COMPUTER PROGRAMMING FUNCTIONS

http://www.tutorialspoint.com/computer_programming/computer_programming_functions.htm

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing. You already have seen various functions like **printf** and **main**. These are called built-in functions provided by the language itself, but we can write our own functions as well and this tutorial will teach you how to write and use those functions in C programming language.

Good thing about functions is that they are famous with several names. Different programming languages name them differently like functions, methods, sub-routines, procedures, etc. So when you come across any such terminology, then just imagine about the same concept, which we are going to discuss in this tutorial.

Let's start with a program where we will define two arrays of numbers and then from each array, we will find the biggest number. As we already have seen following are the steps to find out maximum number from a given set of numbers:

```
1.      Get a list of numbers $L_1$, $L_2$, $L_3$....$L_N$
2.      Assume $L_1$ is the largest,  Set max = $L_1$
3. Take next number $L_i$ from the list and do the following
4.      If max is less than $L_i$
5.         Set max = $L_i$
6.      If $L_i$ is last number from  the list then
7.         Print value stored in max and come out
8.      Else prepeat same process starting from step 3
```

Let's translate above program in C programming language:

```
#include <stdio.h>

main()
{
    int set1[5] = {10, 20, 30, 40, 50};
    int set2[5] = {101, 201, 301, 401, 501};
    int i, max;

    /* Process first set of numbers available in set1[] */
    max = set1[0];
    i = 1;
    while( i < 5 )
    {
        if( max <  set1[i] )
        {
            max = set1[i];
        }
        i = i + 1;
    }
    printf("Max in first set = %d\n", max );

    /* Now process second set of numbers available in set2[] */
    max = set2[0];
    i = 1;
    while( i < 5 )
    {
        if( max <  set2[i] )
        {
            max = set2[i];
        }
        i = i + 1;
    }
    printf("Max in second set = %d\n", max );
}
```

When the above code is compiled and executed, it produces the following result:

```
Max in first set = 50
Max in second set = 501
```

If you are clear about the above example, then it will become easy to understand why do we need a function. Here in above example, I took only two sets of numbers set1, and set2 but consider a situation we have 10 or more similar sets of numbers to find out maximum numbers from each set. In such situation, we will have to repeat same processing 10 or more times and ultimately program will become too large with repeated code. To handle such situation, we write our functions where we try to keep source code which will be used again and again in our programming.

Now, let's see how to define a function in C programming language and then subsequent section will explain how to use that function:

Defining a Function:

The general form of a function definition in C programming language is as follows:

```
return_type function_name( parameter list )
{
   body of the function

   return [expression];
}
```

A function definition in C programming language consists of a *function header* and a *function body*. Here are all the parts of a function:

- **Return Type** : A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword **void**.

- **Function Name** : This is the actual name of the function. The function name and the parameter list together constitute the function signature.

- **Parameter List** : A parameter is like a placeholder. When a function is invoked, you pass a value as a parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

- **Function Body** : The function body contains a collection of statements that define what the function does.

Calling a Function:

While creating a C function, you give a definition of what the function has to do. To use a function, you will have to call that function to perform the defined task.

Now, let's write above example with the help of a function:

```
#include <stdio.h>

int getMax( int set[] )
{
   int i, max;

    max = set[0];
    i = 1;
    while( i < 5 )
    {
        if( max <  set[i] )
        {
            max = set[i];
        }
        i = i + 1;
    }

    return max;
}

main()
{
    int set1[5] = {10, 20, 30, 40, 50};
    int set2[5] = {101, 201, 301, 401, 501};
    int max;

    /* Process first set of numbers available in set1[] */
    max = getMax(set1);
    printf("Max in first set = %d\n", max );

    /* Now process second set of numbers available in set2[] */
    max = getMax(set2);
    printf("Max in second set = %d\n", max );
}
```

When the above code is compiled and executed, it produces the following result:

```
Max in first set = 50
Max in second set = 501
```

For now, its enough for you to know about what are functions and how do they work. If you understood this concept then you can proceed for a detailed tutorial to drill it down further.

Functions in Java

If you are clear about functions in C programming, then its easy to understand them in Java as well. Java programming names them as **methods** , but rest of the concepts remain more or less same as we discussed in C programming.

Following is the equivalent program written in Java programming language. You can try to execute the following program to see the output:

```
public class DemoJava
{
   public static void main(String []args)
   {
      int[] set1 = {10, 20, 30, 40, 50};
      int[] set2 = {101, 201, 301, 401, 501};
      int max;

      /* Process first set of numbers available in set1[] */
      max = getMax(set1);
      System.out.format("Max in first set = %d\n", max );

      /* Now process second set of numbers available in set2[] */
      max = getMax(set2);
      System.out.format("Max in second set = %d\n", max );
   }

   public static int getMax( int set[] )
   {
       int i, max;

       max = set[0];
       i = 1;
       while( i < 5 )
       {
           if( max <  set[i] )
           {
               max = set[i];
           }
           i = i + 1;
       }

       return max;
   }
}
```

When above program is executed, it produces the following result:

```
Max in first set = 50
Max in second set = 501
```

Functions in Python

Once again, if you already understood the concept of functions in C and Java programming, then Python is not much different in defining and calling functions. Following is basic syntax of defining a function in Python:

```
def function_name( parameter list ):
   body of the function

   return [expression]
```

So using this syntax of function in Python, above example can be written as follows:

```
def getMax( set ):
    max = set[0]
    i = 1
    while( i < 5 ):
        if( max <  set[i] ):
            max = set[i]
        i = i + 1
    return max


set1 = [10, 20, 30, 40, 50]
set2 = [101, 201, 301, 401, 501]

# Process first set of numbers available in set1[]
max = getMax(set1)
print "Max in first set = ", max

# Now process second set of numbers available in set2[]
max = getMax(set2)
print "Max in second set = ", max
```

When the above code is executed, it produces the following result:

```
Max in first set =  50
Max in second set =  501
```