Explicit Casts

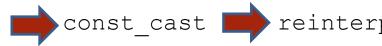
- C++ has four different cast operators
 - dynamic cast, static cast, const cast, and reinterpret cast
- Each cast has a special use-case.
- The syntax:

```
double myDouble{5.5};
int i = static cast<int>(myDouble);
```



```
C-Casts(int i= (int)myDouble; )
```

- should not be used.
- apply eventually a series of casts static_cast const_cast reinterpret_cast



dynamic cast

- Converts a pointer or a reference of a class to a pointer or a reference in the same inheritance hierarchy.
- Can only be used on polymorphic types.



- Allows to cast up, down, and cross the inheritance hierarchy
- Type information at run time is used to determine if the cast is valid.
- If the cast is not possible, you will get a nullptr in case of a pointer or an std::bad cast-exception in case of a reference.

static_cast

- Allows the conversion between related types
 - Pointer types in class hierarchies
 - Integral types into enumerations
 - Floating-point types into integrals types
- Difference to the dynamic_cast
 - Will be performed during compile time
 - Can not be applied to polymorphic types



const_cast

const_cast allows it to remove or add the const or volatile qualifier.



Modifying a const or volatile declared object by removing its constness is undefined behavior.



reinterpret cast

- reinterpret cast allows it to convert
 - a pointer type into another pointer.
 - an integral type into a pointer type and vice versa.

Guarantees that the conversion to and from a pointer returns the same value.