

❖ Data Types

1. **Long**: Represents long integers.

- Size: 4 bytes (32-bit) or 8 bytes (64-bit).
- Range: -2,147,483,648 to 2,147,483,647 (32-bit).
- Usage: Defining integer variables, mathematical calculations, data processing.

1. **Unsigned Long**: Represents unsigned long integers.

- Size: 4 bytes (32-bit) or 8 bytes (64-bit).
- Range: 0 to 4,294,967,295 (32-bit).
- Usage: Defining unsigned integer variables, data processing, system programming.

1. **Double**: Represents double-precision floating-point numbers.

- Size: 8 bytes (64-bit).
- Range: 1.7E-308 to 1.7E+308.
- Usage: Mathematical calculations, scientific data processing, engineering programming.

1. **Pointer**: Represents memory addresses.

- Size: 4 bytes (32-bit) or 8 bytes (64-bit).
- Usage: Direct memory manipulation, system programming, language development.

❖ Memory Allocation

- 32-bit: 4 bytes (Long, Unsigned Long, Pointer) and 8 bytes (Double).
- 64-bit: 8 bytes (Long, Unsigned Long, Pointer) and 8 bytes (Double).

❖ Example Usage

- Defining a long integer variable: `long x = 123456789;`
- Defining a double-precision floating-point variable: `double pi = 3.14159;`
- Defining a pointer to an integer: `int* ptr = &x;`