complex c2) {
    Struct Complex diff;
    diff.real = c1.real - c2.real;
    diff.imag = c1.imag - c2.imag;
    return diff;
}

int main () {
    struct Complex c1, c2, sum, diff;

    printf ("Enter first complex number: \n");
    c1 = readcomplex ();
    printf ("Enter second complex number: \n");
    c2 = readcomplex ();
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
    sum = add (c1, c2);
    diff = subtract (c1, c2);

    printf ("\nFirst complex number: ");
    displayComplex (c1);
    printf ("Second complex number: ");
    displayComplex (c2);
    printf ("\nSum: ");
    displayComplex (sum);
    printf ("Difference: ");
    displayComplex (diff);

    return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

2. Compute the monthly pay of 100 employees (use structure and calculate DA = 52% of basic pay).

```
#include <stdio.h>

Struct Employee {
    char name[50];
    float basic, da, gross;
};

int main () {
    Struct Employee emp[100];
    int n, i;

    printf("Enter number of employees: ");
    scanf("%d", &n);

    for (i=0; i<n; i++) {
        printf("\nEnter name of employee %d : ", i+1);
        scanf("%s", emp[i].name);
        printf("Enter basic pay : ");
        scanf("%f", &emp[i].basic);
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
    emp[i].da = 0.52 * emp[i].basic;
    emp[i].gross = emp[i].basic + emp[i].da;
}

printf("\nEmployee Name\tGross Salary\n");
for(i=0; i<n; i++){
    printf("%s\t\t%.2f\n", emp[i].name, emp[i].gross);
}

return 0;

}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

3. Create a Book structure containing book_id, title, author name and ~~print~~ price. Write a C program to pass a structure as a function argument and print the book details.

```
#include <stdio.h>

struct Book {
    int book_id;
    char title[50];
    char author[50];
    float price;
};

void displayBook(struct Book b) {
    printf("\n Book ID: %d\n", b.book_id);
    printf("Title: %s\n", b.title);
    printf("Author: %s\n", b.author);
    printf("Price: %.2f\n", b.price);
}
```

Teacher's Signature _____

Expt. No. _____ Page No. ☐ Date ☐☐☐☐☐☐

```
int main () {
    struct Book b1;

    printf ("Enter Book ID: ");
    scanf ("%d", &b1.book_id);
    printf ("Enter Title: ");
    scanf ("%[^\n]", b1.title);
    printf ("Enter Author: ");
    scanf ("%[^\n]", b1.author);
    printf ("Enter Price: ");
    scanf ("%f", &b1.price);

    displayBook (b1);
    return 0;

}
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

Experiment 8: Pointers

1. Declare different types of pointers (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to.

```
#include <stdio.h>
int main() {
    int a = 10;
    float b = 3.5;
    char c = 'x';

    int *p1 = &a;
    float *p2 = &b;
    char *p3 = &c;

    printf("Value of a = %d, via pointer = %d\n", a, *p1);
    printf("Value of b = %.2f, via pointer = %.2f\n", b, *p2);
    printf("Value of c = %c, via pointer = %c\n", c, *p3);
    return 0;

}
```

Teacher's Signature _____

Date

Page No.

Expt. No.

2. Perform pointer arithmetic (increment and decrement) ..... on data access.

```
#include <stdio.h>
int main() {
    int a = 5;
    int *p = &a;

    printf("Address = %p\n", p);
    p++;
    printf("After increment = %p\n", p);
    p--;
    printf("After decrement = %p\n", p);
    return 0;
}
```

Teacher's Signature

Date _____

Expt. No. _____ Page No. _____

3. Write a function that accepts pointers as parameters..... values within the func~

```
#include <stdio.h>

void update(int *x, int *y) {
    *x = *x + 5;
    *y = *y + 10;
}

int main() {
    int a = 10, b = 20;
    update(&a, &b);
    printf("Updated values: a = %d, b = %d\n", a, b);
    return 0;
}
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

Experiment 9: File handling in C

1. Write a program to create a new file and write text into it.

