Structured programming

Class :First Class المرحلة الاولى

Lecture : third المحاضرة الثالثة

Lecturer : Maha Ali Hussain

م. مها علي حسين

محاضرات الدراسة المسائية

3.1 What Is C++?

C++ is a high-level programming language that allows a software engineer to efficiently communicate with a computer.

C++ is a highly flexible and adaptable language. Since its creation in 1980, it has been used for a wide variety of programs including firmware for micro-controllers, operating systems, applications, and graphics programming. C++ is quickly becoming the programming language of choice. There is a tremendous demand for people who can tell computers what to do, and C++ lets you do so quickly and efficiently.

C was designed with one goal in mind: writing operating systems. The language was extremely simple and flexible and soon was used for many different types of programs. It quickly became one of the most popular programming languages in the world.

Home Work : Write an article or scientific report on a history of C and C++ language ?

3.2 How to Learn C++

The only way to learn how to program is to write programs. You'll learn a lot more by writing

and debugging programs than you ever will by reading this book. This book contains many programming exercises, and you should try to do as many of them as possible. When doing the

exercises keep good programming style in mind. Always comment your programs, even if you're doing the exercises only for yourself. Commenting helps you organize your thoughts, and commenting your own programs is good practice for when you go into the "real world."

3.3 C++ Program Structure (syntax)

```
#include <iostream>
                                        using namespace std;
int main()        // main() is where program execution begins.
{
C++ statements;
……..
…..
return 0;
}
```

3.4 Character Set

Character set : is a combination of English language (Alphabets) and math's symbols (Digits and Special symbols such that these characters are combined to form sentences and sentences are combined to form paragraphs. The character set are set of words, digits, symbols and operators that are valid in C++. The main types of Character Set are :-

Alphabets : are represented by A-Z or a-z. C++ Language is case sensitive programming language, so it takes different meaning for small and upper case letters. For example; Manpower and manpower are two different identifiers in C++. There are total 26 letters used in C++ programming.

Digits : Digits are represented by 0-----9 or by combination of these digits. By using the digits numeric constant can be written easily. There are total 10 digits used in the C++programming.

Special Character: we have a set of special characters that can be used in C++. All these characters are used for various purposes.

, (comma)	" (double conations)	. (dot)	: (colon)	; (semicolon)
?	'	!	|	\

/	~	_	$	%
#	&	^	*	-
+	<	>	(	)
{	}	[	]	=

3.5 Identifier

A C++ identifier is a name used to identify a variable, function, class,). Functions, array, structures module, or any other user-defined item. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores, and digits (0 to 9).

Examples: Here are some examples of acceptable identifiers:

Mohd zara abc move_name a_123 cont1 flg_min A30m yname50 _temp j a23b9 retVal retval

Examples of invalid identifiers: 3v1, my name, true .

Rules of naming Identifiers :

- Identifiers can have alphabets, digits and underscore sign characters.
- They must not be a keyword or Boolean literal or null literal.
- They must not begin with a digit.
- They can be within a length (up to 127charachter).
- They cannot contain a space
- C++ is case sensitive i.e., upper-case letters and lower-case letters are treated differently.

3.6 Keywords in C++

Keywords are those words who has special meaning for compiler. We can't use keywords as variable name. The keywords should be in lower case letter. The keywords are also identifiers but cannot be user defined.

C++ has 32 Keywords as follows:

Keywords			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	go to	sizeof	volatile
do	If	Static	while

3.7 Data Types in C++

Data types are used to define a variable. Data types represent the type of information present in a variable. Data types are the keywords, which are used for assigning a type to a variable.

There are Different Data Types available in C++:

1. **Integer Type :** Integer data type are like whole numbers, they also include negative numbers but does not support decimal numbers.

Type	Storage size	Value range
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

2. **float-point Type :** Float data type allows user to store decimal values in a variable.

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

3. **Character Type :** Character data type is used to store only one letter, digit, symbol at a time.

Type	Storage size	Value range
Char	1 byte	-128 to 127 or 0 to 255

unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127

3.8 Variables in C++

Variables are used to store values. variable name is the name of memory location where value is stored. It must be alphanumeric, only underscore is allowed in a variable name. It is composed of letters, digits and only underscore. It must begin with alphabet or underscore. It cannot be begin with numeric.

Declaration of Variable

Declaration will allocate memory for specified variable with garbage value.

Syntax :

```
Data-Type    Variable-name;
```

Examples :

```
int a;
float b;
char c;
```

Initialization of Variable

Initialization means assigning value to declared variable. Every value will overwrite the previous value.

Examples :

```
int  a = 10;
float  b = 4.5;
 char  c = 'a';
  Character value must be enclosed with single quotes.
a=4.5;
```

if we assign decimal value to integer variable, it will accept only integer portion of value. In the above example variable a will accept 4 only.