New Mathematical Functions

With C++17 we get lots of new mathematical functions like gcd, lcm, clamp and other special ones.

WE'LL COVER THE FOLLOWING

- `gcd` and `lcm`
- `clamp(v, min, max)`

`gcd` and `lcm`

`std::gcd` and `std::lcm`, introduced in P0295R0, are declared in `<numeric>` header:

```
#include <iostream>
#include <numeric>  // for gcm, lcm

int main() {
    std::cout << std::gcd(24, 60) << ',';
    std::cout << std::lcm(15, 50) << '\n';
}
```

`clamp(v, min, max)`

Another useful function is `clamp(v, min, max)`, declared in `<algorithm>`, from P0025:

```
#include <iostream>
#include <algorithm>  // clamp

int main() {
  std::cout << std::clamp(300, 0, 255) << ',';
  std::cout << std::clamp(-10, 0, 255) << '\n';
}
```

And what's more, here are newly available special functions, defined in the `<cmath>` header.

Function	Description
`assoc_laguerre`	Functions compute the associated Laguerre polynomials of their respective arguments n, m, and x
`assoc_legendre`	Functions compute the associated Legendre functions of their respective arguments l, m, and x
`beta`	Functions Compute the beta function of their respective arguments x and y
`comp_ellint_1`	Complete elliptic integral of the first kind of their respective arguments k
`comp_ellint_2`	Compute the complete elliptic integral of the second kind of their respective arguments k
`comp_ellint_3`	Compute the complete elliptic integral of the third kind of their respective arguments k and nu
`cyl_bessel_i`	Compute the regular modified cylindrical Bessel functions of their respective arguments nu and x
`cyl_bessel_j`	Compute the cylindrical Bessel functions of the first kind of their

`cyl_bessel_j`	functions of the first kind of their respective arguments nu and x
`cyl_bessel_k`	Compute the irregular modified cylindrical Bessel functions of their respective arguments nu and x
`cyl_neumann`	Compute the cylindrical Neumann functions, also known as the cylindrical Bessel functions of the second kind, of their respective arguments nu and x
`ellint_1`	Compute the incomplete elliptic integral of the first kind of their respective arguments k and phi (phi measured in radians)
`ellint_2`	Compute the incomplete elliptic integral of the second kind of their respective arguments k and phi (phi measured in radians)
`ellint_3`	Compute the incomplete elliptic integral of the third kind of their respective arguments k, nu, and phi (phi measured in radians)
`expint`	Compute the exponential integral of their respective arguments x
`hermite`	Compute the Hermite polynomials of their respective arguments n and x
`laguerre`	Compute the Laguerre polynomials of their respective arguments n and x

`legendre`	Compute the Legendre polynomials of their respective arguments l and x
`riemann_zeta`	Compute the Riemann zeta function of their respective arguments x
`sph_bessel`	Compute the spherical Bessel functions of the first kind of their respective arguments n and x
`sph_legendre`	Compute the spherical associated Legendre functions of their respective arguments l, m, and theta (theta measured in radians)
`sph_neumann`	Compute the spherical Neumann functions, also known as the spherical Bessel functions of the second kind, of their respective arguments n and x

> **Extra Info:** The above special functions were introduced in N1542 ver 3

Next up, we’ll learn about a new method of handling arrays introduced in C++ 17.