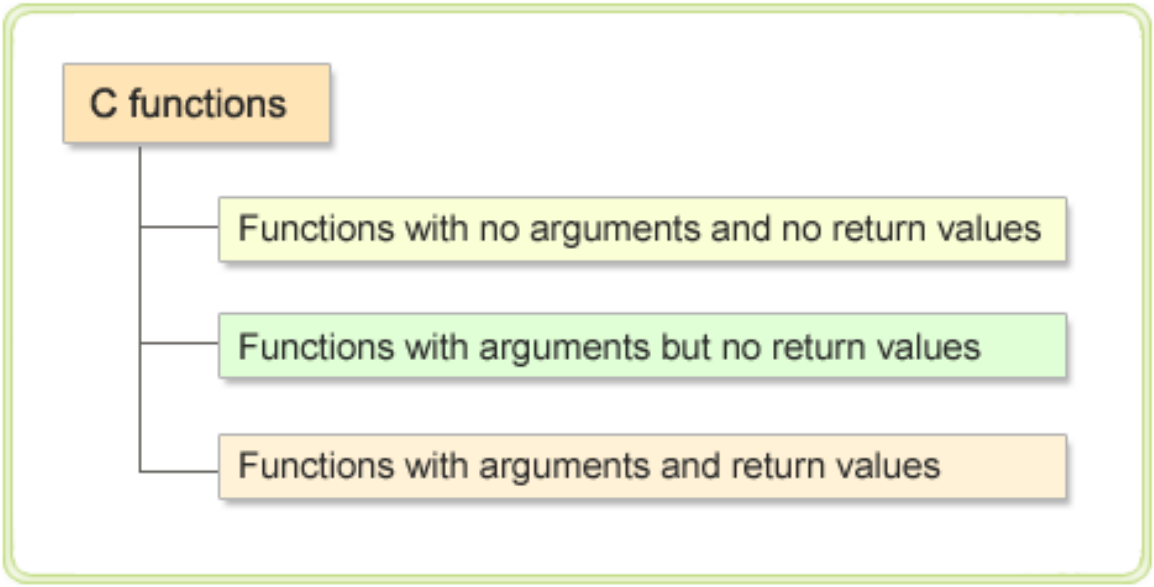
ME 172

C Programming Language Sessional

Lecture 8

Functions

- Functions are passages of code that have been given a name.
- C functions can be classified into two categories
  - Library functions
  - User-defined functions



Function with no arguments and no return values

```
#include<stdio.h>
void printline();          // Function Prototype
void main(void)
{
    printline();
}
void printline()
{
    int i;
    for(i=1;i<=50;i++)
            printf("%c",'-');
    printf("\n");
}
```

Calling Function

Called Function

Function with arguments and no return values

```
#include<stdio.h>
void printline(char ch);          // Function Prototype
void main()
{
    printline('Z');
}
void printline(char ch)
{
    int i;
    for(i=1;i<=35;i++)
            printf("%c",ch);
    printf("\n");
}
```

Function Parameter/ argument

Parameters/Arguments & Prototypes

- Function Parameters/ Arguments
  - A function parameter is a value that can be passed in to a function (inside the round brackets) to modify the behavior of the function.
  - A function parameter is a local variable that has been "passed in" from the calling function.
- Function Prototypes
  - A prototype is a synopsis of the correct use of a function
  - It is essentially the first line of the function (the declaration), with a semicolon on the end
  - It allows the compiler to check your correct use of a function
  - For example:

```
float average3(int n1, int n2, int n3);
```

or

```
float average3(int, int, int)
```

Function with arguments and return values

```
#include<stdio.h>
float value (float p, float r, int n);
void main()

{
    float principle,inrate,amount;
    int period;
    printf("Enter the principle amount interset");
    printf("rate and period");
    scanf("%f %f %d",&principle,&inrate,&period);
    amount = value(principle,inrate,period);
    printf("\nThe full amount is %f\n",amount);
}
```

Function with arguments and return values (Contd..)

```
float value (float p, float r, int n)
 {
     int year;
     float sum;
     sum = p; year =1;
     sum = p;
     year =1;
     while(year <=n)
     {
            sum = sum *(1+r);
            year = year +1;
     }
    return(sum);   /*The value to the right of the keyword return must be of
    the     same type as the function itself */
 }
```

Must match

Example -4

```
//REVERSE THE GIVEN NUMBER //
#include<stdio.h>
 void  reverse(int n);
void main()
{
  int num;
  printf("\nEnter your number:=");
  scanf("%d",&num);
  printf("\nThe Reverse number is:=");
  reverse(num);
  printf("\n");
}
```

```
void  reverse(int n)
{
   int i;
    do
    {
     i=n%10;
     printf("%d",i);
     n=n/10;
    }while(n!=0);
    }
```

Example 5

Write and test a function that accepts three positive integer numbers and returns the largest number.

Use a random number generator to get the three integer numbers.

