The C++ Programming Language

Prepared by: Meet Thakar

Road map

- History
- Application of C++
- Examples
- Input and Output operators
- Tokens
- Structure of a C++ Program

History

History

- C++ is a middle-level programming language which was created by **Bjarne Stroustrup** in **1979** at **Bell Labs**.
- C++ is derived from **C** and **Simula67** (An object-oriented language).
- During initial stage of development, C++ was called ***C with Classes.***
- It was later changed to "***C++***" because he considered the language to be an incremental version of the C language.
- It supports both ***Procedure-Oriented Programming (POP)*** and ***Object-Oriented Programming (OOP)***.
- C++ is superset of C, meaning any valid C program is also a valid C++ program.

Application of C++

Prepared by: Meet Thakar

Application of C++

- Same as C, C++ is also used in various fields of software development such as:
  - Operating Systems
  - Embedded Systems
  - Performance-Critical Software
  - Compilers and Interpreters
  - Game Development
  - Internet-of-Things (IoT)
  - Robotics

The C++ Programming Language

- The classic “Hello, World!” program:

```
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    cout << "Hello, World!";
    getch();

    return 0;
}
```

The C++ Programming Language

- Taking Input from the user:

```
#include <iostream>
#include <conio.h>

using namespace std;

int main()
{
    int num;

    cout << "Enter number: ";
    cin >> num;

    cout << "You entered: " << num;

    getch();
    return 0;
}
```

Input and Output operators

Prepared by: Meet Thakar

Output operator

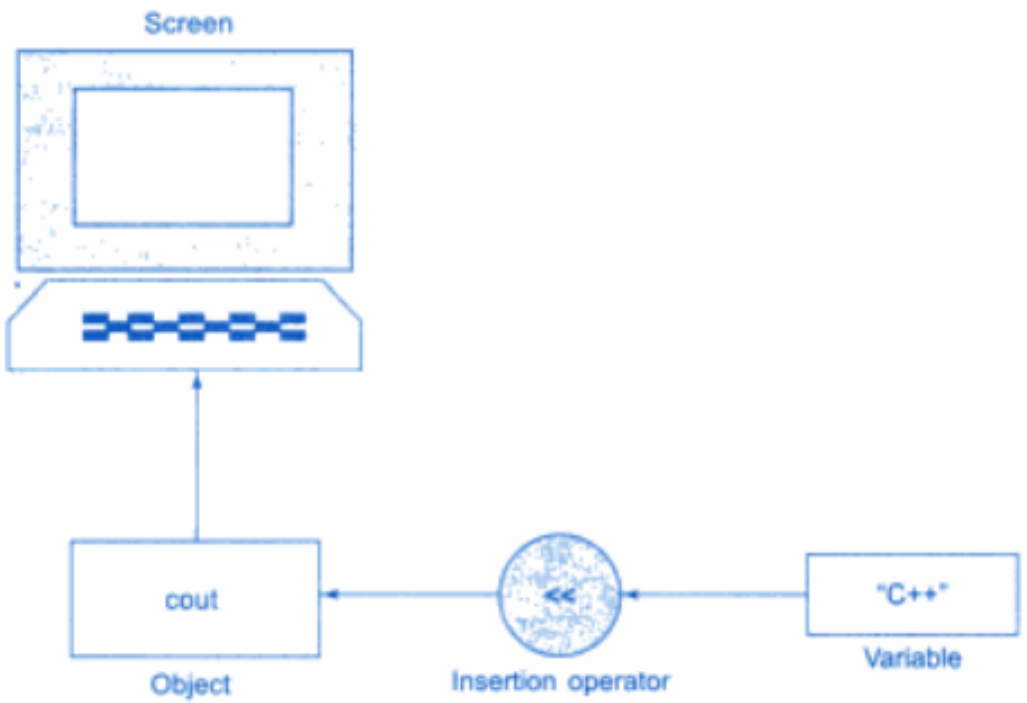
- In the first example, the following statement causes the string to be displayed on the screen:

```
cout << "Hello, World!";
```

- The identifier ***cout*** (pronounced "C out") **:**
  - Stands for *"Console Output"*.
  - It is a predefined object that represents the standard output stream (monitor).
  - It is defined inside the ***iostream*** header file.

