LAB MANUAL

For

COMPUTING LABORATORY (CSS 51)

FIRST YEAR

NATIONAL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB EXERCISE #1

Objective(s):

To be familiar with syntax and structure of C- programming. To learn problem solving techniques using C.

Sample Program: Write a Program to calculate and display the volume of a CUBE having its height (h=10cm), width (w=12cm) and depth (8cm).

Code: (Use comments wherever applicable)

```
#include <stdio.h>
int main()
{                                                    //start the program
    int h,w,d,vol;                                   //variables declaration
    h=10;                                            //assign value to variables
    w=12;
    d=8;
    vol=h*w*d;                                       //calculation using mathematical formula
    printf("The Volume of the cube is: %d", vol);    //display the volume
    getch();
    return 0;
}                                                    //end the main program
```

Program List

1. WAP to print your name, address, year and department.
2. WAP to add, subtract, multiply and divide two numbers.
3. WAP to increase or decrease a number using increment and decrement operators.
4. WAP to swap two numbers using a third variable.
5. WAP to swap two numbers without using a third variable.
6. WAP to find greatest number using conditional operator.
7. WAP to print all ASCII values.

LAB EXERCISE #2

Objective(s):

1. To understand the programming using Loop & nested loop Statements (for, while, do-while)
2. To understand the programming knowledge using Decision Statements (if, if-else, if-else-if ladder, switch and GOTO).

Sample Program: Write a program to print positive integers from 1 to 10.

Code: (Using FOR LOOP)

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i;
    for(i=1; i<=10;i++)
    {
        printf(“%d \n”, i);
    }
    getch();
    return 0;
}
```

Program List

1. WAP to see whether a number is Strong or not.
2. WAP to see whether a number is Armstrong or not.
3. WAP to convert a number from decimal to binary.
4. WAP to convert a number from binary to decimal.
5. WAP to convert a number from binary to octal.
6. WAP to find the Fibonacci series of n terms.
7. WAP to find second largest number among N numbers without using array.

LAB EXERCISE #3

Objective(s):

To understand the programming using Loop & nested loop Statements (for, while, do-while)

Sample Program: Write a program to print half pyramid using *.

```
*
* *
* * *
* * * *
* * * * *
```

Code: (Using nested FOR LOOP)

```
#include<stdio.h>
#include<conio.h>
int main()
{
   int i, j, rows;
   printf("Enter number of rows: ");
   scanf("%d", &rows);
   for (i=1; i<=rows; ++i)
   {
      for (j=1; j<=i; ++j)
      {
          printf("* ");
       }
      printf("\n");
   }
   getch();
   return 0;
}
```

Program List

1. WAP to check whether a given number is the product of even position digits and is divisible by odd position digits.
2. WAP to print

```
A
AB
ABC
ABCD
ABCDE
```

3. WAP to print

```
    *
   **
  ***
 ****
*****
```

4. WAP to print

```
*****
*   *
*   *
*   *
*****
```

5. WAP to print

```
  *
 ***
*****
 ***
```

LAB EXERCISE #4

Objective(s):

To understand programming using different dimensions of Array.

Sample Program: Write a program to insert 5 elements into an array and print the elements of the array.

Code: (Use comments wherever applicable)

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i, arr[5];
    printf(“Enter the elements into the array:”);
    for(i=0; i<=4;i++)
    scanf(“%d”,&arr[i]);
    printf(“The elements of the array are:”);
    for(i=0; i<=4;i++)
   {
      printf(“%d \t”, arr[i]);
    }
   getch();
   return 0;
}
```

Program List

1. WAP to insert an element in a sorted array.
2. WAP to delete duplicate elements in an array.
3. WAP to delete an element in a sorted array.
4. WAP to merge and sort two arrays.
5. Write a c program for delete an element at desired position in an array.
6. Write a c program for insert an element at desired position in an array.
7. WAP to perform bubble sorting and selection sort

LAB EXERCISE #5

Objective(s):

To understand programming using different dimensions of Array.

Program List

1. WAP to print upper and lower triangular matrix using user defined function.
2. WAP to print distinct elements of an array using user defined function.
3. WAP to show that the multiplication of the inverse of a matrix and the matrix is identity matrix.
4. WAP to reverse the diagonal elements of a matrix using user defined function
5. WAP to swap the upper and lower triangle elements in a matrix.
6. WAP to print a matrix in a spiral form.

LAB EXERCISE #6

Objective(s)

To understand function programming, its types and function-call.

