Tutorial 5 - C++ Basic Input/Output & Reserved Keywords

Key Concepts:

Input/Output Streams in C++:

- **Input Stream**:
  - Direction: Input device (e.g., keyboard) → Main memory.
  - Keyword: `cin`.
  - Operator: `>>` (Extraction operator).
- **Output Stream**:
  - Direction: Main memory → Output device (e.g., screen).
  - Keyword: `cout`.
  - Operator: `<<` (Insertion operator).

Practical Code Explanation:

```
#include <iostream>
using namespace std;
int main() {
    int num1, num2;
    cout << "Enter the value of num1:\n"; // Print message (Insertion operator)
    cin >> num1; // Take input from user (Extraction operator)
    cout << "Enter the value of num2:\n"; // Print message
    cin >> num2; // Take input from user
    cout << "The sum is " << num1 + num2; // Print sum of num1 and num2
    return 0;
}

```

Program Output:

1. **Execution 1**:
   Input: `num1 = 54`, `num2 = 4`.
   Output: `The sum is 58`.
2. **Execution 2**:
   Input: `num1 = 5`, `num2 = 8`.
   Output: `The sum is 13`.

Important Points:

1. **Insertion Operator**: `<<`
   - Used with `cout` to print messages or values to the screen.
2. **Extraction Operator**: `>>`
   - Used with `cin` to take input at runtime from the user.
3. **Keywords**:
   - `cout`: Print to the screen.
   - `cin`: Input values at runtime.

Reserved Keywords in C++:

- **Definition**: Keywords that are predefined by the language and cannot be used as variable names.
- Examples: `int`, `char`, `float`, `while`, `if`, `else`, `return`, etc.

Short Notes for Notebook:

Input/Output in C++:

1. `cin` → Input stream (takes input from user).
2. `cout` → Output stream (prints messages/values).
3. `>>` → Extraction operator (used with `cin`).
4. `<<` → Insertion operator (used with `cout`).

Example:

```
int num1, num2;
cin >> num1; // Take input
cin >> num2;
cout << "Sum is " << num1 + num2; // Output result

```

Reserved Keywords:

- Cannot be redefined or used as variable names.
- Examples:
  - Control: `if`, `else`, `while`, `for`, `switch`.
  - Data: `int`, `float`, `char`, `bool`.
  - Others: `return`, `void`, `namespace`.


alignas (since C++11)
alignof (since C++11)
and
and_eq
asm
atomic_cancel (TM TS)
atomic_commit (TM TS)
atomic_noexcept (TM TS)
auto(1)
bitand
bitor
bool
break
case
catch
char
char8_t (since C++20)
char16_t (since C++11)
char32_t (since C++11)
class(1)
compl
concept (since C++20)
const
consteval (since C++20)
constexpr (since C++11)
constinit (since C++20)
const_cast
continue
co_await (since C++20)
co_return (since C++20)
co_yield (since C++20)
decltype (since C++11)
default(1)
delete(1)
do
double
dynamic_cast
else
enum
explicit
export(1)(3)
extern(1)
false
float
for
friend
goto
if
inline(1)
int
long
mutable(1)
namespace
new
noexcept (since C++11)
not
not_eq
nullptr (since C++11)
operator
or
or_eq
private
protected
public
reflexpr (reflection TS)
register(2)
reinterpret_cast
requires (since C++20)
return
short
signed
sizeof(1)
static
static_assert (since C++11)
static_cast
struct(1)
switch
synchronized (TM TS)
template
this
thread_local (since C++11)
throw
true
try
typedef
typeid
typename
union
unsigned
using(1)
virtual
void
volatile
wchar_t
while
xor
xor_eq