C++1* Tech Talks

Compile time constants and const expressions

Hannes Hauswedell
and Wikipedia, and Stackoverflow...

REINERT LAB
GENOMICS, PROTEOMICS, ALGORITHMS

Max Planck Institute
for Molecular Genetics

Freie Universität Berlin

October 14, 2016

Compile time constant values

```
// ugly c-style macro:
#define CSTYLE_SIZE 10

static const int size = 9;

enum SENUM
{
    SIZE = 12
};
```

Value Metafunctions

```
template <typename T>
struct BitsPerValue
{
    static const uint8_t VALUE = 8;
};

template <>
struct BitsPerValue<Dna>
{
    static const uint8_t VALUE = 2;
};

template <>
struct BitsPerValue<Dna5>
{
    static const uint8_t VALUE = 3;
};

// ...
int foo = BitsPerValue<MyType>::VALUE;
```

Value Metafunctions

```
template <unsigned N>
struct Fibonacci
{
    enum { VALUE = Fibonacci<N-1>::VALUE + Fibonacci<N-2>::VALUE };
};

template <>
struct Fibonacci<1>
{
    enum { VALUE = 1 };
};

template <>
struct Fibonacci<0>
{
    enum { VALUE = 0 };
};

// ...
int foo = Fibonacci<7>::VALUE;
```

constexpr values

```
// ugly c-style macro:
#define CSTYLE_SIZE 10

static const int size = 9;

enum SENUM
{
    SIZE = 12
};

static constexpr int scsize = 13;
       constexpr int  csize = 13;
```

Value Metafunctsion

Value Metafunction (old style)

```
template <typename T>
struct BitsPerValue
{
    static const uint8_t VALUE = 8;
};

template <>
struct BitsPerValue<Dna>
{
    static const uint8_t VALUE = 2;
};
```

Constexpr Variable Template

```
template <typename T>
constexpr uint8_t bitsPerValue = 8;



template <>
constexpr uint8_t bitsPerValue<Dna> = 2;

// ...
```

Constexpr (meta)function

```
template <typename T>
constexpr uint8_t bitsPerValue(T   const &) { return 8; }

constexpr uint8_t bitsPerValue(Dna const &) { return 2; }
```

Value Metafunctsion

Value Metafunction (old style)

```
template <typename T>
struct BitsPerValue
{
    static const uint8_t VALUE = 8;
};

template <>
struct BitsPerValue<Dna>
{
    static const uint8_t VALUE = 2;
};
```

Constexpr Variable Template

```
template <typename T>
constexpr uint8_t bitsPerValue = 8;



template <>
constexpr uint8_t bitsPerValue<Dna> = 2;

// ...
```

Constexpr (meta)function

```
template <typename T>
constexpr uint8_t bitsPerValue(T   const &) { return 8; }

constexpr uint8_t bitsPerValue(Dna const &) { return 2; }
```

constexpr functions

Please implement the fibonacci example as either

- variable template; *or*
- constexpr functions
- make the example program print the 92th fibonacci number
- how long does it take to run, how long to compile?

constexpr functions

Constant expression functions:

- not everything can be a constexpr function
- be careful with recursion depth
- constexpr are only required to run at compile time **if they are in a constexpr context!!!**

applications

...