How Can It Simplify The Code

If you still aren't on board with the concept of inline variables, the example below will prove their importance.

A lot of header-only libraries can limit the number of hacks (like using inline functions or templates) and switch to using inline variables.

For example, take a look at the codes written below. Both serve the same purpose:

C++14 | C++17

```
class MyClass
{
  static inline int Seed(); // static method
};
inline int MyClass::Seed() {
  static const int seed = rand();
  return seed;
}
```

C++17 guarantees that `MyClass::seed` will have the same value (generated at runtime) across all the compilation units!

> *Extra Info:* The change was proposed in: P0386R2.

Now that we're done with inline variables, let's move on to the new features in lambda functions.