**VIRTUAL LAB**

\_­­­­

AssignmentVirtual Lab Assignment-1

Virtual Lab Assignment-1

AssignmentVirtual Lab Assignment-2

Virtual Lab Assignment-2

AssignmentVirtual Lab Assignment 3

Virtual Lab Assignment 3

AssignmentVirtual Lab Assignment-4

Virtual Lab Assignment-4

AssignmentVirtual Lab Assignment-5

Virtual Lab Assignment-5

AssignmentVirtual Lab Assignment -6

Virtual Lab Assignment -6

AssignmentVirtual Lab Assignment 7

Virtual Lab Assignment 7

AssignmentVirtual Lab Assignment 8

Virtual Lab Assignment 8

AssignmentVirtual Lab Assignment 9

Virtual Lab Assignment 9

AssignmentVirtual Lab Assignment 10

Virtual Lab Assignment 10

Live Session

LAB MANUAL

Visible

Add ExistingCreate New

LAB MANUAL

***SUBJECT TITLE***

***Programming for Problem Solving***

***SUBJECT CODE V20UDS103***

***OBJECTIVES***

1*.To introduce students to the basic knowledge of Programming development*

*2. To impart writing skill of programming to the students and solving problems.*

*3. To impart the concepts like looping, array, functions, pointers, file, structure in        C Programming.*

***COURSE OUTCOME***

*1. Understand the logic for a given problem.*

*2. Write the Process Steps (algorithm ) of a given problem.*

*3. Apply all the concepts that have been covered in the theory course*

*4. Evaluate methodology of a given problem.*

*5. Recognize and understand the syntax and construction of programming code.*

*6. Know the steps involved in compiling, linking and debugging code.*

*7. Know the alternative ways of providing solution to a given problem*

***INTRODUCTION ABOUT VIRTUAL LAB***

*Virtual  Lab Content  is prepared by Course Coordinator of concern subject  to help the students with their practical understanding and development of programming skills, and may be used as a base reference during the Practical Assignments.  The model lab programs and List of Exercise Assignment prepared  by staff members  will be upload in LMS .Students have to submit Lab Exercise through LMS as Assignment Sections as Separate Folder of concern subject . The course coordinator of concern subject  can be evaluated  after students  submit all program assignments for  end semester sessional examination.. The lab Program  reporting style in the prescribed format (Appendix-I) and List of Lab Exercises as Assignments prescribed format (Appendix-II)*

**APPENDIX-I**

1. **ARITHMETIC OPERATIONS**

**Aim:**

          Write a C program to perform all the basic arithmetic operations.

**Algorithm:**

* Start the program.
* Declare the variables A, B, C of integer type.
* Read the values for the variables A and B.
* Find the sum of A and B and store it in C. ie C = A + B.
* Print the C value.
* Find the difference between A and B, ie C = A – B, print the C value
* Find the product of A and B by the expression C = A \* B, then print the C value.
* Divide the A by B and store the quotient in C as C = A / B, then print the C value.
* Stop the program.

|  |
| --- |
| Start |

|  |
| --- |
| C = A + B; |

|  |
| --- |
| C = A - B; |

|  |
| --- |
| C = A  B; |

|  |
| --- |
| C = A / B; |

|  |
| --- |
| Stop |

**Program:**

#include <stdio.h>

int main() {

int a, b, c;

float x;

printf("\nEnter 2 Nos : ");

scanf("%d%d", &a, &b);

printf("\nTotal : %d\nDifference : %d\nMul : %d\nDiv : %0.2f\nMod : %d",

(c = a + b), (c = a - b), (c = a \* b), (x = (float) a / b), (c = a % b));

return 0;

}

**Output:**

Enter 2 Nos : 12

23

Total: 35

Difference: -11

Mul: 276

Div. : 0.52

Mod: 12

**Result:**

              Thus, the C program to perform the basic arithmetic operations is written and executed successfully.

1. **FIBONACCI SERIES**

**Aim:**

          Write a C program to generate the Fibonacci series.

**Algorithm:**

* Start the program.
* Read the number of terms you want to print.
* If the number of terms is Zero, then print 0.
* else

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

{

f=f+a;

a=b;

b=f;

          }

* Print the f value in every incrimination.
* Stop the program.

|  |
| --- |
| Start |

|  |
| --- |
| f=f+a;  a=b;  b=f; |

|  |
| --- |
| Stop |

**Program:**

#include<stdio.h>

int main() {

int n1 = 0, n2 = 1, n3, i, number;

printf("Enter the number of elements:");

scanf("%d", &number);

printf("\n%d %d", n1, n2);

for(i = 2; i < number; ++i) {

n3 = n1 + n2;

printf(" %d", n3);

n1 = n2;

n2 = n3;

}

return 0;

}

**Output:**

Enter the no of terms: 5

Fibonacci sequence up to 5 terms: 0 1 1 2 3

**Result:**

              Thus, the Fibonacci series is generated successfully by using the C coding.

1. **ONE DIMENSIONAL ARRAY – Number**

**Aim:**

          Write a C program to store any five numbers in an one dimensional array.

**Algorithm:**

* Start the program.
* Read any five numbers for the array a..
* Store the five numbers in their locations by using for loop.
* Display the numbers as follows,

for(i=0;i<5;i++)

{

Printf(“%d”,a[i]);

}

* Stop the program.

**Flow chart**

|  |
| --- |
| Start |

|  |
| --- |
| Stop |

**Program:**

#include<stdio.h>

int main() {

int arr[50], size, i;

printf("Enter the number of elements in the array: ");

scanf("%d", &size);

printf("Enter %d elements: ", size);

for (i = 0; i < size; i++)

scanf("%d", &arr[i]);

printf("\nElements of the array are:\n");

for (i = 0; i < size; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}

**Output:**

Enter the number of elements in the array:5

Enter 5 Elements: 34

56

67

87

98

Elements of the array are:

34 56 67 87 98

**Result:**

              Thus the one dimensional integer array concept is tested successfully.