```
#include <stdio.h>
int main() {
    FILE *fp = fopen("sample.txt", "w");
    fprintf(fp, "This is a test file.");
    fclose(fp);
    return 0;
}
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

2. open an existing file and read its content character ... the file.

```
#include <stdio.h>
int main() {
    FILE *fp = fopen ("sample.txt", "r");
    char ch;
    while ((ch = fgetc (fp)) != EOF)
        printf ("%c", ch;
    fclose (fp);
    return 0;

}
```

Teacher's Signature _____

Date

Expt. No. _____ Page No. _____

3 Open a file, read its content line by line, and display each line on the console.

```
#include <stdio.h>
int main() {
    FILE *fp = fopen("sample.txt", "r");
    char line[200];
    while (fgets(line, sizeof(line), fp))
        printf("%s", line);
    fclose(fp);
    return 0;

}
```

Teacher's Signature _____

Date

Expt. No. Page No.

Experiment 10: Dynamic Memory Allocation

1. Write a program to create a simple linked list in C using pointer and structure.

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node *next;
};

int main() {
    struct Node *head = NULL, *newNode, *temp;
    int n, value;
    printf("Enter number of nodes: ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        printf("Enter value: ");
        scanf("%d", &value);
        newNode = (struct Node *) malloc (sizeof(struct Node));
        newNode->data = value;
        newNode->next = NULL;
```

Teacher's Signature

Date

Expt. No. Page No.

```
        if (head == NULL) head = newNode;
        else {
            temp = head;
            while (temp -> next != NULL) temp = temp -> next;
            temp -> next = newNode;
        }
    }

    printf("Linked list: ");
    temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp -> data);
        temp = temp -> next;
    }
    return 0;

}
```

Teacher's Signature

Date _____

Expt. No. _____ Page No. _____

2. Write a program to insert item in middle of the linked list.

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node *next;
};

int main () {
    struct Node *head = NULL, *newNode, *temp;
    int pos, value;

    for (int i = 1; i <= 3; i++) {
        newNode = (struct Node *) malloc (sizeof (struct Node));
        newNode->data = i * 10;
        newNode->next = head;
        head = newNode;
    }

    printf("Enter value to insert: ");
    scanf ("%d", &value);
    printf ("Enter position: ");
scanf ("%d", &pos);
```

Teacher's Signature _____

Date _____

Expt. No. _____ Page No. _____

```
newNode = (struct Node *)malloc(sizeof(struct Node));
newNode->data = value;

temp = head;
for (int i = 1; i < pos - 1; i++) temp = temp->next;

newNode->next = temp->next;
temp->next = newNode;

printf("Updated List: ");
temp = head;
while (temp != NULL) {
    printf("%d -> ", temp->data);
    temp = temp->next;
}

return 0;
}
```

Teacher's Signature _____

Date

Expt. No. _____ Page No. _____

Experiment 11: Bitwise Operator

1. Write a program to apply bitwise OR, AND and NOT operators on bit level.

```
#include <stdio.h>
int main () {
    int a = 5, b = 3;
    printf("a & b = %d \n", a & b);
    printf("a | b = %d \n", a | b);
    printf("~a = %d \n", ~a);
    printf("a << 1 = %d \n", a << 1);
    printf("a >> 1 = %d \n", a >> 1);
    return 0;

}
```

Teacher's Signature _____

Date

Expt. No. _____ Page No. _____

2. Write a program to apply left shift and right shift.

```
#include <stdio.h>

int main () {
    int num, ls, rs;

    printf ("Enter a number: ");
    scanf ("%d", &num);

    ls = num << 1;
    rs = num >> 1;

    printf ("Original number = %d\n", num);
    printf ("After left shift (num << 1) = %d\n", ls);
    printf ("After right shift (num >> 1) = %d\n", rs);

    return 0;

}
```

Teacher's Signature _____

Date

Page No.

