

Slides

Development > Programming Languages > C++

## The C++ 20 Masterclass : From Fundamentals to Advanced

Learn and Master Modern C++ From Beginning to Advanced in Plain English : C++11, C++14, C++17, C++20 and More!

4.7 ★★★★★

Created by [Daniel Gakwaya](#)

# Section :Enums and type aliases

Slide intentionally left empty

# Enums And Type Aliases : Introduction

Month

Jan

May

Sep

Feb

June

Oct

March

July

Nov

April

Aug

Dec

```
unsigned long long int huge_num {18'446'744'073'709'551ull};
```

Slide intentionally left empty

# Enum Classes





## enum Declaration

```
enum class Month {  
    Jan , Feb, Mar, Apr ,  
    May, Jun, Jul, Aug,  
    Sep, Oct, Nov, Dec  
};
```

## Using the enum

```
enum class Month {  
    Jan , Feb, Mar, Apr ,  
    May, Jun, Jul, Aug,  
    Sep, Oct, Nov, Dec  
};  
  
Month month {Month::Jul};  
  
//std::cout << "month : " << month << std::endl; //Compiler Error  
std::cout << "month : " << static_cast<int>(month) << std::endl;
```



Each enumeration is represented by an integral value under the hood



Month

Jan (0)

May(4)

Sep(8)

Feb(1)

June(5)

Oct(9)

March(2)

July(6)

Nov(10)

April(3)

Aug(7)

Dec(11)

## Customize integral values

```
enum class Month {  
    Jan = 1, Feb, Mar, Apr ,  
    May, Jun, Jul, Aug = 18,  
    Sep, Oct, Nov, Dec  
};
```

Can even have negative integral values associated with an enumerator

## Multiple enumerators for the same value

```
enum class Month {  
    Jan=1 ,January =1, Feb=2,February =2, Mar,March=3, Apr ,  
    May, Jun, Jul, Aug,  
    Sep, Oct, Nov, Dec  
};
```

## Enumerators in terms of others

```
enum class Month {  
    Jan = 1 , January = Jan,  
    Feb , February = Feb,  
    Mar , March = Mar,  
    Apr , April = Apr,  
    May,  
    Jun, June = Jun,  
    Jul, July = Jul,  
    Aug, August = Aug,  
    Sep, September = Sep,  
    Oct, October = Oct,  
    Nov, November = Nov,  
    Dec, December = Dec  
};
```





Size of an enum

```
enum class Month {  
    Jan , Feb, Mar, Apr ,  
    May, Jun, Jul, Aug,  
    Sep, Oct, Nov, Dec  
};  
  
Month month {Month::Jul};  
  
std::cout << "sizeof month : " << sizeof(month) << std::endl;
```

int is the default type associated with enums in C++



int



0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1



A diagram consisting of a horizontal rectangle divided into two equal halves. The left half is light green and contains the text '4 bytes'. The right half is dark blue and contains the text '-2,147,483,648 ~ 2,147,483,647'. The background of the slide is a gradient of blue, with several white diagonal lines on the right side.

4 bytes

-2,147,483,648 ~ 2,147,483,647

## Compiler error

```
enum class Month {  
  
    Jan = -30 , January = Jan,  
    Feb , February = Feb,  
    Mar , March = Mar,  
    Apr , April = Apr,  
    May,  
    Jun, June = Jun,  
    Jul, July = Jul,  
    Aug, August = Aug,  
    Sep, September = Sep,  
    Oct, October = Oct,  
    Nov = 2'147'483'647, November = Nov,  
    Dec, December = Dec  
};
```

## Custom type

```
enum class Month : unsigned char {  
    Jan = 0 , January = Jan,  
    Feb , February = Feb,  
    Mar , March = Mar,  
    Apr , April = Apr,  
    May,  
    Jun, June = Jun,  
    Jul, July = Jul,  
    Aug, August = Aug,  
    Sep, September = Sep,  
    Oct, October = Oct,  
    Nov, November = Nov,  
    Dec, December = Dec  
};
```

## Custom type

```
enum class Month : unsigned char {  
    Jan = 0 , January = Jan,  
    Feb , February = Feb,  
    Mar , March = Mar,  
    Apr , April = Apr,  
    May,  
    Jun, June = Jun,  
    Jul, July = Jul,  
    Aug, August = Aug,  
    Sep, September = Sep,  
    Oct, October = Oct,  
    Nov, November = Nov,  
    Dec, December = Dec  
};
```

