Unit 1: Introduction to C++

---

1. History of C++

C++ was developed by **Bjarne Stroustrup** in 1979 at AT&T Bell Labs. It is an extension of the C language with support for **Object-Oriented Programming (OOP)**. Initially called **C with Classes**, it was later renamed C++ (increment operator of C).

---

2. Advantages of C++

- Object-oriented language (supports class, object, inheritance)
- Fast execution (compiled language)
- Supports data abstraction and encapsulation
- Rich library support
- Platform independent

Reusable and maintainable code

---

3. Difference between C++ and Old Programming Languages

C++	Old Languages (C, BASIC)
Object-oriented	Procedural
Supports classes	No classes
Data security	Less secure
Code reusability	No reusability

4. Character Set

Character set includes:

- Alphabets (A–Z, a–z)
- Digits (0–9)
- Special symbols (+, -, *, /, @)
- White spaces (space, tab, newline

---

5. Tokens

Tokens are smallest elements of a C++ program.
Types:

- Keywords (int, float, if)
- Identifiers (sum, total)
- Constants (10, 3.14)
- Operators (+, -, *)
- Special symbols (;, {}, [])

---

6. Operators

Operators are symbols used to perform operations.
Types:

- Arithmetic (+, -, *, /, %)
- Relational (<, >, ==)
- Logical (&&, ||, !)
- Assignment (=, +=)
- Increment/Decrement (++ , --)

---

7. Operator Precedence and Associativity

Operator precedence decides order of execution.
Associativity decides direction (left to right / right to left).
Example: *, / have higher precedence than +, -

Unit 2: Control Structure and Array

1. If Statements

- if statement
- if-else
- if-else-if ladder
- nested if
- switch statement

---

2. Loops

Loops are used to repeat statements.
Types:

- while loop (entry controlled)
- for loop
- do-while loop (exit controlled)
- nested loop

---

3. Expressions and Qualifiers

Expression: combination of variables and operators.
Qualifiers modify variable behavior.
Examples: const, volatile

---

4. Arrays

Array is a collection of same type elements stored in continuous memory.
Types:

- One-dimensional array
- Multidimensional array

---

5. Strings

String is a collection of characters.
Types:

- String using char array
- Array of strings
  Common string functions:
- strlen()
- strcpy()
- strcat()
- strcmp()

Unit 3: Array, Structure and Union

1. Structure Declaration and Definition

Structure is a user-defined data type.

2. Use of Structure and Union

Structure stores different data types together.
Union stores different data types in same memory location.

3. Difference between Structure and Union

Structure	**Union**
Separate memory	Shared memory
More memory usage	Less memory usage

4. Function Components

Function has:

- Function declaration
- Function definition

- Function call

5. Parameter Passing

- Pass by value
- Pass by address
- Pass by reference

6. Inline, Recursive and Friend Function

- Inline function: expanded at compile time
- Recursive function: calls itself
- Friend function: can access private members

Unit 4: Pointers and Preprocessor Directives

1. Introduction to Pointers

Pointer stores address of another variable.

2. Uses of Pointers

- Dynamic memory allocation
- Function arguments
- Arrays and strings

3. Pointer to Function and Void Pointer

- Function pointer holds address of function
- Void pointer can store address of any data type

4. Memory Management Operators

- new: allocates memory
- delete: deallocates memory

5. Preprocessor Directives

- #define (macro definition)
- #include (header file inclusion)
- #error (generate error message)

Exam Tips

- Write definitions first
- Use tables for differences
- Draw syntax where required
- Keep answers point-wise