Lecture 2 : Variables, data-Types & Operators

First C++ Program Structure (Boilerplate code) :

Every C++ program starts with this exact structure :

```
#include <iostream>      // Preprocessor directive
using namespace std;     // Uses standard namespace (std::cout → cout)

int main() {             // Starting point of execution
    cout << "Hello";     // Output statement
    return 0;           // Good practice - returns int value
}
```

Key Programming Rules :

- C++ is case sensitive — cout ≠ Cout
- semicolon (;) — Statement terminator like full stop in English
- Comments — //single line - ignored by compiler , green color in VS code.

Output (cout) Syntax :

```
cout << "Text";                  // Prints text as-is
cout << "Hello" << " World";      // Multiple items same line
cout << "Hello" << endl;          // New line (endline)
cout << "Hello\n";                // \n = newline (faster)
```

Without `endl` or `\n` → `%` symbol appears (trailing character)

VS code Setup & Compilation :

File Creation :

1. Create Folder → Open in VS Code
2. New File → `filename.cpp`

3. Ctrl + S to save

Compilation & Execution :

```
g++ filename.cpp     // Stage 1: Compile → creates a.exe (Windows) / a.out (M
ac) 【767.399, type: source】
./a.exe              // Stage 2: Execute
# Combined (recommended):
g++ filename.cpp -o filename && ./filename
```

Variables - Memory Containers :

Variable = Named memory location that can change value.

```
Syntax: datatype variableName = value;
int age = 25;          // Integer 【1609.08, type: source】
char grade = 'A';      // Single character (single quotes) 【1383.88, type: sourc
e】
```

Valid Variable Name :

- Start with letter or _ (NOT digits)
- `age , _age , myAge || 9age - x`

Printing Variables :

```
cout << age;     // Prints 25 (no quotes) 【1645.559, type: source】
cout << "age";   // Prints age (with quotes) 【1632.76, type: source】
```

Primitive Data Types & Memory :

Type	Size	Example	Stores	Notes
int	4 bytes → 32 bits	`int x = 25;`	whole number	no decimal
char	1 byte	`char c = 'A';`	single char	ASCII : 'A' = 65;
float	4 byte	`float pi = 3.14f;`	decimal	needs f suffix

double	8 byte	`double price = 99.99;`	decimal	default for decimal
bool	1 byte	`bool isSafe = true;`	true / false	prints 1 / 0

check size : `sizeof(variable)`

Type Casting :

Implicit Conversion (automatic - small → large)

```
char grade = 'A';
int value = grade;     // char(1B) → int(4B), value = 65
```

Explicit Conversion (Manual - Large → small)

```
double price = 99.99;
int newPrice = (int)price;  // 99.99 → 99 (decimal truncated) 【2466.64, type: source】
```

Input (cin) :

```
int age;
cout << "Enter age: ";
cin >> age;           // Stores user input in age 【2676.52, type: source】
cout << "Your age: " << age;
```

Arithmetic Operators :

Operator	Symbol	Example	Result
Addition	+	5 + 3	8
Subtraction	-	5 - 3	2
Multiplication	*	5 * 3	15
Division	/	5 / 3	2
Modulo	%	5 % 3	1

Logical Operator :

Operator	Symbol	OR Logic	AND Logic
less than	<	3 < 5	1 (true)
greater than	>	3 > 5	0 (false)
≤ , ≥	≤ , ≥		
equal	=	3 == 3	1
not equal	≠	3 ≠ 5	1

Logical Operator :

Operator	Symbol	OR Logic	AND Logic
OR	||	1 if any true	-
AND	&&	-	1 if all true
NOT	!	reverse true → false	

Unary Operators : (Increment / Decrement)

Type	Postfix	Prefix	Effect
Increment	a++	++a	a = a + 1
Decrement	a- -	- -a	a = a - 1

Complete Program Example :

```
#include <iostream>
using namespace std;

int main() {
    int a, b;
    cout << "Enter a: ";
    cin >> a;
    cout << "Enter b: ";
    cin >> b;
    int sum = a + b;
```

```
    cout << "Sum = " << sum << endl;
    return 0;
}
```