```
  // [1 Per VkInstance]:- All of these options should basically be set by the one who is gonna
use amVK and create an APP/Software 🤷‍♀
VkApplicationInfo amVK_AppInfo = {
  .sType = VK_STRUCTURE_TYPE_APPLICATION_INFO,
    // [implicit valid usage]:- must     be <above_value>
  .pNext = nullptr,
    // [implicit valid usage]:- must     be NULL
  .pApplicationName = "amVK_ApplicationInfo.pApplicationName not given",
    // [implicit valid usage]:- must not be NULL
  .applicationVersion = VK_MAKE_API_VERSION(0, 0, 0, 0),
  .pEngineName = "amVK_ApplicationInfo.pEngineName not given",
    // [implicit valid usage]:- must not be NULL
  .engineVersion = VK_MAKE_API_VERSION(0, 0, 0, 0),
  .apiVersion = VK_API_VERSION_1_0
    // version of the Vulkan API against which the application expects to run on
};
```

```
VkInstanceCreateInfo amVK_InstanceCI = {
  .sType = VK_STRUCTURE_TYPE_INSTANCE_CREATE_INFO,
    // [implicit valid usage]:- must     be <above_value>
  .pNext = nullptr,
    // [implicit valid usage]:- must     be NULL
  .flags = 0,
    // [implicit valid usage]:- must     be 0
  .pApplicationInfo = &amVK_AppInfo,
    /**
     * This can help Vulkan implementations [hardware-vendors] to perform ad-hoc optimizations.
     *      ref:- https://paminerva.github.io/docs/LearnVulkan/01.A-Hello-Window
     * For our educational purposes,
     *      specifying this information is irrelevant �
     *      since we are not creating AAA games that hardware-vendors are aware of.
     *      However, it’s still good to know what the VkApplicationInfo structure is used for.
     */
  .enabledLayerCount = 0,
  .ppEnabledLayerNames = nullptr,
  .enabledExtensionCount = 0,
  .ppEnabledExtensionNames = nullptr
};
```

Many Vulkan structures include two common fields: `.sType` and `.pNext`.

1. `.sType` :- is an enumeration defining the type of the structure. It may seem somewhat redundant, but this information can be useful for the loader, layers, and implementations to know what type of structure was passed in through `.pNext`.
2. `.pNext` :-llows to create a linked list between structures. It is mostly used when dealing with extensions that expose new structures to provide additional information to the loader, layers, and implementations, which can use the `.sType` field to know the type of the elements in the linked list.