

Programming Fundamentals

Week 1

Why to develop C++

C	C++
Built-in data types is supported in C.	Built-in & user-defined data types is supported in C++.
C is a function driven language because C is a procedural programming language.	C++ is an object driven language because it is an object oriented programming.
Function and operator overloading is not supported in C.	Function and operator overloading is supported by C++.
C is a function-driven language.	C++ is an object-driven language
Functions in C are not defined inside structures.	Functions can be used inside a structure in C++.
Namespace features are not present inside the C.	NAMESPACE is used by C++, which avoid name collisions.
Header file used by C is stdio.h.	Header file used by C++ is iostream.h.

History of C++

- The C++ programming language has a history going back to 1979, when [Bjarne Stroustrup](#) was doing work for his Ph.D. thesis.
- C++ is an OOP i.e. Object Oriented Programming, which allows programmers to develop large and complex applications.
- The OOP languages existed before C++ were slow and inefficient.
- C++ programming language is extension to C Language.
- In C we have already used increment operator (++) . Therefore we called C++ as “**Incremented C**” means **Extension to C**.

History of C++ (2)

- In 1967, BCPL language was developed by Master Martin to write operating system and compilers.
- After two years B language was developed with advanced features. B UNIX operating system was first developed in 1970 at Bell Laboratory.
- In 1972, advanced version of B language was developed known C language and the next version of UNIX was revised and written in C.
- In early 1980's , C++ was developed by Bjarne Stroustrup .
- In 1985, Stroustrup's reference to the language entitled *The C++ Programming Language* was published.
- That same year, C++ was implemented as a commercial product.
- The language was updated again in 1989 to include protected and static members, as well as inheritance from several classes.

Basic Elements of C++

- C++ character set
- Character sets are the character used to write a language.
 - C++ character set means the characters and the symbols that are understandable and acceptable by the C++ Program.
 - These are grouped to create and give the commands, expressions, words, c-statements, and some of the other tokens for the C++ Language.

C++ character set

- Alphabets
 - The alphabets are symbolized by A-Z & a-z, whichever is applicable. C- Language is case sensitive so it always takes dissimilar meanings for the lower case and the upper case letters. Thus, all 26 letters are usable in C-programming.
- Digits
 - The digits are symbolized from 0-9 or by combining these digits together. With the use of digits, the numeric constant is easily writable. All the 10 digits starting from 0 and ending at 9 are usable in the C-programming.

C++ character set

- Special symbols
 - There is a total of thirty special symbols that we can use in the C-programming. Moreover, special symbols are usable for the C-statements like; to make an arithmetic statement +, -, * etc., for creating relational statement <, >, <=, >=, == etc. , for creating assignment statement =, for creating logical statement &&, || etc. are necessary.
- White spaces
 - White spaces have blank space, newline return, tab space (Horizontal), Form feed, carriage ctrl, etc. are all usable for special purposes.

C++ words

- Reserved words
 - User defined words/identifier
1. Reserved words
 - Keywords (also known as reserved words) have special meaning to the C++ compiler and are always written or typed in short (lower) cases.
 - Keywords are predefined reserved identifiers that have special meanings.
 - E.g. C++ provides 64 keywords – for, break, continue, switch, int, float, double, char, try, catch, while, etc.

C++ words : Examples of Reserved words

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

C++ words : used defined words/identifiers

- User Defined / Identifier

- Identifier are the user defined names given to different parts of C++ program. Identifiers are not reserved words.
- It refers to the names of variable, function, arrays , classes etc.
- An identifier can consist of alphabetical characters, digits and underscores.
- All C++ variables must be identified with unique names. These unique names are called identifiers.
- Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

Rules for identifier

- The first character must be an alphabet or underscore (`_`) e.g. `age`, `marks`, `_balance`
- The remaining can be alphabetic character, digits and underscore as `age2`, `total_marks`
- The reserved words can't be used as identifiers. Example `int` , `double` .
- Upper case and lower case character are distinct. E.g. `name` is different from `NAME`.
- It cant contain whitespaces and special characters.