

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal  
New Scheme of Examination as per AICTE Flexible Curricula  
II Semester (Group A) Bachelor of Technology (B.Tech.)  
BT-205 Basic Computer Engineering

## Topic Covered

### **Introduction to C++: Program Structure, Variables, Data Types, Keywords, Operators, Precedence and Associativity**

#### **1. Introduction**

##### **What is C++**

C++ is a general purpose, case-sensitive, free-form programming language that supports object-oriented, procedural and generic programming.

##### **Object-Oriented Programming (OOPs)**

C++ supports the object-oriented programming, the four major pillar of object-oriented programming (OOPs) used in C++ are:

1. Data Encapsulation
2. Data Abstraction
3. Inheritance
4. Polymorphism

##### **Applications of C++**

By the help of C++ programming language, we can develop different types of secured and robust applications:

- Window application
- Client-Server application
- Device drivers
- Embedded firmware etc.

##### **C++ Program**

File Name: test.cpp

1. `#include <iostream.h>`

```
2. #include<conio.h>
3. void main() {
4.     clrscr();
5.     cout << "Welcome to C++ Programming.";
6.     getch();
7. }
```

We can analyse the above program as following points:-

1. **#include<iostream.h>** includes the **standard input output** library functions. It provides **cin** and **cout** methods for reading from input and writing to output respectively.
2. **#include <conio.h>** includes the **console input output** library functions. The **getch()** function is defined in **conio.h** file.
3. **void main()** The **main()** function is the **entry point of every program** in C++ language. The **void** keyword specifies that it returns no value.
4. **cout << "Welcome to C++ Programming."** is used to **print the data "Welcome to C++ Programming."** on the console.
5. **getch()** The **getch()** function **asks for a single character**. Until you press any key, it blocks the screen.

## 2. Working with C++ Variable

A variable is a name of memory location. It is used to store data. Its value can be changed and it can be reused many times.

It is a way to represent memory location through symbol so that it can be easily identified.

Let's see the syntax to declare a variable:

```
type variable_name;
```

The example of declaring variable is given below:

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

Here, a, b, c are variables and int, float, char are data types.

### 2.1 Rules for defining variables

1. A variable can have alphabets, digits and underscore.
2. A variable name can start with alphabet and underscore only. It can't start with digit.
3. No white space is allowed within variable name.
4. A variable name must not be any reserved word or keyword e.g. char, float etc.

Valid variable names:

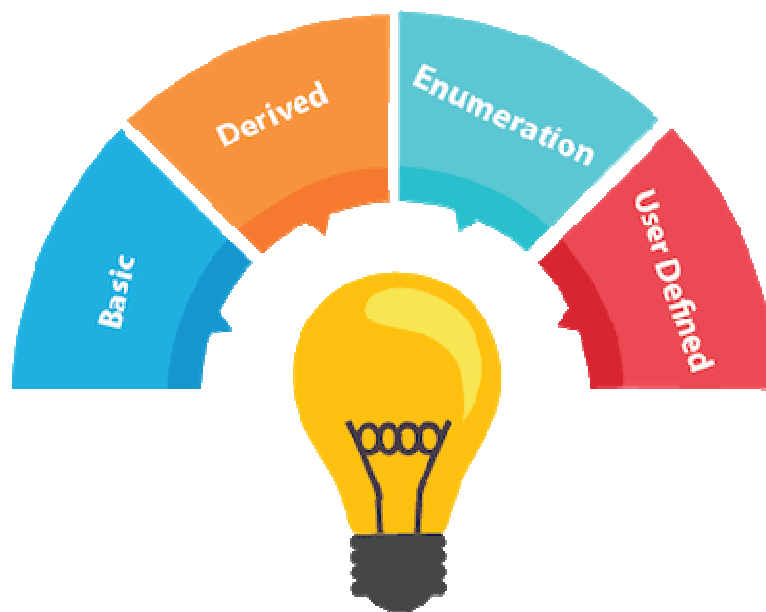
```
int a;  
int _ab;  
int a30;
```

Invalid variable names:

```
int 4;  
int a b;  
int double;
```

### 3. Working with C++ Data Types

A data type specifies the type of data that a variable can store such as integer, floating, character etc.



## Data Types in C++

There are 4 types of data types in C++ language.

Types	Data Types
Basic Data Type	int, char, float, double, etc
Derived Data Type	array, pointer, etc
Enumeration Data Type	enum
User Defined Data Type	structure, class

The memory size of basic data types may change according to 32 or 64 bit operating system.

Let's see the basic data types. Its size is given according to 32 bit OS.

Data Types	Memory Size	Range
char	1 byte	-128 to 127
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 127
short	2 byte	-32,768 to 32,767
signed short	2 byte	-32,768 to 32,767
unsigned short	2 byte	0 to 32,767
int	2 byte	-32,768 to 32,767
signed int	2 byte	-32,768 to 32,767
unsigned int	2 byte	0 to 32,767
short int	2 byte	-32,768 to 32,767
signed short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 32,767
long int	4 byte	
signed long int	4 byte	
unsigned long int	4 byte	
float	4 byte	
double	8 byte	
long double	10 byte	

#### 4. Working with C++ Keywords

A keyword is a reserved word. You cannot use it as a variable name, constant name etc. A list of 32 Keywords in C++ Language which are also available in C language are given below.

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

A list of 30 Keywords in C++ Language which are not available in C language are given below.

asm	dynamic_cast	namespace	reinterpret_cast	bool
explicit	new	static_cast	false	catch
operator	template	friend	private	class
this	inline	public	throw	const_cast
delete	mutable	protected	true	try
typeid	typename	using	virtual	wchar_t

## 5. Working with C++ Operators

An operator is simply a symbol that is used to perform operations. There can be many types of operations like arithmetic, logical, bitwise etc.

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operator
- Unary operator
- Ternary or Conditional Operator
- Miscellaneous Operator

	Operator	Type
<b>Binary Operator</b>	+, -, *, /, %	Arithmetic Operators
	<, <=, >, >=, ==, !=	Relational Operators
	&&,   , !	Logical Operators
	&,  , <<, >>, ~, ^	Bitwise Operators
	=, +=, -=, *=, /=, %=	Assignment Operators
<b>Unary Operator</b>	→ ++, --	Unary Operator
<b>Ternary Operator</b>	→ ?:	Ternary or Conditional Operator

## 6. Working with Precedence of Operators in C++

The precedence of operator species that which operator will be evaluated first and next. The associativity specifies the operators' direction to be evaluated, it may be left to right or right to left.

Let's understand the precedence by the example given below:

```
int x=5+10*10;
```

The "x" variable will contain 105 because \* (multiplicative operator) is evaluated before + (additive operator).

The precedence and associativity of C++ operators is given below:

Category	Operator	Associativity
Postfix	() [] -> . ++ --	Left to right
Unary	+ - ! ~ ++ -- (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Right to left
Shift	<<>>	Left to right
Relational	<<=>>=	Left to right
Equality	= = !=>	Right to left
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR		Right to left
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=>>= <<= &= ^=  =	Right to left
Comma	,	Left to right

### Assignment Questions

1. Explain various data types in C++?
2. Explain various operators in C++? Also discuss precedence and associativity of operators.
3. What is object oriented programming? Write down its characteristics.

### Bibliography

- <https://www.javatpoint.com/cpp-tutorial>