C++ -> MLP

1, a, A, @ → data types

C++ (1, A) → data type

integer data type → value

int a = 23; → variable name

1 -> int
-> char

23 (a)

int b; 4 bytes

bits => 0, 1
8 bit = 1 byte
32 bit = 4 bytes

Note: Garbage 4 bytes X

P → 16 → 2 bytes
M → 32 / 64 → 4 bytes

sizeOf( )

**1. Primitive / Built-in Data Types**

1) integer
int a = 23;
4 bytes -> CA

2) float
decimal
4 bytes

3) double
8 bytes

4) char
'a', 'A'
1 byte -> 8 bit
char d = 'a';

d (a)

5) bool
True -> 1
false -> 0
bool e = 1

e (1)

0, 1
1 bit
X
1 byte
7 ? 's bit

6) void → hold

**2. Derived Data Types**

int, char, float
↓
1 value

array ✓
pointer ✓
function ✓
reference ✓
} → X

**3. User-defined Data Types** ✓ OOP

nam = { }

struct, union, class, enum

Primitive
Derivate
user defined ✓

**Modifiers**

↳ int ↑ ↓

1) signed → (+ve, -ve)
↳default

int a = 23
int b = -23

Type - 4 bytes
(-2147483648 to 2147483647)

2) unsigned (+ve) → 0

unsigned int a = 23;
4 bytes
0 → 4294967295

3) short
↓
2 bytes
-32768 to 32767

short int a = 25;

4) long
Zyada

long int // 4 bytes (8 bytes)
long long int // 8 bytes ✓

**Typecasting** — int -> DT

1) Implicit typecasting (Automatic conversion)
↓
Type promotion

```
int a = 5;
float b = 2.5;
float c = a + b
```

a -> ASCII
97

-explore
ASCII

'a' b c d
↳ 97 98 99 100//

2) Explicit Typecasting (Manual Conversion)

(type) variableName;

```
float pi = 3.14;
int num = (int) pi
```

3//
float -> int
....

**Data Type Limits**

1) integer

maximum -> INT_MAX ✓
int -> 4 bytes = 32 bits

signed (+ve, -ve) ✓
Total = 32 bits

1 bit (sign)
31 bits value
Max => $2^{31}-1$ ✓

unsigned (+ve)
Max = $2^{32}-1$ ✓

minimum
INT_MIN
Signed = $-2^{-31}$
Unsigned = 0

2) Character
1 byte — 8 bits
$2^8$ = 256 values

Signed
1 bit
7 value
Max => $2^7-1$
Min = $-2^7$

Unsigned
Max = $2^8-1$ ✓ (256)
Min = 0
(0 to 255)
255//

$2^{17}$ -> output ?

int num = 123658 -> 4 bytes
char ch = num -> 1 bytes

? B