Output operator

- The operator <<:
  - It is called the ***Insertion Operator*** or ***Put-to Operator***.
  - It sends the content on the right to object on the left.



Input operator

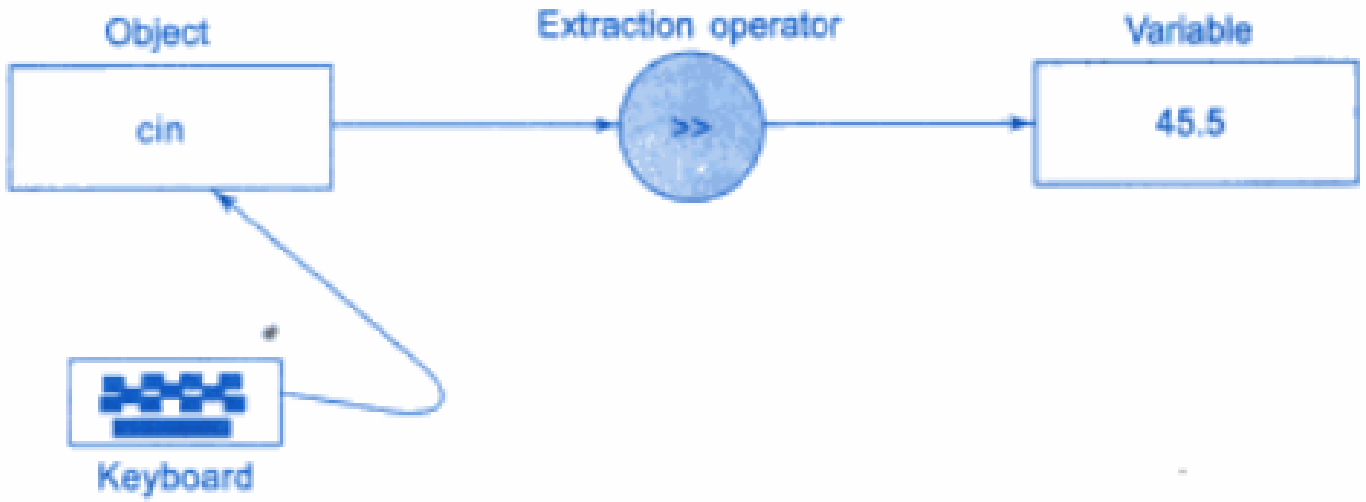
- In the second example, the following statement causes the program to wait for the user to enter a number:

```
cin >> num;
```

- The identifier ***cin*** (pronounced "C in") **:**
  - Stands for *"Console Input"*.
  - It is a predefined object that represents the standard input stream (keyboard).
  - It is defined inside the ***iostream*** file.

Input operator

- The operator **>>**:
  - It is called the ***Extraction Operator*** or ***Get-from Operator***.
  - It extracts (gets) the content from the keyboard and sends it to the variable on its right.



Tokens

Prepared by: Meet Thakar

Tokens

- A token is the smallest element of a C++ program that is meaningful to the compiler.
- They’re the basic building blocks of a program.
- The following are the C++ tokens:
  - Identifiers
  - Keywords
  - Constants
  - Strings
  - Operators

Identifiers

- The names given to variables, functions, structures or any user-defined item is known as ***Identifier.***
- Rules for a valid Identifier:
  - Only alphabets, numbers and underscores (_) are allowed.
  - Name must start with either an alphabet or an underscore.
  - Uppercase and Lowercase letters are considered different.
    - Means variables ***number*** and ***Number*** are considered different.
  - Keyword cannot be used.

Keywords

- In a programming language, keywords are reserved identifiers that have predefined meaning.
- They cannot be used as identifiers.

Keywords

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

Constants/Literals

- Constants are fixed values which cannot be changed.
- They're also called ***Literals.***
- These are the types of literals:
  - Integer constants (for example, 10)
  - Float constants (for example, 10.5)
  - Octal constants (for example, 065)
  - Hexadecimal constants (for example, 0xFF)
  - Character constants (for example, 'A')
  - String constants (for example, "Hello, World")

Structure of a C++ Program