C++ Associate Programing

Day one - First step in C++

Lesson Objectives

- Introduce course
- History of C++
- Set up C++ Development Environment
- Writing first C++ program
- Compile and Execute C++ Program
- Basic Syntax

Section 1

INTRODUCE COURSE

Introduce course

- *C ++ Associate Training is a basic training program of Fsoft Academy.*
- *Includes general knowledge such as data type, variable, function, pointer, reference, operator, control flow, …*
- *Students will learn theory and practice in parallel with the help of experienced trainers and mentors.*
- *Thereby understanding the core knowledge specific to the language and used to solve practical problems in future projects.*

Section 2

HISTORY OF C++

History of C++

- In 1979, Bjarne Stroustrup, a Danish computer scientist, began work on "C with Classes", the predecessor to C++.
- In 1982, Stroustrup started to develop a successor to C with Classes, which he named "C++" .
- In 1985, the first edition of *The C++ Programming Language* was released.
- In 1989, C++ 2.0 was released, followed by the updated second edition of *The C++ Programming Language* in 1991.
- In 1998, C++98 was released, standardizing the language, and a minor update (C++03) was released in 2003. And next versions C++11 (in 2011), C++14(2014), C++17(2017-2018).
- As of 2019, C++ is now the fourth most popular programming language, behind Java, C, and Python.

Section 3

SET UP C++ DEVELOPMENT ENVIRONMENT

Set up C++ Development Environment

- *All examples will be written by Visual Studio (2019). Students can set up an IDE or editors: Code::Block, Eclipse, DevC++, …*
- *Go to link: https://visualstudio.microsoft.com/downloads/*
- *Choose Community or Professional to download Installer*
- *Run Installer file, choose folder location to storage*
- *Pick components of C++ and set up*
- *Setup another editor to support: Notepad++, Sublime Text*

Section 4

WRITING FIRST C++ PROGRAM

Writing first C++ program

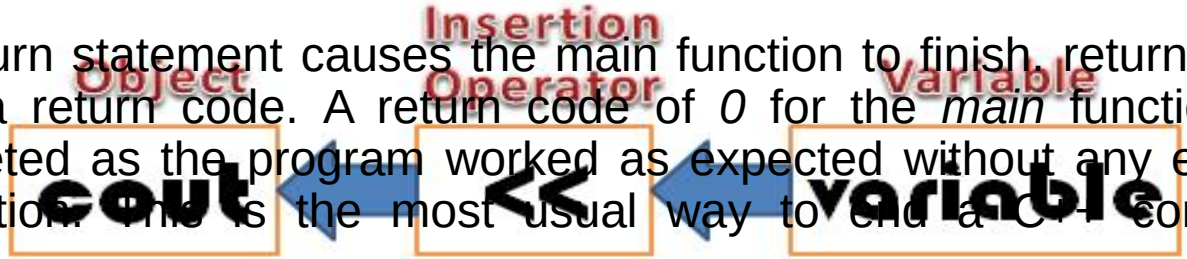
```
/* This is a simple C++ Program Structure.
Call this file is sample.cpp. */
#include <iostream>
using namespace std;
int main() // A C++ program begins at main().
{
    cout << "Hello everybody!";   // output is: Hello everybody!
    return 0;
}
```

Writing first C++ program: Explanation P1

- /* */, //: Are comment line in C++ which is ignored by compiler.
- #include <iostream>: Lines beginning with a hash sign (#) are directives for the preprocessor. This include tells the preprocessor to include the iostream standard file.
- using namespace std: All the elements of the standard C++ library are declared within what is called a namespace, the namespace with the name std. So in order to access its functionality we declare with this expression that we will be using these entities.
- {}: Left brace { begins function body and corresponding right brace } ends function body.
- ";": Statements end with a semicolon

Writing first C++ program: Explanation P2

- int main(): Each C++ program starts with the main function. Return type of the C++ main function is integer, whenever main function returns 0 value to operating system then program termination can be considered as smooth or proper.
- cout: Is the name of the standard output stream in C++, and the meaning of the entire statement is to insert a sequence of characters into the standard output stream.
- <<: Insertion operator (<<) is used to display value to the user on the console screen.
- return 0: The return statement causes the main function to finish. return may be followed by a return code. A return code of *0* for the *main* function is generally interpreted as the program worked as expected without any errors during its execution. This is the most usual way to end a C++ console program.



