IEEE SCT SB

IEEE

Code in 10 days

Day 6

dotSlash
Workshop series

Topics for Today

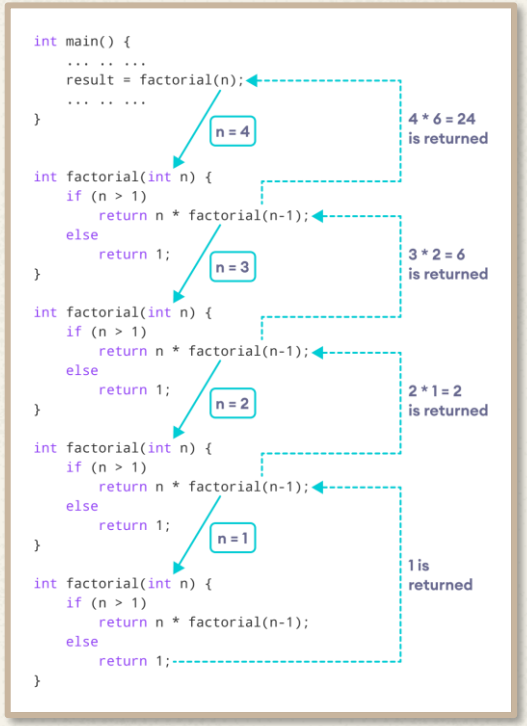
- Recursive Functions
- Scope Rules
- C++ Header Files
- Built-in Functions

Recursive Functions

- A function that calls itself in its own body is called a recursive function.

e.g.

```
int factorial (int n)
{
    if (n>1)
      return n * factorial(n-1);
    else
      return 1;
}
```



Program 1

//Recursive program to find the Fibonacci series of n where n>2

```
#include<iostream>
using namespace std;
int fib(int n)
{
        if (n==0)
           return 0;
        if (n==1 || n==2)
           return 1;
        else
          return (fib(n-1) + fib(n-2));
}
```

```
int main()
{
        int n=5;
        cout<<“Fibonacci series for
        5 numbers: “;
        for(int i=0; i<n; i++)
        {
           cout<<fib(i)<<“ “;
        }
        return 0;
}
```

Scope Rules

- The parts of a program where a particular piece of code or data value can be accessed is known as variable scope.
- The scope of variables is determined by the place of their declaration.
- There are two types:
  - Global Variables
  - Local Variables
- Similarly, function prototypes can also be of two types:
  - Global Prototypes
  - Local Prototypes

Lifetime

- The time interval for which a particular variable or data value lives in the memory.
- A global variable’s lifetime is the program’s runtime
- Whereas, for local variables, they are created the moment the method is activated and are destroyed when the activation of the method terminates.

Global Variables

- A global variable is available to every function and block of code defined in the file.
- The declaration of such a variable appears outside all of the functions.
- It comes into existence when the program execution starts and is destroyed when the program terminates.

Local Variables

- A local variable is defined within a function.
- It comes into existence when the function is entered and is destroyed upon exit.
- It is defined and initialised each time a function call occurs.

Examples

```
// Using local variables
#include<iostream>
using namespace std;

void func()
{
        int age=19;
        cout<<age;
}

int main()
{
        int age =18;
        cout<<"Age is: "<<age;
        cout<<"\nAge is: ";
        func();

        return 0;
}
```

```
// Using global variables
#include<iostream>
using namespace std;

int global = 5;          // global variable

void display()
{
        cout<<global<<endl;
}

int main()
{
        display();
        global = 10;
        display();
}
```

Static Local Variables

- It is defined and initialised the first time a function is called occurs.
- It holds its value throughout the whole program run.
- Its scope is still within the function.

Example

```
// Static variables in a Function
#include <iostream>
#include <string>
using namespace std;

void demo()
{
        static int count = 0;       // static variable
        cout << count << " ";
        count++;     /* value is updated and will be
                        carried to next function calls */

}

int main()
{
        for (int i=0; i<5; i++)
            demo();
        return 0;
}
```

Global Prototypes

- If a function’s prototype appears outside all other functions in a program, it is said to be a global prototype.
- Such a function can be accessed from any of the functions within that file, i.e. they are globally available to all the functions in that file

Local Prototypes

- If a function's(B's) prototype appears inside another function(A), the function(B) is said to be locally available to that function(A).

