#21 DAY

C++ COMPLETE BOOTCAMP

INSPIRE CLUB, MANIT BHOPAL

BRINGS

C++ Complete Bootcamp

Learn How To Apply Problem Solving Skills

Hello CPPBudddies

Day No. 21

Welcome

To

C++ COMPLETE BOOTCAMP

Your Guide To A Solid Foundataion in C++

Let us begin

Concept:

Pass by Value

Pass by Reference

Pass by Pointer

Reference vs Pointers

References are used to refer an existing variable in another name whereas pointers are used to store address of variable.

References cannot have a null value assigned but pointer can have null

A pointer in C++ is a variable that holds the memory address of another variable.

A reference is an alias for an already existing variable.

Once a reference is initialized to a variable, it cannot be changed to refer to another variable.

Hence, a reference is similar to a const pointer.

Pointer

- A pointer can be initialized to any value anytime after it is declared.

```
int a = 5;
// some code
int *p = &a;
```

- A pointer can be assigned to point to a *NULL* value.
- Pointers need to be dereferenced with a `*`.
- A pointer can be changed to point to any variable of the same type.

Reference

- A reference must be initialized when it is declared.

```
int a = 5;
int &ref = a;
```

- References cannot be *NULL*.
- References can be used ,simply, by name.
- Once a reference is initialized to a variable, it cannot be changed to refer to a variable object.

DEMO

Default Arguments

A default argument is a value provided in a function declaration that is automatically assigned by the compiler if the caller of the function doesn't provide a value for the argument with a default value.

Things to Remember

1. Once we provide a default value for a parameter, all subsequent parameters must also have default values. For example,

```
// Invalid
void add(int a, int b = 3, int c, int d);

// Invalid
void add(int a, int b = 3, int c, int d = 4);

// Valid
void add(int a, int c, int b = 3, int d = 4);
```

Case 1 : No argument is passed

```
void temp(int = 10, float = 8.8);

int main() {
  ... ...
  temp();
  ... ...
}

void temp(int i, float f) {
  // code
}
```

Case 2 : First argument is passed

```
void temp(int = 10, float = 8.8);

int main() {
  ... ...
  temp(6);
  ... ...
}

void temp(int i, float f) {
  // code
}
```

Case 3 : All arguments are passed

```
void temp(int = 10, float = 8.8);

int main() {
  ... ...
  temp(6, -2.3);
  ... ...
}

void temp(int i, float f) {
  // code
}
```

Case 4 : Second argument is passed

```
void temp(int = 10, float = 8.8);

int main() {
  ... ...
  temp(3.4);
  ... ...
}

void temp(int i, float f) {
  // code
}
```

Error

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

// A function with default arguments, it can be called with
// 2 arguments or 3 arguments or 4 arguments.
int sum(int x, int y, int z=0, int w=0)
{
    return (x + y + z + w);
}

/* Driver program to test above function*/
int main()
{
    cout << sum(10, 15) << endl;
    cout << sum(10, 15, 25) << endl;
    cout << sum(10, 15, 25, 30) << endl;
    return 0;
}
```

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

// A function with default arguments, it can be called with
// 2 arguments or 3 arguments or 4 arguments.
int sum(int x, int y, int z=0, int w=0)
{
    return (x + y + z + w);
}
int sum(int x, int y, float z=0, float w=0)
{
    return (x + y + z + w);
}
/* Driver program to test above function*/
int main()
{
    cout << sum(10, 15) << endl;
    cout << sum(10, 15, 25) << endl;
    cout << sum(10, 15, 25, 30) << endl;
    return 0;
}
```

Strings in C++

STRING

String is a collection of characters.
There are two types of strings commonly used in C++ programming language:

- Strings that are objects of string class (The Standard C++ Library string class)
- C-strings (C-style Strings)

C-strings

In C programming, the collection of characters is stored in the form of character arrays.
This is also supported in C++ programming.
Hence it's called C-strings.

C-strings are arrays of type char terminated with null character, that is, \0
(ASCII value of null character is 0).

DEMO

How to define a C-string?

```
char str[] = "C++";
```

In the above code, `str` is a string and it holds 4 characters.

Although, "`C++`" has 3 character, the null character `\0` is added to the end of the string automatically.

Alternative ways of defining a string

```
char str[4] = "C++";

char str[] = {'C','+','+','\0'};

char str[4] = {'C','+','+','\0'};
```

Like arrays, it is not necessary to use all the space allocated for the string. For example:

```
char str[100] = "C++";
```

string Object

In C++, you can also create a string object for holding strings.

Unlike using char arrays, string objects has no fixed length, and can be extended as per your requirement.

DEMO

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

int main()
{
    // Declaring a string object
    string str;
    cout << "Enter a string: ";
    getline(cin, str);

    cout << "You entered: " << str << endl;
    return 0;
}
```

Passing Strings to a Function

THANK YOU

keep calm,
wear mask,
and
study hard

ISSUED IN PUBLIC INTEREST

whoami

AKASH MAJI
[TCS DIGITAL]
Your Mentor