Chapter 1: Fundamental Notions

Main Elements of C++ Programs

The program in the example below performs a very simple function. This program is used to find the product of two numbers.

```
#include<iostream>    // include the standard input-output Header File
using namespace std;
main()                // main function
{                     // begin of the main function
    int a,b,c;        // declaration
    a=2;              // assign value to variable a
    b=5;              // assign value to variable b
    c=a*b;            // find the product result
    cout<<c;          // print the result on the screen
}                     // end of the main function
```

- ✓ Each program has one **main()** function.
- ✓ Each program has a compound statement **{}**, which begins immediately after the **main()** function and extends until the end of the program. The compound statement is the body of the program itself.
- ✓ Each variable in the program must be **declared**, the declaration of the variables **a**, **b**, and **c** are integer variables in this program.
- ✓ You must have the header file (**iostream**) to print values using the standard output device.
- ✓ Each step in C++ language must end with (**;**)

To help us declare library functions, the C++ implementation provides header files containing the necessary declarations for various parts of the library. For example, the header file (**math.h**) contains declarations for mathematical functions, the header file (**iostream**) contains declarations for standard input and output functions, and so on.

Programming With C++ Language

Output Function (cout)

This function is used to print any type of variables or phrases on the screen.

Example 1:

```
#include<iostream>
using namespace std;
main()
{
    cout<<"This is the first program.";
}
```

Output:

```
This is the first program.
```

Example 2:

```
#include<iostream>
using namespace std;
main()
{
    cout<<"This is the first program.";
    cout<<" It is so easy.";
}
```

Output:

```
This is the first program. It is so easy.
```

NOTE: To print in a new line use **endl**.

Example 3:

```
#include<iostream>
using namespace std;
main()
{
    cout<<"This is the first program.";
    cout<<endl;
    cout<<"It is so easy.";
}
```

Output:

```
This is the first program.
It is so easy.
```

The Comments

A comment that is included in a program for the benefit of a person who reads the program in the future. To advise the compiler that a group of characters is a comment and should be **ignored**. Two types of comments can be written:

1. The comment consists of one statement, use (//) before the statement.
2. The programmer must place the comment between (/*) and (*/) when the comment consists of several statements.

Example 4:

```
/* This program will print into the screen
   in the first line "Welcome"
   and will print in the second line "Student" */
#include<iostream>
using namespace std;
```

```
main()
{   // Starting of the main function
    cout<<"Welcome"<<endl<<"Student";
}   // Ending of the main function
```

Output:

```
Welcome
Student
```

Data Type

Type	Size (Bytes)	Range
`bool`	1	0 **to** 1
`char`	1	-128 **to** 127
`int`	2	-32,768 **to** 32,767
`long`	4	-2,147,483,648 **to** 2,147,483,647
`float`	4	$3.4 \times (10^{-38})$ **to** $3.4 \times (10^{+38})$
`double`	8	$1.7 \times (10^{-308})$ **to** $1.7 \times (10^{+308})$

Identifiers

- ✓ Identifiers can consist of the **UPPERCASE** letters (**A** to **Z**), the **lowercase** letters (**a** to **z**), the **digits** (**0** to **9**), and the **underscore** character (**_**).
- ✓ The first character **must** be a **letter** or an **underscore**.
- ✓ There can be **no** embedded **blanks**.
- ✓ **Reversed** words **cannot** be used as identifiers.
- ✓ Identifiers are **case-sensitive**. Therefore, **tax**, **Tax**, and **TAX** are distinct.

Valid Identifiers

a	b6	Sum	_new	x_ray	OK_

Invalid Identifiers

7_up	B6.1	Suzan!	π2

Keywords

Also known as (**reserved words**), are always written in **lowercase**. Keywords are words that the C++ language uses for a special purpose.

asm	continue	float	new	signed	try
auto	default	for	operator	sizeof	typedef
break	delete	friend	private	static	union
case	do	goto	protected	struct	unsigned
catch	double	if	public	switch	virtual
char	else	inline	register	template	void
class	enum	int	return	this	volatile
const	extern	long	short	throw	while

Character

Each character in the computer is represented by **ASCII** (American Standard Code for Information Interchange). A total of **128** characters can be represented in ASCII.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Variable Declaration

