

Constexpr and consteval classes



```
/*
 * Topics:
 *   . Creating constexpr friendly types
 *   . The requirement is to have a constexpr constructor.
 */

*/
```

Constexpr and consteval classes



```
/*
```

- . The requirement for a class to be `constexpr` is that it provides:
 - . A `constexpr` constructor
- . `Constexpr` or `consteval` classes can have non `constexpr` member functions.
 - . see the `print_time()` function in the `Time24` class.
- . A `constexpr` function member function can't call a non-`constexpr` function.
 - . `fmt::println` is not `constexpr` in the `fmt` library.
 - . We can't call it from the `print_time()` function.
- . The definitions of `constexpr` or `consteval` functions must be available in the module interface file. If you put it in the implementation file, you may face issues.

```
*/
```

Constexpr and consteval classes

```
// This class is constexpr
export class Time24 {
    public:
        // Constructor
        constexpr Time24(int h = 0, int m = 0) : hour(h), minute(m) {
            validate_time();
        }

        // Getter functions
        constexpr int get_hour() const { return hour; }
        constexpr int get_minute() const { return minute; }

        // Add hours
        constexpr void add_hours(int h) {
            hour = (hour + h) % 24;
        }

        void print_time() const {
            fmt::print("{:02}:{:02}\n", hour, minute);
        }

    private:
        int hour;    // 0-23
        int minute; // 0-59
};
```

Constexpr and consteval classes

```
// This class is consteval
export class Time24 {
    public:
        // Constructor
        consteval Time24(int h = 0, int m = 0) : hour(h), minute(m) {
            validate_time();
        }

        // Getter functions
        consteval int get_hour() const { return hour; }
        consteval int get_minute() const { return minute; }

        // Add hours
        consteval void add_hours(int h) {
            hour = (hour + h) % 24;
        }

        void print_time() const {
            fmt::print("{:02}:{:02}\n", hour, minute);
        }

    private:
        int hour;    // 0-23
        int minute; // 0-59
};
```

Constexpr and consteval classes: Using objects



```
it_1::Time24 t1{12, 30};      // Run-time object
fmt::print("Time: {:02}:{:02}\n", t1.get_hour(), t1.get_minute());
t1.add_minutes(45);
fmt::print("Time: {:02}:{:02}\n", t1.get_hour(), t1.get_minute());
t1.add_hours(2);
fmt::print("Time: {:02}:{:02}\n", t1.get_hour(), t1.get_minute());

constexpr it_1::Time24 t2{23, 59}; // Compile-time object
fmt::print("Time: {:02}:{:02}\n", t2.get_hour(), t2.get_minute());
//t2.add_minutes(2); // This will not compile. constexpr implies const

int scores [t2.get_hour()] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

//t3 is from a consteval class
it_2::Time24 t3{12, 30};      // Run-time object
//t3.add_minutes(45); // Error. consteval class can't be used to create run time objects, modifiable objects.
```