

Static Cast

This lesson highlights the key features of the `static_cast` operator.

WE'LL COVER THE FOLLOWING ^

- Features
- Example

Features

- `static_cast` is the simplest casting operator and is used for simple conversions.
- It can only perform all the conversions that are well-defined by the compiler. For example, a string to integer cast won't work.
- It allows bidirectional conversion between related data types such as:
 - pointer types in class hierarchies
 - integrals and floating-point numbers
 - integrals and enumerations
- `static_cast` cannot be used with polymorphic types.
- Unlike `dynamic_cast`, a `static_cast` is performed during compile time.

Example

```
#include <iostream>

class Account{};
class BankAccount: public Account{};

enum Color{
    red,
    blue,
    green
```



```

};

int main(){

    std::cout << std::endl;

    Account * a = nullptr;
    BankAccount * b = nullptr;

    a = static_cast<Account*> (b);           // upcast
    a = b;                                   // upcast

    b = static_cast<BankAccount*>(a);        // downcast

    int i{2};
    Color col = static_cast<Color>(i);
    std::cout << "i: " << i << std::endl;
    std::cout << "col: " << col << std::endl;

    int i2= static_cast<int>(3.14);
    std::cout << "i2: " << i2 << std::endl;

    std::cout << std::endl;

}

```



- In lines 19 and 22, we can see how `static_cast` supports up and down casting between pointers of the same class hierarchy.
- An integer can be cast into an enum state using `static_cast`. This is evident in line 25.
- A simple conversion from float to integer can be seen in line 29.

The next named cast operator on our list is `const_cast`.