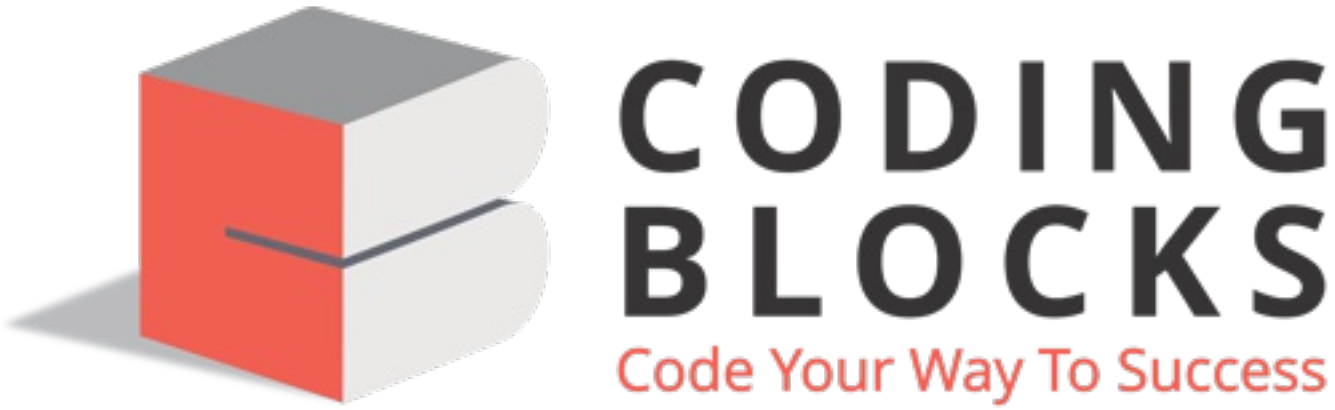
Functions



What is a function?

- A function groups a number of program instructions into one unit and gives it a name. This can then be invoked from other parts of the program.
- It adds conceptual organization and increases reusability of the code.

Lets move some programs to functions!

- Write function to print a table of Fahrenheit to Celsius table from 0 to 300.
- Lets convert the above to take limits from the user.
- Write a function to check if a number is prime or not.
- Use above in another function to generate all prime numbers from 2 to N, where N is given by user.

So a function looks like :

```
type name ( parameter1, parameter2, â€¦) {
  statements }
```

- **type** is the type of the value returned by the function.
- **name** is the identifier by which the function can be called.
- **parameters** (as many as needed): Each parameter consists of a type followed by an identifier, with each parameter being separated from the next by a comma. Each parameter looks very much like a regular variable declaration (for example: int x), and in fact acts within the function as a regular variable which is local to the function. The purpose of parameters is to allow passing arguments to the function from the location where it is called from.
- **statements** is the function's body. It is a block of statements surrounded by braces { } that specify what the function actually does.

Time to try?

- Write a function to return factorial of a number ?
- Write a function which uses above to calculate NCR ?

More about functions

- A function generally has three parts
- Declaration

- Definition
- Invoking
    - Declaration is optional if function is defined above main()
    - A function needs to be defined or declared before it can be called i.e. if you are calling a function A() in function B() then A should be declared or defined above B.

do some more problems?

- Write a function which takes an array and its length as argument and returns sum of its elements.
- Write a function which takes an array as argument and sorts them using selection Sort.
- There are two sorted arrays. First one is of size m+n containing only m elements. Another one is of size n and contains n elements. Write a function to merge these two arrays into the first array of size m+n.

Home Work

- Revise Binary Search
- Write a function which takes a number X and a array and prints all prints all pairs which sum to X.
- Write a function which takes two sorted arrays, and their lengths as arguments and returns combined median of them without using the third array.
- Implement Insertion Sort.

Inline Functions

- If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.
- To inline a function, place the keyword inline before the function name and define the function before any calls are made to the function
- The compiler can ignore the inline qualifier in case defined function is more than a line.

Default Parameter

- A default parameter is a function parameter that has a default value provided to it.
- If the user does not supply a value for this parameter, the default value will be used. If the user does supply a value for the default parameter, the user-supplied value is used

Scope and Duration of Variables

- When discussing variables, itâ€™s useful to separate out the concepts of **scope** and **duration**.
- A variableâ€™s scope determines where a variable is accessible
- A variableâ€™s duration determines where it is created and destroyed

Local Variables

- Variables defined inside a block are called
  local variables.
- Local variables have automatic duration, which means they are created when the block they are part of is entered, and destroyed when the block they are part of is exited.
- Local variables have block scope (also called local scope), which means they are visible only within the block that they are defined in.

Global Variables

- Variables declared outside of a block are
  called global variables
- Global variables have static duration, which means they are created when the program starts and are destroyed when it ends
- Global variables have global scope (also called â€œglobal namespace scopeâ€ or â€œfile
  scopeâ€), which means they are visible until the end of the file in which they are declared
- By convention, global variables are declared at the top of a file, below the includes, but above any code.

Local Static Variables

- Using the static keyword on local variables changes them from automatic duration to static duration.
- A static duration variable (also called a â€œstatic variableâ€) is one that retains its value even after the scope in which it has been

created has been exited!
- Static duration variables are only created (and initialized) once, and then they are persisted throughout the life of the program.

Function Overloading

- You can have multiple definitions for the same function name in the same scope
- The definition of the function must differ from each other by the types and/or the number of arguments in the argument list.
- You can not overload function declarations that differ only by return type.