Type Invariants

The definition of `Time` is straightforward, but, it will cause problems. There are **certain restrictions** on what values members of a `Time` object may and may not have. Given our specification, for instance, `hours` must be between `0..23` and `minutes` must be between `0..59`. We call these the type's **invariants**.

But, with structures, we have **no means of enforcing those restrictions**. There is **nothing to prevent** someone, (most likely you, if you aren't careful), from constructing a bogus `Time` object like this:

```
Time bed_time = {27, 95};
```

Both values supplied here makes the `Time` object, `bed_time`, **invalid**. But, the code compiles fine; everything is perfectly legal C++, and the compiler has no idea that something bad might happen in the future.