Introduction to C++ Programming

erfanamkh2002@gmail.com

Table of contents


01
Problem Solving


02
Functions

Why do we need function?

- What is a function?
- Why should we use a function?
- Tell some examples...

Definition Syntax

```
return_type function_name(list_of_parameterTypes_and_parameters)
{
    code...
}
```

Function and Variables

```cpp
#include <iostream>

using namespace std;

void my_function()
{
    int a = 2 ;
    cout << "in function " << a << endl ;
}

int main()
{
    int a = 3 ;
    cout << "before " << a << endl ;
    my_function() ;
    cout << "after " << a << endl ;

    return 0;
}
```

```
before 3
in function 2
after 3
```

Function and Variables

```cpp
#include <iostream>

using namespace std;

void my_function(int input)
{
    input = 2 ;
    cout << "in function " << a << endl ;
}

int main()
{
    int a = 3 ;
    cout << "before " << a << endl ;
    my_function(a) ;
    cout << "after " << a << endl ;

    return 0;
}
```

Compilation failed due to following error(s).

```
main.cpp: In function ‘void my_function(int)’:
main.cpp:16:31: error: ‘a’ was not declared in this scope
   16 |     cout << "in function " << a << endl ;
      |                               ^
```

Function and Variables

```cpp
#include <iostream>

using namespace std;

void my_function(int input)
{
    input = 2 ;
    cout << "in function " << input << endl ;
}

int main()
{
    int a = 3 ;
    cout << "before " << a << endl ;
    my_function(a) ;
    cout << "after " << a << endl ;

    return 0;
}
```

```
before 3
in function 2
after 3
```

Function and Variables

```cpp
#include <iostream>

using namespace std;

void my_function()
{
    int my_number = 2 ;
    cout << "in function " << my_number << endl ;
}

int main()
{
    my_function() ;
    cout << my_number << endl ;

    return 0;
}
```

Compilation failed due to following error(s).

```
main.cpp: In function 'int main()':
main.cpp:22:13: error: 'my_number' was not declared in this scope
   22 |     cout << my_number << endl ;
      |             ^~~~~~~~~
```

Function and Global Variables

```cpp
#include <iostream>
using namespace std;

int a ;

void my_function()
{
    a = 3 ;
    cout << "in function " << a << endl ;
}

int main()
{
    a = 2 ;
    cout << "before " << a << endl ;
    my_function() ;
    cout << "after " << a << endl ;
    return 0;
}
```

```
before 2
in function 3
after 3
```

Variable Address

Call by Value/Call by Reference

Call by value	Call by reference
A copy of the value is passed to the function	An address of value is passed to the function
Changes made inside the function are not reflected on other functions	Changes made inside the function are reflected outside the function as well
Actual and formal arguments will be created at different memory location	Actual and formal arguments will be created at same memory location.

Call by Value/Call by Reference

```cpp
#include <iostream>

using namespace std;

void my_function(int &input)
{
    input = 3 ;
    cout << "in function " << input << endl ;
}

int main()
{
    int a = 2 ;
    cout << "before" << a << endl ;
    my_function(a) ;
    cout << "after" << a << endl ;
    return 0;
}
```

```
before2
in function 3
after3
```

Function and Arrays

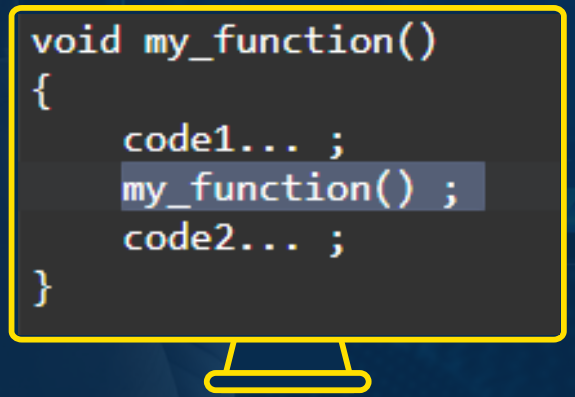
- Array_name
- Array_name and Size
- Using pointers

```cpp
#include <iostream>

using namespace std;

void my_function(int input[])
{
    for (int i=0 ; i<5 ; i++)
    {
        cout << input[i] ;
    }
}

int main()
{
    int a[5] = {1, 2, 3, 4, 5} ;
    my_function(a) ;
    return 0;
}
```

```cpp
#include <iostream>

using namespace std;

void my_function(int input[], int s)
{
    for (int i=0 ; i<s ; i++)
    {
        cout << input[i] ;
    }
}

int main()
{
    int a[5] = {1, 2, 3, 4, 5} ;
    my_function(a, 5) ;
    return 0;
}
```

Recursion

- Call a function in its own definition
- Recursive Algorithms



Default Arguments

```cpp
#include <iostream>
using namespace std;

void my_function(int input = 10)
{
    cout << input << endl ;
}

int main()
{
    my_function() ;
    my_function(3) ;
    return 0;
}
```

Overloading Function

- Different argument sets
- Different data types

Thanks!