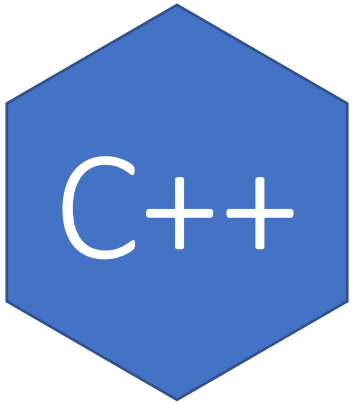
Computer Programming with



Baby Steps

Presentation By AwesomeKen

Outline

- Data types
- Operators
- Variables
- Arrays
- Comments
- Loops
- c-strings

Data Types

- C++ is a strongly typed language. ie the type of data to be stored or used in the program must be explicitly defined by the programmer.
- Data typed in C++ are grouped into in-built and user-defined types
- In-built: those that come by default with the language
- User-defined: Those that are created by the programmer or 3rd party companies.
- In-built types include char, bool, int, float, double, long

Data Types

Data Type	Example
char	‘a’, ‘b’, ‘A’, ‘8’, ‘*’, ‘@’
bool	true, false
int	3, 34, 89, 12341
float	32.23, 76.3123
double	66.666, 55.2902123
long	9834050183274209

Operators

- Operators allows the performance of mathematical operations on data.

Operator	Name	Operation
+	plus	Add numbers eg. 3 + 22.6
-	minus	Subtract numbers eg. 89 - 25
*	times	Multiply numbers eg. 6 * 8
/	Division	Divide numbers eg. 3/2
%	Modulo	Find the modulus eg. 19%2
=	Assignment	Assign value to a variable eg. Age = 53;
++	increment	Increment the current value by 1
--	decrement	Decrement the current value by 1

Operator	Operator Name	Operation
<	Less than	Add numbers eg.
>	Greater than	Subtract numbers eg. 89 - 25
<=	Less than or equal to	Multiply numbers eg. 6 * 8
>=	Greater than or equal	Divide numbers eg. 3/2
==	equality	Find the modulus
!=	Not equal to	Assign value to a variable
!	Not	Invert the a Boolean value

Operator	Name
&&	and
||	or

Bitwise Operators	Operator Name
&	Bitwise AND
|	Bitwise OR
^	XOR

Variables

- Variable allow for storage of input
- Format: *data_type name;*
- Eg. *int* age; *This form of creating a variable is called* Declaration
- A variable can be created and given a value at the point of creation this way is called Initialization.
- eg. *double* pie = 3.142;

Arrays

Arrays allows for multiple values of the same type

Arrays are zero index; indexing begins with zero

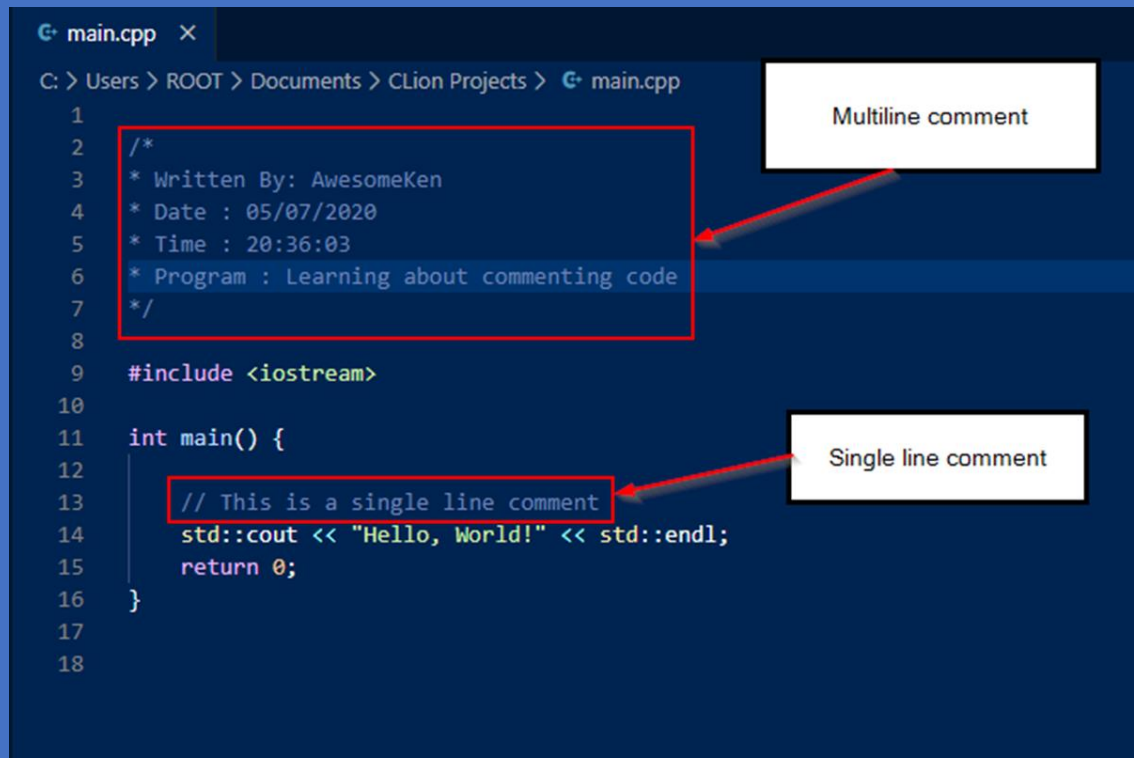
```
//Format for array
// data_type name[size];
```

```
//Creating an array
int ages[50];

//Initializing an array
int age[4] = { 23, 62, 59, 30 };
```

```
//Creating an array
int ages[50];

//Initializing an array
int age[4] = { 23, 62, 59, 30 };
```