Sample Program: Write a program to calculate factorial of a number using recursion.

Code:

```
#include<stdio.h>
long factorial(int); //Function declaration
int main()
{
   int num;
   long fact;
   printf(“Enter a number to find factorial: \n”);
   scanf(“%d”, &num);
   if(num<0)
   printf(“Factorial of negative no. is not defined. \n”);
   else
   {
     fact = factorial(num);
     printf(“%d!=%d \n”, num, fact);
   }
  return 0;
  getch();
}
//Function definition
long factorial(int num)
{
   if(num==0)
   return 1;
   else
    return(num*factorial(num-1));
}
```

Program List

1. WAP to find the occurrence of digits in a numbers using user defined function.
2. WAP to perform binary addition of two numbers.
3. WAP to replace a digit in a number by another digit.
4. WAP to find largest and smallest number in an array using user defined function.
5. WAP to perform linear search.

LAB EXERCISE #7

Objective(s):

To understand programming with recursive function call.

Program List

1. WAP to find the LCM and GCD using recursive function.
2. Write a c program to find out sum digits of a number using recursion.
3. Write a c program to find power of a number x^n using function recursion.
4. WAP to print fibonanci series using recursion.

LAB EXERCISE #8

Objective(s):

To understand programming with Pointer.

Sample Program: Write a program to find biggest among three numbers using pointer.

Code:

```
 #include<stdio.h>
#include<conio.h>
int main()
{
    int a,b,c;
    int*ptra=&a,*ptrb=&b,*ptrc=&c;
    printf("enter three values");
    scanf("%d%d%d",ptra,ptrb,ptrc);
    printf("a=%d\n b=%d\n c=%d\n",*ptra,*ptrb,*ptrc);
    if((*ptra>*ptrb && *ptra>*ptrc))
         printf("biggest number=%d",*ptra);
    else if((*ptrb>*ptra && *ptrb>*ptrc))
        printf("biggest number =%d",*ptrb);
    else
        printf("biggest number=%d",*ptrc);
    getch();
    return 0;
}
```

Program List

1. WAP to swap two arrays using pointers and function.
2. WAP to copy an array to another array using pointers.
3. WAP to find the maximum and minimum elements of a matrix using dynamic memory allocation.
4. WAP to find highest and lowest frequency character in a given string.

LAB EXERCISE #9

Objective(s):

To understand programming with Pointer and string

Program List

1. WAP to remove all characters in a string.
2. WAP to find a pattern in the given string.
3. WAP to reverse a string using recursion and check whether the string is palindrome or not.
4. WAP to trim white space from a given string.

LAB EXERCISE #10

Objective(s):

To understand programming with Structure.

Program 1: Write a C program to create, declare and initialize structure

Code:

```
 #include <stdio.h>
/*structure declaration*/
 struct employee{
 char name[30];
 int empId;
 float salary;
 };
 int main()
 {
    /*declare and initialization of structure variable*/
   struct employee emp={"Anil",201,80000.00};
    printf("\n Name: %s" ,emp.name);
    printf("\n Id: %d" ,emp.empId);
    printf("\n Salary: %f\n",emp.salary);
    return 0;
  }
```

Program List

1. Add two complex numbers and display the result.
2. WAP to print the details of a student using structure.
3. WAP to print the details of 5 students using array of structure.
4. WAP to store the details of employees in an array of structure and find the highest salary and the average salary.
5. WAP to add two distances in inches and feet using structure pointers.

LAB EXERCISE #11

Objective(s):

To understand data files and file handling in C.

Sample Program: Write a program to create a file called emp. rec and store information about a person, in terms of his name, age and salary.

Code:

```
#include <stdio.h>
void main()
{
   FILE *fptr;
   char name[20];
   int age;
   float salary;
   /* open for writing */
   fptr = fopen("emp.rec", "w");
   if (fptr == NULL)
   {
     printf("File does not exists \n");
     return;
   }
   printf("Enter the name \n");
   scanf("%s", name);
   fprintf(fptr, "Name = %s\n", name);
   printf("Enter the age\n");
   scanf("%d", &age);
   fprintf(fptr, "Age = %d\n", age);
   printf("Enter the salary\n");
   scanf("%f", &salary);
   fprintf(fptr, "Salary = %.2f\n", salary);
   fclose(fptr);
}
```

Program List

1. WAP to open a file, close a file.
2. WAP to read a file and write to a file.