Fundamental Datatype
Integer (Part-2)

Video 07

If integer size is 4 bytes
- Short int (may be) 2 bytes
- Long int (may be) 8 bytes

} depends of machine

sizeof(short int) <= sizeof(int) <= sizeof(long int)
↳ change depends on machine

Modifiers: (Short, long, Signed, Unsigned) used for integers.

```
#include<stdio.h>
int main() {
    printf("%d", sizeof(long int));
    return 0;
}
```

→ 8 byte (depending on machine)

By default,

int variable_name;
↳ signed integer variable

Unsigned int variable_name;
↳ allows only positive value

```
#include<stdio.h>
#include<limits.h>
int main() {
    int var1 = INT_MIN;
    int var2 = INT_MAX;
    printf("Range of signed integer %d to %d", var1,
                                              var2);
    return 0;
}
```

→ Range of signed integer -2147483648 to 2147483647

```
#include <stdio.h>
#include <limits.h>
int main() {
    unsigned int var1 = 0;
    unsigned int var2 = UINT_MAX;
    Printf("Range from %d to %d", var1, var2);
    return 0;
}
```

to represent unsigned operator %u should be used.
%d won't print unsigned operator.

→ Range from 0 to 4294967295

```
#include<stdio.h>
#include<limits.h>
int main(){
    short int var1 = SHRT_MIN;
    short int var2 = SHRT_MAX;
    printf("Range from %d to %d", var1, var2);
    return 0;
}
```

2 byte ← short
and signed

→ Range from -32768 to 32767

```
#include<stdio.h>
#include<limits.h>
int main(){
    short unsigned int var1 = 0;
    short unsigned int var2 = USHRT_MAX;
    printf("Range from 0 to %u", var2);
    return 0;
}
```

order doesn't matter ←

→ Range from 0 to 65535

```
#include<stdio.h>
#include<limits.h>
int main() {
    long int var1 = LONG_MIN;
    long int var2 = LONG_MAX;
    long unsigned int var3 = 0;
    long unsigned int var4 = ULONG_MAX;
    printf("For Signed Range: %d to %d", var1, var2);
    printf("For unsigned Range: %lu to %lu", var3, var4);
    return 0;
}
```

→ For Signed Range: -2147483648 to 2147483647
→ For Unsigned Range: 0 to 4294967295

Another one is (long long int)

if sizeof (long int) = 4 byte (32 bit.)
sizeof (long long int) = 8 byte (64 bit)

depends on machine

max limit 64 bit.

Summary:

(*) sizeof(short int) <= sizeof(int) <= sizeof(long int)

(*) signed int variable_name is equivalent to writting int variable_name. Which means by default variable is signed.

(*) %d used to print signed ~~operator~~ integer.

(*) %u " " " unsigned integer

(*) %ld " " " long integer

(*) %lu " " " long unsigned integer

(*) %lld " " " long long integer

(*) %llu " " " long long unsigned integer