C++ Programming

C++

C++ is a general-purpose programming language

Implementations of C++ exist from some of the most modest microcomputers to the largest supercomputers and for almost all operating systems.

C++ is a superset of the C programming language. In addition to the facilities provided by C, C++ provides flexible and efficient facilities for defining new types.

The C++ language support four programming styles:
- Procedural programming
- Data abstraction
- Object-oriented programming
- Generic programming

C++ supports systems programming.

History:

C++ developed by **Bjarne Stroustrup** in 1979 at AT&T Bell Laboratory.

C++ developed using class concept of Simula(already existing language which supported OOP paradigm) and C's efficiency and flexibility for systems programming.

Templates and exceptions added to C++ later.

1979: Work on ''C with Classes'' started. The initial feature set included classes and derived classes, public/private access control, constructors and destructors, and function declarations with argument checking.

1984: ''C with Classes'' was renamed to C++. By then, C++ had acquired virtual functions, function and operator overloading, references, and the I/O stream, number libraries.

1985: First commercial release of C++

1991: The C++ Programming Language, Second Edition, presenting generic programming using templates and error handling based on exceptions

1997: The C++ Programming Language, Third Edition [Stroustrup,1997] introduced ISO C++, including namespaces, dynamic_cast, and many refinements of templates. The standard library added the STL framework of generic containers and algorithms

2011: ISO C++11 standard was formally approved. In 2012 The first complete C++11 implementations emerged. 2012 Work on future ISO C++ standards (referred to as C++14 and C++17) started

2017: the final standard was published in December 2017.[

Need of C++

C++ is a systems programming language used for developing native applications

Software written in C++ is everywhere: It is in your computer, your phone, your car, and other many electronic devices

Many operating systems have been written in C++ like Windows, Apple's OS, Linux, and most portable-device OSs..

C++ is used to write device drivers and other software that rely on direct manipulation of hardware under real-time constraints.

C++ is used to write some critical parts of most widely used systems like Amazon, Google, Facebook.

Many other advanced technologies depends on C++'s performance and reliability in their implementations – JVM of Java Technologies, Web services framework of Microsoft's .NET technologies, JavaScript interpreters of many browsers like Microsoft's Internet Explorer, Mozilla's Firefox, Apple's Safari, and Google's Chrome.

Games has been another major applications area for C++.

C Programming Language	C++ Programming Language
C was developed by Dennis Ritchie between the year 1969 and 1973 at AT&T Bell Labs.	C++ was developed by Bjarne Stroustrup in 1979 at AT&T Bell Labs
C supports procedural programming.	C++ is hybrid language, because it supports procedural and object oriented programming and generic programming.
C is a function driven language	C++ object driven language
Functions are written and data is passed as parameter in functions.	Data and functions are encapsulated together in an object in C++.
Different operators are supported on built-in data types	Different operators are supported on built-in data types as well as user defined data type
malloc(), calloc(), realloc(), free() functions are used for memory management	Operators new and delete are used for memory management
	Features like – namespace, template, exception handling, reference variable, function and operator overloading are supported in C++
Total 32 keywords in Standard C	Total 63 keywords in C++

First C++ Program :

```cpp
#include<iostream>

using namespace std;

int main()

{
   cout<<"Hello World..";

   return 0;
}
```

Tokens in C++ : In C++, tokens can be defined as the smallest building block of C++ programs that the compiler understands. Every word in a C++ source code can be considered a token.

Before translation parser does parsing of each statement and finds out tokens. Each token has some pre-defined meaning in compiler. Compiler translates each token in machine understandable form and returns .exe file. This .exe file is executable file, which can be executed directly on OS prompt.

Types of tokens :

- Identifiers
- Keywords
- Constants
- Datatypes
- Strings
- Operators
- Escape sequence

Identifier :   It is user-defined word.   Used to define and declare variable_name, function_name, user_defined data types etc..

Rules to define identifier:

1. It should start with a alphabet or _ (underscore)
2. White space and special characters are allowed in name of identifier
3. It can have alphabet, digits and _ (underscore)
4. It cannot be a keyword
5. It must be unique (can only be declared once in a namespace)
6. C++ is case sensitive, so be careful.

Keywords : C++ is case sensitive. All keywords are in lower-case.

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

#define MAX 100

   Preprocessor directive, works like find and replace. Before compilation all occurrences of word MAX in program are replaced by 100.

Constants :

const int i = 10;
   The value of i will remain unchanged for entire program.

constexpr double area = 3.14 * r * r;
   The expression is evaluated at compile time and value is kept in area name, which is constant.