Before a variable appears in a program, you must **declare** it. The general variable declaration is:

```
data type variables list;
```

Example 5:

```
main()
{
    int a,b;
    float c;
}
```

Variable Assignment

A value can be assigned to the variable by placing an equal sign (=) and a value for the variable. The general variable assignment is:

```
variable=value;
```

Example 6:

```
#include<iostream>
using namespace std;
main()
{
    int a;
    a=67;
    cout<<"Venus is "<<a<<" million miles from the Sun.";
}
```

Output:

```
Venus is 67 million miles from the Sun.
```

Initialization

A variable can be **initialized**, i.e., a value can be **assigned** to the variable, during its definition. Initialization is achieved by placing an **equal** sign (=) and an initial value for the variable immediately after the name of the variable.

Example 7:

```
main()
{
    int z=1;
    float x=2.5;
    char c='a';
}
```

Constant Objects

The **const** keyword is used to create a "**read-only**" object. As an object of this type is constant, it **cannot** be modified at a later stage and **must** be initialized during its definition.

Example 8:

```
main()
{
    const double PI=3.141593;
}
```

Example 9:

```
#include<iostream>
using namespace std;
main()
{
```

```
    int x;
    char y;
    float z;
    x=1;
    y='I';
    z=3.9;
    cout<<"The total current "<<y<<" in case "<<x<<" is "<<z<<" amps";
}
```

Output:

```
The total current I in case 1 is 3.9 amps
```

Output Function: printf()

The **printf()** function can have any number of arguments and prints them all on the screen. The general form is:

```
printf("format specifier",variables list);
```

Format Specifier

Format Specifier is used to specifying the types of variables you want to print. The general format used with the print function is listed below.

`%c`	Single Character
`%s`	String
`%d`	Signed decimal integer
`%f`	Floating point (decimal notation)
`%e`	Floating point (exponential notation)
`%g`	Floating point (shorter)
`%u`	Unsigned decimal integer
`%x`	Unsigned hexadecimal integer
`%o`	Unsigned octal integer
`%l`	Prefix used with others (long)

Example 10:

```
#include<iostream>
using namespace std;
main()
{
    int x;
    char y;
    float z;
    x=1;
    y='I';
    z=3.9;
    printf("The total current %c in case %d is %.1f amps",y,x,z);
}
```

Output Formatting (setw)

There are many applications in which we would like to have more control over the appearance of our output.

NOTE: To use the (**setw**) function, you must include the (**iomanip**) header file.

Example 11:

```
#include<iostream>
#include<iomanip>
using namespace std;
main()
{
    cout<<"*"<<setw(6)<<"*"<<endl;
```

```
    cout<<"*"<<setw(6)<<"*"<<endl;
    cout<<"*"<<setw(6)<<"*"<<endl;
    cout<<"*******"<<endl;
    cout<<"*"<<setw(6)<<"*"<<endl;
    cout<<"*"<<setw(6)<<"*"<<endl;
    cout<<"*"<<setw(6)<<"*"<<endl;
}
```

Output:

```
*      *
*      *
*      *
*******
*      *
*      *
*      *
```

Escape Sequences

Sequence	Character Representation	Sequence	Character Representation
\a	alert (bell) character	\v	vertical tab
\b	backspace	\\	backslash
\n	new line	\?	question mark
\r	carriage return	\'	single-quote
\t	horizontal tab	\"	double quote

Example 12:

```
#include<iostream>
using namespace std;
main()
{
```

```
    cout<<"*\t\b\b*\n";
    cout<<"*\t\b\b*\n";
    cout<<"*\t\b\b*\n";
    cout<<"*******\n";
    cout<<"*\t\b\b*\n";
    cout<<"*\t\b\b*\n";
    cout<<"*\t\b\b*\n";
}
```

Input Function (cin)

This function is used for interactive input.

Example 13:

```
#include<iostream>
using namespace std;
main()
{
    char a;
    int b;
    cin>>a>>b;
    cout<<"You typed the letter "<<a<<" & number "<<b;
}
```

Output:

```
a
1
You typed the letter a & number 1
```

Example 14: Write a C++ program that accepts **unsigned integers** in decimal, and print it in all three notations (**decimal**, **octal**, and **hexadecimal**).