Expt. No.

Experiment 12: Preprocessor and Directives in C

1. Write a program to define some constant variable in preprocessor.

```
#include <stdio.h>
#define PI 3.14159

int main() {
    float r, area;
    printf("Enter radius: ");
    scanf("%f", &r);

    area = PI * r * r;

    printf("Area of circle = %.2f\n", area);
    return 0;

}
```

Teacher's Signature

Date

Expt. No.

Page No.

2. Write a program to define a function in directives.

```
#include <stdio.h>
#define SUM (a, b) (a+b)

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);

    printf("Sum = %d\n", SUM(x, y));
    return 0;

}
```

Teacher's Signature

Date

Expt. No. Page No.

Experiment 13: Macros in C

1. WAP to define multiple macro to perform arithmetic functions.

```
#include <stdio.h>

#define ADD(a, b) ((a) + (b))
#define SUB (a,b) ((a) - (b))
# define MUL (a,b) ((a) * (b))
# define DIV (a,b) ((b) != 0 ? (a) / (b) : 0)

int main ( ) {
    int x, y;

    printf ("Enter two numbers: ");
    scanf ("%d %d", &x, &y);

    printf ("Addition = %d\n", ADD(x, y));
    printf ("Subtraction = %d\n", SUB (x, y));
    printf ("Multiplication = %d\n", MUL (x, y));
    printf ("Division = %d\n", DIV (x, y));

    return 0;

}
```

Teacher's Signature

Expt. No. _____ Page No. _____

Experiment 14: Static Library in C

1. Write a program to create a static library for performing arithmetic function.

```
#ifndef ARITH_H
#define ARITH_H
int add(int a, int b);
int sub(int a, int b);
int mul(int a, int b);
int divide(int a, int b);
#endif
```

```
#include "arith.h"
int add(int a, int b){
    return a+b;
}
int sub(int a, int b){
    return a - b;
}
int mul(int a, int b) {
    return a *b;
}
int divide(int a, int b) {
   if (b == 0)
       return 0;
  return a/b;
}
```

Teacher's Signature _____

Date

Expt. No.

Page No.

```
#include main () {
    int a=8, b=4;
    printf ("Addition = %d\n", add (a, b));
    printf ("Subtraction = %.d\n", sub (a, b));
    printf ("Multiplication = %d\n", mul (a, b));
    printf ("Division = %.d\n", divide (a, b));
    return 0;

}
```

Teacher's Signature _____

Date

Expt. No. Page No.

2. Write a program to use static library in other program.

```
#include <stdio.h>
#include "arith.h"

int main() {
    int x=20, y=5;
    printf("Addition = %d\n", add(x, y));
    printf("Subtraction = %d\n", sub(x, y));
    printf("Multiplication = %d\n", mul(x, y));
    printf("Division = %d\n", divide(x, y));
    return 0;

}
```

Teacher's Signature

Date

Expt. No. Page No.

Experiment 15: Shared Library in C

1. WAP to create a shared library for performing arithmetic functions.

```
/* arith.h */
int add (int a, int b);
int sub(int a, int b);
int mul( int a, int b);
int divi( int a, int b);

/* arith.c */
#include "arith.h"
int add(int a, int b) { return a+b; }
int sub (int a, int b) { return a-b; }
int mul (int a, int b) { return a*b; }
int divi (int a, int b) { if (b==0) return 0; return a/b; }

/* main.c */
#include <stdio.h>
#include "arith.h"
int main () {
    int a, b;
    scanf ("%d %d", &a, &b);
    printf ("%d \n", add (a,b));
    printf ("%d \n", sub (a,b));
    printf ("%d \n" mul (a,b));
    printf ("%d \n" divi (a,b));
    return 0;
}
```

Teacher's Signature