



# RAMA UNIVERSITY

[www.ramauniversity.ac.in](http://www.ramauniversity.ac.in)

## FACULTY OF ENGINEERING & TECHNOLOGY

### CSPS103: Object Oriented Programming

#### Lecture-05

Preeti Singh

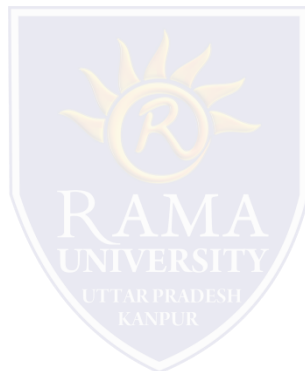
Department of Computer Science & Engineering  
Rama University, Kanpur

[preeti.ru@ramauniversity.ac.in](mailto:preeti.ru@ramauniversity.ac.in)

# OBJECTIVES

In this lecture, you will learn to:

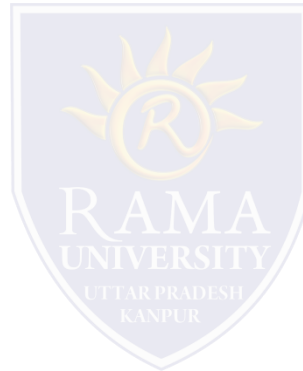
- ❖ Operators
- ❖ Arithmetical operators
- ❖ Relational operators
- ❖ Logical operators
- ❖ Assignment operators
- ❖ Conditional operators
- ❖ Comma operator



# OPERATORS

- ❑ An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations.
- ❑ C++ is rich in built-in operators.
- ❑ Generally, there are seven type of operators:

1. Arithmetical operators
2. Relational operators
3. Logical operators
4. Assignment operators
5. Conditional operators
6. Comma operator
7. Sizeof operators



# ARITHMETICAL OPERATORS

Arithmetical operators +, -, \*, /, and % are used to perform an arithmetic (numeric) operation

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus



- You can use the operators +, -, \*, and / with both integral and floating-point data types.
- Modulus or remainder % operator is used only with the integral data type.

# RELATIONAL OPERATORS

- ❑ The relational operators are used to test the relation between two values.
- ❑ All relational operators are binary operators and therefore require two operands.
- ❑ A relational expression returns zero when the relation is false and a non-zero when it is true.
- ❑ The following table shows the relational operators.



<b>Relational Operators</b>	<b>Meaning</b>
<	Less than
<=	Less than or equal to
=	Equal to
>	Greater than
>=	Greater than or equal to
!=	Not equal to

# LOGICAL OPERATORS

- ❑ The logical operators are used to combine one or more relational expressions.
- ❑ The logical operators are

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A    B) is true.
!	Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	!(A && B) is true.

# ASSIGNMENT OPERATOR

- ❑ The assignment operator '=' is used for assigning a variable to a value.
- ❑ This operator takes the expression on its right-hand-side and places it into the variable on its left-hand-side.

**For example:**

**m = 5;**

- ❑ The operator takes the expression on the right, 5, and stores it in the variable on the left, m.

**x = y = z = 32;**

**This code stores the value 32 in each of the three variables x, y, and z. In addition to standard assignment operator shown above, C++ also support compound assignment operators.**

## Compound Assignment Operators

Operator	Example	Equivalent to
<b>+=</b>	<b>A += 2</b>	<b>A = A + 2</b>
<b>-=</b>	<b>A -= 2</b>	<b>A = A - 2</b>
<b>%=</b>	<b>A %= 2</b>	<b>A = A % 2</b>
<b>/=</b>	<b>A /= 2</b>	<b>A = A / 2</b>
<b>*=</b>	<b>A *= 2</b>	<b>A = A * 2</b>

# INCREMENT AND DECREMENT OPERATORS

- C++ provides two special operators viz '+' and '-' for incrementing and decrementing the value of a variable by 1.
- The increment/decrement operator can be used with any type of variable but it cannot be used with any constant.
- Increment and decrement operators each have two forms, pre and post.

## The syntax of the increment operator is:

Pre-increment: ++variable  
Post-increment: variable++

## The syntax of the decrement operator is:

Pre-decrement: --variable  
Post-decrement: variable--

- In Prefix form first variable is first incremented/decremented, then evaluated
- In Postfix form first variable is first evaluated, then incremented / decremented.





# CONDITIONAL OPERATOR

- ❑ The conditional operator ?: is called ternary operator as it requires three operands.
- ❑ The format of the conditional operator is :

```
Conditional_ expression ? expression1 : expression2;
```



## Example

If the value of conditional expression is true then the expression1 is evaluated, otherwise expression2 is evaluated.

```
int a = 5, b = 6;  
big = (a > b) ? a : b;
```

The condition evaluates to false, therefore big gets the value from b and it becomes 6.