Notes:

The **rand( )** function generates a sequence of random numbers.
Each time it is called, an integer between zero and **RAND_MAX** is returned.
RAND_MAX will be at least 32,767 (depending on machine & compiler).
Header file : #include <stdlib.h>

```
int rand(void);
```

Solution of Example 5

```
#include<stdio.h>
#include<stdlib.h>
int largest(int, int, int);
void main(void){
        int   a,b,c,larg;
        a = rand() % 100;
        b = rand() % 100;
        c = rand() % 100;
printf(" random numbers are : %d %d %d\n\n", a,b,c);

        larg = largest(a,b,c);
        printf("The largest number is = %d",larg);
}
```

Solution of Example 5

```
int largest(int a, int b, int c)

        {
        if (a>b)
                if (a>c)        return a;
                else            return c;
        else
                if (b>c)        return b;
                else            return c;
        }
```

Example 6

Write and test a function to swap two number.
Use pass by reference protocol.

Solution of Exercise 6

```
#include<stdio.h>

int swap(int &a,int &b)
/*When using a reference parameter,
the address of an argument is passed to the function
and the function operates on the argument, not a
    copy.*/
{
    int c;
    c=a;
    a=b;
    b=c;
    return 0;
}
```

Solution of Exercise 6

```
int main()
{
    int x,y;
    printf("\nEnter value for x:=");
   scanf("%d",&x);
   printf("\nEnter value for y:=");
   scanf("%d",&y);
   printf("\nBefore swap:x=%d and y=%d",x,y);
    printf("\nx=%d y= %d\n",&x,&y);
    swap(x,y);
    printf("\nx=%d y= %d\n",&x,&y);
   printf("\nAfter swap :x=%d and y=%d",x,y);
    return 0;
}
```

Global Variable & Constants

// Program to convert temperature from degree to radian

```
#include<stdio.h>
#define pi 3.1416
// to define a constant
double m;
 /* gloabal varaible (A "global variable" is a
    variable that can be accessed by more
    than one function)*/
double radian(double n1)
{
    m = 180;
    n1 = (pi/180)*n1;
   return n1;
}
```

```
void main()
{
    double degree;
    double rad;

    printf("Please enter the temp at
    degree:");
    scanf("%lf", &degree);

    rad = radian(degree);

    printf("The radian value %5.4lf\n", rad);
}
```

Recursion

- When a function calls itself in its body it is called a recursive function.

// Factorial with recursive funtion & without recursive funtion //

```
#include<stdio.h>

long int fact1(long int n)
{
 if(n==0)
 return 1;
 else
 return (n*fact1(n-1));
}
```

Recursion

```
int main()
{
 long int num,j,fact=1;
 printf("\nEnter a number:=");
 scanf("%ld",&num);

 printf("\nWithout using recursive funtion\n");
printf("************************************");
 if(num<0)
 printf("\nGiven number has no factorial");
 else if(num==0)
 printf("\nFactorial of 0 is:=1");
 else
 {
  for(j=1;j<=num;j++)
  fact=fact*j;
 }
 printf("\nFactorial of %ld is:=%ld\n",num,fact);
 printf("\nUsing recursive funtion\n");
printf("************************************");
 if(num<0)
 printf("\nGiven number has no factorial");
 else
 printf("\nFactorial of %ld
     is:=%ld\n",num,fact1(num));
return 0;
}
```

Try Yourself

Write and test a function that accepts a positive integer number n as argument and returns the factorial of n.

Exercise 1

Write a function named “area” to calculate the area of a triangle. The function accepts the six coordinate points (x1, y1, x2, y2, x3, y3).

The function “area” will call another function named “arms” to calculate the lengths (a, b, c) of the three arms of the triangle using corresponding coordinates. Then using the values a, b, and c function “area” will calculate the area.

Exercise 1

```
float area(float, float, float, float, float, float);
float distance(float, float, float, float);
void main(void){
    float x1,x2,x3,y1,y2,y3;
    float tri_area;
    printf("Enter the coordinates for point 1: ");
    scanf("%f%f",&x1,&y1);
    printf("Enter the coordinates for point 2: ");
    scanf("%f%f",&x2,&y2);
    printf("Enter the coordinates for point 3: ");
    scanf("%f%f",&x3,&y3);
    tri_area= area(x1,y1,x2,y2,x3,y3);
    printf("The area of the triangle is: %5.2f\n",tri_area);
}
```

Exercise 1

```
float area(float x1, float y1, float x2, float y2, float x3, float y3)
{
    float s,area;
    float d12, d23, d31;

    d12 = distance(x1, y1, x2, y2);
    d23 = distance(x2, y2, x3, y3);
    d31 = distance(x3, y3, x1, y1);

    s = (d12 + d23 + d31)/2.0;
    area = sqrt(s*(s-d12)*(s-d23)*(s-d31));
    return area;
}
```

Exercise 1

```
float distance(float xx1, float yy1, float xx2, float yy2)
{
  float dd;
  dd = sqrt((xx1-xx2)*(xx1-xx2)+(yy1-yy2)*(yy1-yy2));
  return dd;
}
```

That’s all about today….