CPP-06

★ Static variables and method

@ Static member variable of a class

→ Exist exactly once per class, not per object.

→ The value is equal across all instance

→ Must be defined in *.cpp files.

@ Static member function of a class

→ Can call without an object.

ClassName :: MethodName (<params>)

---

sizeof (<type>) ⟶ return size of the type. (in bytes)

★ RAM

→ It is the working memory.

→ It has linear addressing.

→ Every byte has a address usually Presented in hexadecimal form.

→ Any address can be accessed at random.

→ Pointer is a type to store memory addr.

* Pointer

⇒ <Type>* defines a pointer to type <TYPE>

⇒ Uninitialized pointer points to a random address

⇒ Always initialize pointers to an address or a nullptr

```
int* a = nullptr;
YourType* c = nullptr;
```

* Non-owning pointers

→ Memory pointed to by a raw pointer is not removed when pointer goes out of scope.

→ Pointers can either own memory or not

→ Owning memory mean being responsible for its cleanup.

→ Raw pointers should never own memory.

* Address operator for pointer.

→ Operator & returns the address of the variable in memory.

→ Example

```
int a = 45;
int* a_ptr = &a;
```

* Pointer to Pointer

Example

```
int a = 45;
int* a_ptr = &a;
int** a_ptr_ptr = &a_ptr;
```

* Pointer dereferencing

⇒ Operator * returns the value of the variable to which the pointer points.