1. **STRING COMPARISON using strcmp**

**Aim:**

          Write a C program to compare the two given strings.

**Algorithm:**

* Start the program.
* Read the two strings to the variables say S1 and S2.
* Compare the two strings by the function strcmp(), and assign the value to variable n.
* If n == 0 then, print,” The given strings are same”.
* If n != 0 then, print “ the given strings are different”.
* Stop the program.

**Program:**

#include <stdio.h>

int compare(char[], char[]);

int main() {

char str1[20], str2[20];

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

printf("Strings are %s\n", compare(str1, str2) ? "not the same" : "the same");

return 0;

}

int compare(char a[], char b[]) {

int i = 0;

while (a[i] != '\0' && b[i] != '\0') {

if (a[i] != b[i])

return 1;

i++;

}

return a[i] != b[i];

}

**Output 1:**

Enter the first string: ram

Enter the second string: ram

The given strings same.

**Output 2:**

Enter the first string: raj

Enter the second string: ram

The strings are different.

**Result:**

  Thus, the given string is reversed by using the string handling function.

1. **FACTORIAL USING FUNCTION**

**Aim:**

          Write a C program to find the factorial of a given number by using function.

**Algorithm:**

main() program

* Start the program.
* Read the given number.
* Call the function fact ();
* Assign the value of the function fact () to f;
* Print f.
* Stop  the program

fact() function

* start the program.
* if( (n = = 0) || ( n = = 1 ), return 1.
* else return (n\*fact(n-1)).
* stop the program.

getch();

}

int fact(int n)

{

if(n==0 || n==1)

{

return(1);

}

else

{‘

return(n\*fact(n-1));

}

}

**coding**

#include<stdio.h>

long factorial(int n)

{

if (n == 0)

return 1;

else

return(n \* factorial(n-1));

}

void main()

{

int number;

long fact;

printf("Enter a number: ");

scanf("%d", &number);

fact = factorial(number);

printf("Factorial of %d is %ld\n", number, fact);

return 0;

}

**Output:**

Enter a number: 5

Factorial of 5 is 120

**Result:**

              Thus the factorial of the given number is found successfully by using function concept in C program.

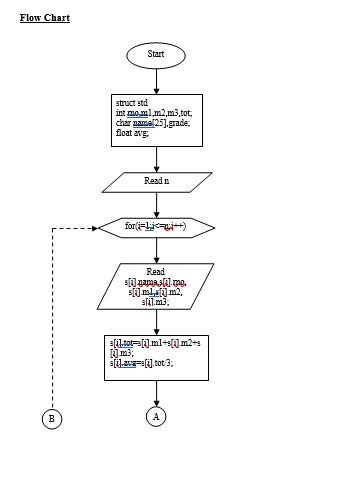
1. **STRUCTURE**

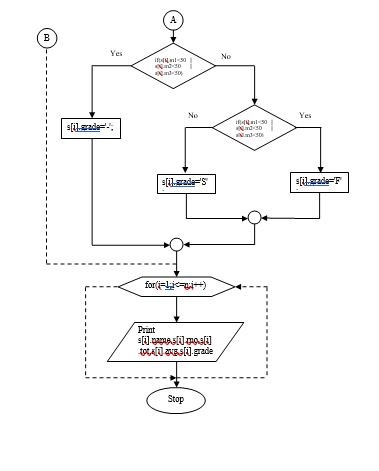
**Aim:**

          Write a C program to create students mark list by using the structure concept.

**Algorithm:**

* Start the program.
* Declare the structure std with its members.
* Declare the structure variable s [].
* Get the number of students.
* Read the name, roll no and three subject marks of each student(s).
* Calculate the total and average of the student.
* Find the grade of the student based on the average.
* Print the name, roll no, total, average and grade of the student(s).
* Stop the program.

****

****

**Program:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

struct std

{

int rno,m1,m2,m3,tot;

char name[25],grade;

float avg;

}s[100];

void main()

{

int i,n;

clrscr();

printf("\n Enter the number of students : ");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

printf("\n Enter student %d's  name :",i);

scanf("%s",s[i].name);

printf("\n Enter student %d's Roll No: ",i);

scanf("%d",&s[i].rno);

printf("\n Enter student %d's 3 marks : ",i);

scanf("%d%d%d",&s[i].m1,&s[i].m2,&s[i].m3);

s[i].tot=s[i].m1+s[i].m2+s[i].m3;

s[i].avg=s[i].tot/3;

if(s[i].m1<50 || s[i].m2<50 || s[i].m3<50)

s[i].grade='-';

else

if(s[i].avg >= 75)

s[i].grade='F';

else if(s[i].avg >= 50)

s[i].grade='S';

}

printf("\n Name Roll No  Total     Avg   Grade ");

for(i=1;i<=n;i++)

{

printf("\n %s\t%d\t%d\t%4.2f\t%c",s[i].name,s[i].rno,s[i].tot,s[i].avg,s[i].grade);

}

getch();}

**Output:**

 Enter the number of students : 2

 Enter student 1's  name :raj

 Enter student 1's Roll No: 2

 Enter student 1's 3 marks : 70 80 90

 Enter student 2's  name :kumar

 Enter student 2's Roll No: 2

 Enter student 2's 3 marks : 60 70 80

 Name    Roll No   Total     Avg     Grade

 raj             2           240      80.00      F

 kumar       2           210      70.00      S

**Result:**

              Thus the student mark list is prepared successfully by using the structure in C programming.