[**Task 2 (Structures)**](http://testconfluence.training.local/pages/viewpage.action?pageId=14942314)

[Skip to end of metadata](http://testconfluence.training.local/pages/viewpage.action?pageId=14942314#page-metadata-end)

* Created by [Andriy Korkuna](http://testconfluence.training.local/display/~akorkuna) on [Feb 26, 2020](http://testconfluence.training.local/pages/viewpreviousversions.action?pageId=14942314)

[Go to start of metadata](http://testconfluence.training.local/pages/viewpage.action?pageId=14942314#page-metadata-start)

**As a** developer  
**I want** to have a class that can store single values of [C++ fundamental type](http://en.cppreference.com/w/cpp/language/types) (except void and std::nullptr\_t) and a function/method performs type-safe access to the conteined value.  
**so that** I will be able use a single variable to store an *int*, then a *double*, then a *bool* etc.

**Acceptance Criteria:**

* The class should contain all type of constructors
* The class should contain all type of assignment operators
* The class should contain a method to destroy contained object
* The class should contain a method to swap two objects
* The class should contain a method returns the type of the contained value
* The function/method that extracts a value should provide type-safe access. If the requested type does not match to stored type exception should be thrown

**Usage example:**

...

AnyType anyType = 1;

anyType = true;

anyType = 1.7;

try

{

int storedValue = anyType.ToInt();

}

catch(exceptionType&)

{

//Bad cast exception

}

double storedValue = anyType.ToDouble();

...

See:  
[C++ Data Structures](https://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm)  
[C++ Classes and Objects](https://www.tutorialspoint.com/cplusplus/cpp_classes_objects.htm)  
[C - Unions](https://www.tutorialspoint.com/cprogramming/c_unions.htm)  
[C++ Overloading](https://www.tutorialspoint.com/cplusplus/cpp_overloading.htm)  
[C++ Exception Handling](https://www.tutorialspoint.com/cplusplus/cpp_exceptions_handling.htm)