Solution:

```
#include<iostream>
using namespace std;
main()
{
    int n;
    cout<<"n = ";
    cin>>n;
    printf("decimal : %d\n",n);
    printf("  octal : %o\n",n);
    printf("   hexa : %x\n",n);
}
```

Output:

```
n = 10
decimal : 10
  octal : 12
   hexa : a
```

Colored Background & Colored Text

In C++ programming, the background of the output screen is black and the text color is white. We can color both the background and text color in the output screen in the following way

```
system("color XY");
```

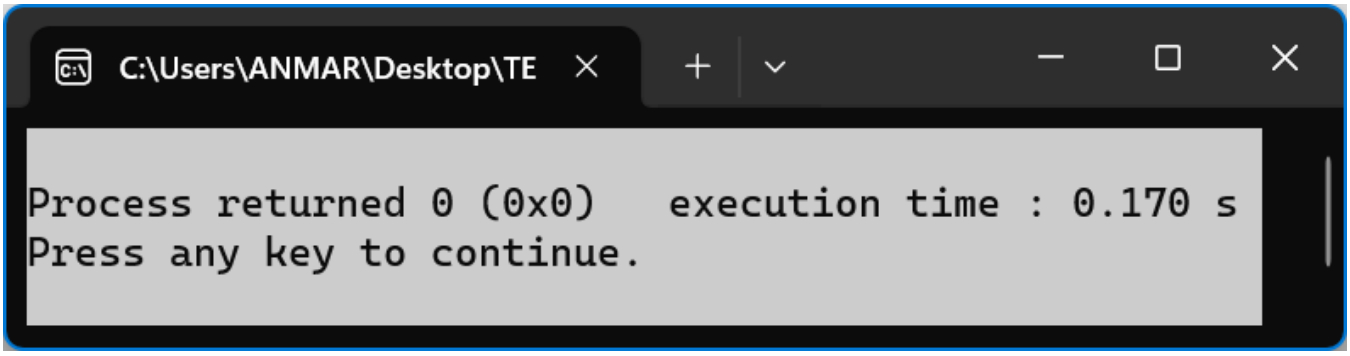
In the above command to change the background color change the value (**X**) of the above syntax to the corresponding color you want and to change the text color change the value (**Y**) of the above syntax to the corresponding color you want.

Color	ID	Color	ID
Black	0	Gray	8
Blue	1	Light Blue	9
Green	2	Light Green	A
Aqua	3	Light Aqua	B
Red	4	Light Red	C
Purple	5	Light Purple	D
Yellow	6	Light Yellow	E
White	7	Bright White	F

Example 15:

```
#include<iostream>

main()

{

  system("color 70");

}
```

Output:



System Pause & System Clear

In C++ programming, we can **pause** a program during execution. This will allow the user to input or read data. The pause approach makes the system more readable and user-friendly by allowing the user to read the instructions before performing any task. Also, improve the user interface by **clearing** the console screen and presenting a clean slate for new output.

```
system("pause");
```

```
system("cls");
```

Example 16:

```
#include<iostream>
using namespace std;
main()
{
    cout<<"Ministry of Higher Education & Scientific Research";
    cout<<endl;
    system("pause");
    cout<<"University of Baghdad";
    cout<<endl;
    system("pause");
    cout<<"College of Engineering";
    cout<<endl;
    system("pause");
    cout<<"Department of Electrical Engineering";
    cout<<endl;
    system("pause");
    cout<<"First Class";
    cout<<endl;
    system("pause");
    cout<<"Computer Programming";
    cout<<endl;
    system("pause");
    cout<<"C++ Language";
    cout<<endl;
    system("pause");
}
```

Output:

```
Ministry of Higher Education & Scientific Research
Press any key to continue . . .
University of Baghdad
Press any key to continue . . .
College of Engineering
Press any key to continue . . .
Department of Electrical Engineering
Press any key to continue . . .
First Class
Press any key to continue . . .
Computer Programming
Press any key to continue . . .
C++ Language
Press any key to continue . . .
```

Example 17:

```
#include<iostream>
using namespace std;
main()
{
    printf("%c%c%c%c%c",65,78,77,65,82);
    printf("%c",32);
    printf("%c%c%c%c%c%c%c",75,72,65,87,87,65,77);
    printf("%c",32);
    printf("%c%c%c",65,76,73);
}
```