Adding Comments

- One way of improving code readability is to comment code.
- Comments are not compiled, they are only texts that give information to the programmer.
- /* */ is used for multiline commenting
- // is used for single line commenting



Loops

Loops are used to perform repetitive tasks. For example to get he index numbers and names of students. This code has to do one thing several times. Hence with a loop the code is write just once and is repeated in a loop for the required number of times.

There are 3 basic types of loops in C/C++ :

- for loop
- while loop
- do while loop

for Loops

Suitable for Repetitive task who's last index is known

Loops until the end is reached.

```
//Format for FOR LOOP
/*
for (begin; condition; increment){
    //code
}
*/
```

testcode.cpp > main()

```
#include <iostream>

using namespace std;

int main () {

//Format for FOR LOOP
/*
for (begin; condition; increment){
    //code
}
*/

int lbound = 10;

for (int i=0; i < lbound; i++) {
    cout << i << endl;
}

    return 0;
}
```

while loop

A while loop will run forever until a conditi on is specified for it to stop. Lets look at the format.

```
/*
format

    while (condition) {
        //code
    }

*/
```

```
bool isOut = true;

while ( isOut == true ) {
    //code
    cout << " I am still out " << endl;
}
```

- A while loop without a condition to break out is called an Infinite Loop.
- Care must be taken not to create an infinite loop when not intended.

```
#include <iostream>

using namespace std;

int main()
{

    //print the value of the index from 0 to 9
    /* for (int i= 0; i <10; i++) {
        cout << i << endl;
    }
    */

    ///While loop to print from 1 to 10

    int i = 0; ///counter
    //bool isOut = true;

    while (i < 10 ) {
        cout << (i + 1) << endl;
        i++;
    }

    return 0;

}
```

begin

condition

increment

This is just like the for loop we wrote earlier

do while Loops

- What make the do while loop so special is its ability to run the code at least once.
- A careful look at the code you will see that the code first runs before the condition is checked.

```
/*
    Format

    do {
        // code
    } while ( condition);
*/
```

```
//Do while loop
int i = 0;
do {

    cout << "\tI run i = " << i << endl;

} while ( i > 1);
```

Runs code at least once

doWhileloop.cpp > main()

```
#include <iostream>

using namespace std;

int main() {

bool some_condition = true;

int key = 2345;
int id = 9090;
int user_id;
int key_code;
do
{
    /* code */
    cout << "UserId : " ;
    cin >> user_id;
    cout << "Key Code : " ;
    cin >> key_code;

    if ( key != key_code || id != user_id ){
        cout << "Incorrect UserId or key code. " << endl;
    }

} while ( key_code != key || id != user_id );

    return 0;
}
```

C-Style Strings

- These are null terminated strings.
- Eg “kofi” in reality is stored as ‘k’, ‘o’, ‘f’, ‘I’,’\0’
- ‘\0’ – is the null terminator. It indicates the end of the string.
- Therefore in creating a variable to store a string the null terminator must be taken into account.
- A string variable is nothing but a char array.
- `char str[6] = “kofi”;`

C-Style Strings

- Functions to help manipulate c-style string dat can be found in the `cstring` header file.
- It must be included to have access to the functions.
- `#include <cstring>`
- Functions such as strcmp, strlen etc.
- Strcmp => string compare returns 0 for match/same strings.

pwordCheck.cpp > ...

```cpp
#include <iostream>
#include <cstring> //include files containing string
//manipulation functions

using namespace std;

int main()
{

char uname[11] = "AwesomeKen";
char pword[9] = "nicecode";

char username[255];
char password[255];
do
{
    /* code */
    cout << "Username : " ;
    cin >> username; //get username
    cout << "Password : " ;
    cin >> password; //get password

//If password or username is incorrect display error msg
    if ( strcmp(password, pword) || strcmp(username, uname) ){
        cout << "Incorrect UserId or key code. " << endl;
    }
} while ( strcmp(password, pword) || strcmp(username, uname));

cout << "Log in successful! " << endl;

return 0;

}
```

End of Slides

Thank You