Demonstration: constDemo.cpp
Discuss Assignment #1

Datatypes:

We use the variables to store values in programs, but the OS has to know what kind of data we want to store in them, since OS is going to allocate some amount of memory to store a value.

A declaration is statement that introduces a variable name into the program. It specifies a datatype for the variable

A datatype defines a set of possible values and a set of operations we can perform on those values.

C++ offers a variety of primitive (built-in) datatypes.

Datatypes supported in C++ : Size and range depends on the system the program is compiled for,

Name	Description	Size*	Range*
char	Character or small integer.	1byte signed:	-128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	Wide character.	2 or 4 bytes	1 wide character

Discuss Assignment #2

string in C++ :

Variables that can store non-numerical values that are longer than one single character are known as strings.

The C++ language library provides support for strings through the standard string class. This is not a primitive type, but it behaves like a primitive types in its most basic usage.

Demonstration: stringDemo.cpp
Discuss Assignment #3

Operator: C++ has very rich set of operators.

Assignment operator =

The part at the left of the assignment operator (=) is known as the lvalue (left value) and the right one as the rvalue (right value).
The lvalue has to be a variable whereas the rvalue can be either a constant, a variable, the result of an operation or any combination of these.
The most important rule when assigning is the right-to-left rule:
The assignment operation always takes place from right to left, and never the other way:

    a = b;

This statement assigns to variable a (the lvalue) the value contained in variable b (the rvalue).
The value that was stored until this moment in a is not considered at all in this operation, and in fact that value is lost.

A property that C++ has is that the assignment operation can be used as the rvalue (or part of an rvalue) for another assignment operation.
  For example: a = 2 + (b = 5);

is equivalent to:

b = 5; a = 2 + b;

that means: first assign 5 to variable b and then assign to a the value 2 plus the result of the previous assignment of b (i.e. 5), leaving a with a final value of 7.

The following expression is also valid in C++:

a = b = c = 5;   It assigns 5 to the all the three variables: a, b and c

Arithmetic operators (+, -, *, /, %) : Need 2 operands.

The five arithmetical operations supported by the C++ language are:

+ addition
- subtraction
* multiplication
/ division
% modulo

Compound assignment (+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=)

Increment and decrement (++, --)

Relational and equality operators (==, !=, >, >=, <, <=)

Logical operators (!, &&, ||)   **Conditional operator**   **or**   **Ternary operator**   (? :)

Bitwise Operators (&, |, ^, ~, <>)

Comma operator (,)

Explicit type casting operator   **sizeof() operator**

dot operator (.)   **Scope resolution operator** (::)

Arrow operator (->)

- **OperatorDemo.cpp**
- **Implicit_Explicit_Casting.cpp**

Discuss Assignment #4

Special Characters and Escape Sequence :

Sequence	Meaning
\a	Alert.
\b	Backspace.
\f	Form feed.
\n	Newline.
\r	Carriage return.
\t	Horizontal tab.
\v	Vertical tab.
\\	Backslash (\ ).
\'	Single quote (').
\"	Double quote (").
\?	Question mark (?).

Demo: escSequence.cpp

Control Structures in C++:

Condition Structure:
    if..else if..else

Loop Structures:

    for loop
    while loop
    do..while loop

Keywords work with loop structure:

    break
    continue

** Assignments based on condition and loop structures **
Discuss Assignment #5 and #6

Functions in C++ :

A function is a group of statements that is executed when it is called from some point of the program.

Functions are used to structure our programs in a more modular way.

Gives feature of reusability because of its write once and call multiple times mechanism.

They are written to perform a specific task and are also called as module or sub-program or procedure or sub-routine.

A function can optionally define input parameters that enable callers to pass arguments into the function.

A function can optionally return a value as output.

Every function should be called directly or indirectly from main(). main() function is entry point function for C++ program.

type function_name (parameter1, parameter2, ...) { statements }

where:

• type is the data type specifier of the data returned by the function.
• name is the identifier by which it will be possible to call the function.
• parameters (as many as needed): Each parameter consists of a data type specifier followed by an identifier, like any regular variable declaration (for example: int x) and which acts within the function as a regular local variable. They allow to pass arguments to the function when it is called. The different parameters are separated by commas.

• statements is the function's body. It is a block of statements surrounded by braces { }.

Different forms of functions:

- Function returning nothing and taking no arguments.
- Function return a value and taking no arguments.
- Function returning nothing and taking arguments.
- Function returning a value and taking arguments.

Function declaration, definition and call should match to avoid syntax error.

Function parameters with default value: Demonstration (DefaultParameter.cpp)