Fixing Compiler Errors: Cheat Sheet

1. Which part of the pipeline causes the error?

2. Are functions overloaded?   Use wrapper lambda

3. Do functions accept the correct type?

4. Do functions return the correct type?

5. For ranges::views

   a) Too many / too few calls to join?

   b) const views?

6. For optional / expected

   a) Mixed up and_then with transform?

```cpp
// Split the pipeline
auto s1 = projects |
views::transform(getFilesInProject);
auto s2 = s1 | views::join;
auto s3 = s2 | views::transform(compile);
```

```cpp
// Make return types explicit
auto f1 = [](auto radius)
    { return calcArea(radius); };

auto f2 = [](double radius) -> double
    { return calcArea(radius); };
```

Function returns Thing
You expected vector<Thing> (or vice versa)

Will lead to weird compiler errors. Just Don't.

Function returns Thing
You expected optional<Thing> (or vice versa)