Section 5

COMPILE AND EXECUTE C++ PROGRAM

Compile and Execute C++ Program

- Open a text editor (IDE) and add the code as above.
- Save the file as: sample.cpp
- In IDE click on Build and Run.

```cpp
/*
This is a simple C++ program.
Call this file Sample.cpp.
*/
#include <iostream>
using namespace std;

// A C++ program begins at main().
int main()
{
    cout << "Hello everybody!\n";
    return 0;
}
```

```
Hello everybody!

C:\Users\Administrator\Desktop\Project1\x64\Debug\Project
1.exe (process 10744) exited with code 0.
      To automatically close the console when debugging s
tops, enable Tools->Options->Debugging->Automatically clo
se the console when debugging stops.
Press any key to close this window . . .
```

Section 6

BASIC SYNTAX

Basic Syntax: Identifier

- Various data items with symbolic names in C++ is called as Identifiers. Following data items are called as Identifier in C++ : Names of functions, arrays, variables, classes.
- Rules of naming identifiers in C++ :

- C++ is **case-sensitive** so that Uppercase Letters and Lower Case letters are different.

- The name of identifier **cannot begin with a digit**. However, Underscore can be used as first character while declaring the identifier.

- Only **alphabetic characters, digits and underscore** (_) are permitted in C++ language for declaring identifier.

- Other **special characters** are not allowed for naming a variable / identifier

- **Keywords** cannot be used as Identifier.

Basic Syntax: Identifier - Variable Naming

- **Some Facts About Identifier :**
  - ✓ It is name given to program element.
  - ✓ Identifier are the names is given by the programmer.
  - ✓ An identifier is used for any variable, function, data definition etc.
  - ✓ We can give any valid name to the identifier.

- **The rules in C++ for identifiers are :**
  - ✓ Only Alphabets,Digits and Underscores are permitted.
  - ✓ Identifier name cannot start with a digit.
  - ✓ Key words cannot be used as a name.
  - ✓ Upper case and lower case letters are distinct.
  - ✓ Special Characters are not allowed
  - ✓ Global Identifier cannot be used as “Identifier”.

Basic Syntax: Identifier - Variable Naming Sample

Valid examples are :

Identifier	Note
Name	Capital Letter and Small Letters are Allowed
name	Small Letters are allowed
name_1	Digits and Underscore is allowed along with alphabets
Int	Keywords are allowed but we have to change case of any letter or complete word
INT	Keywords are allowed but we have to change case of any letter or complete word
_SUM	Underscore at the first position is allowed in C++ language
sum_of_the_numbers	We can concatenate multiple words with underscore
firstName	Best Style to concatenate multiple words (Changing case of First Letter of Successive Word)
Identifier	We can give concept name as Identifier name
printf	As we are not going to include stdio.h header file we can use printf as identifier.

Basic Syntax: Identifier - Variable Naming Sample

Invalid examples are :

Identifier	Explanation
int	Keyword name cannot be given to Variable/Identifier
pow	pow() is defined in math.h. This variable is legal if we haven't included math.h in our program. As soon as we include math.h header file in program this identifier will be illegal.
$sum	$ sign can be used in other programming language for creating identifier, however C/C++ do not support '$' sign.
num^2	special characters are not allowed.
num 1	Spaces are not allowed in C++ programming language for declaring identifier.
2num	Digits are allowd but not as first Character

Basic Syntax: Legal Characters

Token	Example
Alphabets	A-Z and a-z
Digits	0-9
Special characters	$%^

Special Characters Type	Example
Arithmetic Operators	– + * / %
Logical Operators	& !
Brackets	( ) { } [ ]
Relational Operators	< > = #
Other Symbols	: ; _(underscore) >> ?

Term	Definition
Alphabet	An alphabet is a standard set of letters (basic written symbols or graphemes) for the listing of words
Digits	A digit is an element of a set that taken as a whole comprises a system of Numeration
Special Character	ASCII printable characters are called as Special Characters

Basic Syntax: Keywords

- In C++, keywords are reserved identifiers which cannot be used as names for the variables in a program.

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

Lesson Summary

Summarize the main points in the lesson, compared to the lesson objectives:

1. Introduce course
2. History of C++
3. Set up C++ Development Environment
4. Writing first C++ program
5. Compile and Execute C++ Program
6. Basic Syntax

Thank you

Q&A