Size is now 1 byte



## Compiler Error

```
enum class Month : unsigned char {  
    Jan = 0 , January = Jan,  
    Feb , February = Feb,  
    Mar , March = Mar,  
    Apr , April = Apr,  
    May,  
    Jun, June = Jun,  
    Jul, July = Jul,  
    Aug, August = Aug,  
    Sep, September = Sep,  
    Oct, October = Oct,  
    Nov = 300, November = Nov,  
    Dec, December = Dec  
};
```

Slide intentionally left empty



using enum

```

enum class Month {
    Jan , Feb, Mar, Apr ,
    May, Jun, Jul, Aug,
    Sep, Oct, Nov, Dec
};

std::string_view moth_to_string(Month month){
    switch(month){
        case Month::Jan : return "January";
        case Month::Feb : return "February";
        case Month::Mar : return "March";
        case Month::Apr : return "April";
        case Month::May : return "May";
        case Month::Jun : return "June";
        case Month::Jul : return "July";
        case Month::Aug : return "Aug";
        case Month::Sep : return "Sep";
        case Month::Oct : return "Oct";
        default : return "None";
    }
}

```

```

enum class Month {
    Jan , Feb, Mar, Apr ,
    May, Jun, Jul, Aug,
    Sep, Oct, Nov, Dec
};

std::string_view moth_to_string(Month month){
    switch(month){
        using enum Month;
        case Jan : return "January";
        case Feb : return "February";
        case Mar : return "March";
        case Apr : return "April";
        case May : return "May";
        case Jun : return "June";
        case Jul : return "July";
        case Aug : return "August";
        case Sep : return "September";
        case Oct : return "October";
        case Nov : return "November";
        case Dec : return "December";
    }
}

```

GCC 10	NO
GCC 11	YES
MSVC2019	YES

Slide intentionally left empty

# Old Enums



```
enum Direction {
    TopLeft,
    TopRight,
    Center,
    BottomLeft,
    BottomRight
};

enum Tool {
    Pen, Ikaramu = Pen,
    Marker,
    Eraser,
    Rectangle,
    Circle,
    PaintBucket
};
```

Legacy enums implicitly transform to int, which is good for `std::cout` , but that introduces the problem that we can compare different enum types which is very confusing.

## Legacy Enum Problems

```
enum Direction {
    TopLeft, TopRight, Center, BottomLeft, BottomRight
};

enum Tool {
    Pen, Marker, Eraser, Rectangle, Circle, PaintBucket
};

Direction direction{BottomLeft};
Tool tool {Pen};

std::cout << std::boolalpha;
std::cout << "tool > direction : " << (tool > direction) << std::endl;

std::cout << "direction : " << direction << std::endl;
std::cout << "Tool : " << tool << std::endl;
```

Slide intentionally left empty

# Type Aliases

```
unsigned long long int huge_num {18'446'744'073'709'551ull};
```

## Type alias

```
using HugeInt = unsigned long long int;  
HugeInt huge_num {18'446'744'073'709'551ull};  
  
std::cout << "huge_num : " << huge_num << std::endl;  
  
HugeInt other_huge_num {18'112'334'073'709'752ull};  
  
std::cout << "other_huge_num : " << other_huge_num << std::endl;
```

## typedef

```
//typedef (older) syntax : Not recommended in new code  
typedef unsigned long long int HugeInt;  
HugeInt huge_num {18'446'744'073'709'551ull};  
std::cout << "huge_num : " << huge_num << std::endl;
```



Slide intentionally left empty

# Enums And Type Aliases : Summary

Month

Jan

May

Sep

Feb

June

Oct

March

July

Nov

April

Aug

Dec

## Using the enum

```
enum class Month {  
    Jan , Feb, Mar, Apr ,  
    May, Jun, Jul, Aug,  
    Sep, Oct, Nov, Dec  
};  
  
Month month {Month::Jul};  
  
//std::cout << "month : " << month << std::endl; //Compiler Error  
std::cout << "month : " << static_cast<int>(month) << std::endl;
```

```
unsigned long long int huge_num {18'446'744'073'709'551ull};
```

## Type alias

```
using HugeInt = unsigned long long int;  
HugeInt huge_num {18'446'744'073'709'551ull};  
  
std::cout << "huge_num : " << huge_num << std::endl;  
  
HugeInt other_huge_num {18'112'334'073'709'752ull};  
  
std::cout << "other_huge_num : " << other_huge_num << std::endl;
```

Slide intentionally left empty