Crash Course

Antonio Scapellato

Intro

C++ is a cross-platform language that can be used to create high-performance applications.
C++ was developed by Bjarne Stroustrup, as an extension to the C language.
C++ gives programmers a high level of control over system resources and memory.
The language was updated 3 major times in 2011, 2014, and 2017 to C++11, C++14, and C++17.

Example

myfirst.cpp

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World!";  return 0;
}
```

Variables

In C++, there are different types of variables (defined with different keywords), for example:
int - stores integers (whole numbers), without decimals, such as 123 or -123.
double - stores floating point numbers, with decimals, such as 19.99 or -19.99.
char - stores single characters, such as 'a' or 'B'.
string - stores text, such as "Hello World".
bool - stores values with two states: true or false.

variable.cpp

```
int myNum;
myNum = 15;
cout << myNum;
```

User Input

input.cpp

```
int x;
cout << "Type a number: "; // Type a number and press enter
cin >> x; // Get user input from the keyboard
cout << "Your number is: " << x; // Display the input value
```

Loops

for

```
for (int i = 0; i < 5; i++) {
    cout << i << "\n";
}
```

while

```
int i = 0;
while (i < 5) {
    cout << i << "\n";
    i++;
}
```

do .. while

```
do {
    cout << i << "\n";
    i++;
}while (i < 5);
```

Arrays

```
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
cout << cars[0];// Outputs Volvo
```

Functions

```
// Create a function
void myFunction() {
    cout << "I just got executed!";
}
int main() {
    myFunction(); // call the function
    return 0;
}
// Outputs "I just got executed!"
```

OOP

OOP stands for Object-Oriented Programming. Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions. Object-oriented programming has several advantages over procedural programming:OOP is faster and easier to executeOOP provides a clear structure for the programsOOP helps to keep the C++ code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debugOOP makes it possible to create full reusable applications with less code and shorter development time.

Class

```
class MyClass {      // The class
    public:          // Access specifier
    int myNum;       // Attribute (int variable)
    string myString;  // Attribute (string variable)
};
```

Object

```
MyClass myObj;  // Create an object of MyClass
myObj.myNum = 15;
myObj.myString = "Some text";
// Print attribute values
cout << myObj.myNum << "\n";
cout << myObj.myString;
return 0;
```

Thank you!

antonioscapellato.com