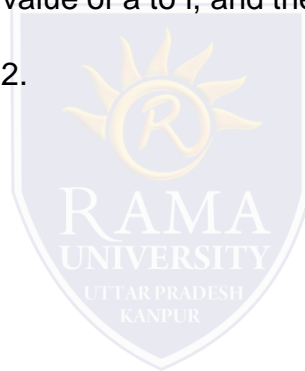
# THE COMMA OPERATOR

- ❑ The comma operator gives left to right evaluation of expressions.
- ❑ When the set of expressions has to be evaluated for a value, only the rightmost expression is considered.

```
int a = 1, b = 2, c = 3, i; // comma acts as separator, not as an operator
```

```
i = (a, b); // stores b into i would first assign the value of a to i, and then assign value of b to variable i.
```

- ❑ So, at the end, variable i would contain the value 2.



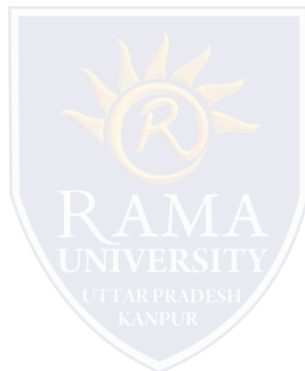
# THE SIZEOF OPERATOR

The sizeof operator can be used to find how many bytes are required for an object to store in memory.

## For example

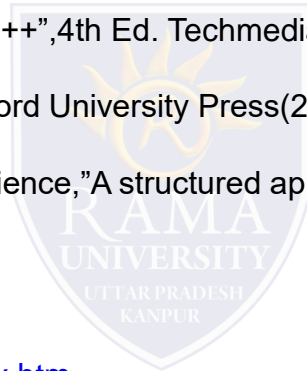
**sizeof (char) returns 1**

**sizeof (float) returns 4**



# REFERENCES

- Kernighan, Brian W., and Dennis M. Richie. The C Programming Language. Vol. 2. Englewood Cliffs: Prentice-Hall, 1988.
- King, Kim N., and Kim King. C programming: A Modern Approach. Norton, 1996.
- Bjarne Stroustrup, "C++ Programming language", 3rd edition, Pearson education Asia (1997)
- Lafore R. "Object oriented Programming in C++", 4th Ed. Techmedia, New Delhi (2002).
- Yashwant Kenetkar, "Let us C++", 1st Ed., Oxford University Press (2006)
- B.A. Forouzan and R.F. Gilberg, Compiler Science, "A structured approach using C++" Cengage Learning, New Delhi.
- <https://www.javatpoint.com/cpp-tutorial>
- <https://www.tutorialspoint.com/cplusplus/index.htm>
- [https://ambedkarcollegevasai.com/wp-content/uploads/2019/03/ CPP.pdf](https://ambedkarcollegevasai.com/wp-content/uploads/2019/03/_CPP.pdf)
- [https://onlinecourses.nptel.ac.in/noc20\\_cs07/unit?unit=3&lesson=19](https://onlinecourses.nptel.ac.in/noc20_cs07/unit?unit=3&lesson=19)



# MULTIPLE CHOICE QUESTION

## Multiple Choice Question:

**Q1. What is this operator called ?::?**

- a) conditional
- b) relational
- c) casting operator
- d) unrelational



# MULTIPLE CHOICE QUESTION

## Multiple Choice Question:

**Q2. Which operator is having the right to left associativity in the following?**

- a) Array subscripting
- b) Function call
- c) Addition and subtraction
- d) Type cast

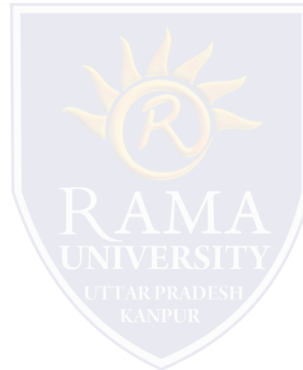


# MULTIPLE CHOICE QUESTION

## Multiple Choice Question:

**Q3. Which operator is having the highest precedence?**

- a) postfix
- b) unary
- c) shift
- d) equality



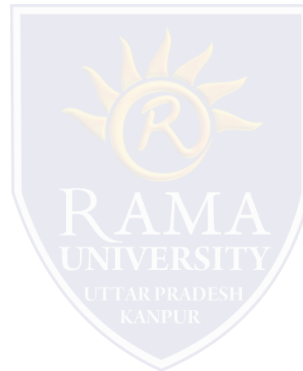
# MULTIPLE CHOICE QUESTION

## Multiple Choice Question:

Q4. What will be the output of the following C++ code?

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 5 + 3 * 5;
    cout << a;
    return 0;
}
```

- a) 35
- b) 20
- c) 25
- d) 30



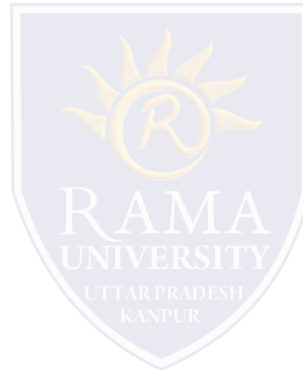


# MULTIPLE CHOICE QUESTION

## Multiple Choice Question:

**Q5. How many types of comments are there in c++?**

- a) 1
- b) 2
- c) 3
- d) 4



# Summary

## In this lecture, you learned that:

- Seven types of operators
  1. Arithmetical operators
  2. Relational operators
  3. Logical operators
  4. Assignment operators
  5. Conditional operators
  6. Comma operator
  7. Sizeof operators