- Their scope is local to the function that contains their declaration.

Header Files in C++

C++ has around 49 header files.

1. #include<iostream> (Standard input-output header)
   Used to perform input and output operations like cout and cin.

2. #include<cstring> (String header)
   Perform string manipulation operations like strlen and strcpy.

3. #include<cstdio> (Standard input-output header)
   Used to perform input and output operations like gets(), puts(), getc(), putc(), etc.

Header Files in C++

4. #include<math> (Math header)
   Perform mathematical operations like sqrt() and pow().

5. #include<ctype>(Character type header)
   Perform character type functions like isalpha() and isdigit.

Character Functions

Function	Description
int isalnum(int ch)	• The isalnum() function returns nonzero if its argument is a letter or a digit. • If the character is not an alphanumeric, isalnum() returns zero.
int isalpha(int ch)	• This function returns nonzero if ch is an alphabet, otherwise it returns zero.
int isdigit(int ch)	• Returns nonzero if ch is a digit (i.e., 0-9). Otherwise it returns zero.
int islower(int ch)	• Returns nonzero if ch is a lowercase letter ; otherwise it returns zero.

Function	Description
int isupper(int ch)	• This function returns nonzero if ch is uppercase. • Otherwise it returns zero.
int toupper(int ch)	• Returns the uppercase equivalent of ch if ch is a letter. • Otherwise, ch is returned unchanged.
int tolower(int ch)	• Returns the lowercase equivalent of ch if ch is a letter. • Otherwise ch is returned unchanged.

String Functions

Function	Description
char *strcat(char *str1, const char *str2)	• This functions concatenates a copy of str2 to str1 and terminates str1 with a null. • str1 should be large enough to hold both its original contents and those of str2.
int strcmp(const char *str1, const char *str2)	• This functions alphabetically compares two strings. • It returns a -ve value if str1 is less than str2. • 0 if str1 is equal to str2; and >0 (+ve value) if str1 is greater than str2.
char *strcpy(char *str1, const char *str2)	• Copies the contents of str2 into str1.
int strlen(char *str)	• Returns the length of the null-terminated string pointed to by str. The null is not counted.

Mathematical Functions

Function	Description
abs(x)	Returns the absolute value of x
cbrt(x)	Returns the cube root of x
ceil(x)	Returns the value of x rounded up to its nearest integer
cos(x)	Returns the cosine of x
cosh(x)	Returns the hyperbolic cosine of x
exp(x)	Returns the value of E^x
floor(x)	Returns the value of x rounded down to its nearest integer
pow(x, y)	Returns the value of x to the power of y
sin(x)	Returns the sine of x (x is in radians)
sinh(x)	Returns the hyperbolic sine of a double value
tan(x)	Returns the tangent of an angle
tanh(x)	Returns the hyperbolic tangent of a double value
sqrt(x)	Returns the square root of x

Program 2

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;

int main()
{

char str[30];
int len= 0;
gets(str);
len= strlen(str);
cout<<"Length: "<<len;
cout<<endl;
cout << "Original String:\n"<< str<< endl;
cout<<"String in upper:\n";
for (int x=0; x<strlen(str); x++)
        putchar(toupper(str[x]));
return 0;
}
```

Program 3

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
   char str1[100], str2[50];
   cout<<"Enter the First String: ";
   gets(str1);
   cout<<"Enter the Second String: ";
   gets(str2);
   strcat(str1, str2);
   cout<<"\nString after Concatenation:
"<<str1;
   cout<<endl;
   return 0;
}
```

Program 4

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char str1[50], str2[50];
    int len1, len2;
    cout<<"Enter the First String: ";
    cin>>str1;
    cout<<"Enter the Second String: ";
    cin>>str2;
    len1 = strlen(str1);
    len2 = strlen(str2);
    if(len1==len2)
    {
      if(strcmp(str1, str2)==0)
        cout<<"\nStrings are Equal";
      else
        cout<<"\nStrings are not Equal";
    }
    else
        cout<<"\nStrings are not Equal";
    cout<<endl;
    return 0;
}
```

Thank You