Data Flow Checklists

Consider the `string::getline(in, str)` function:

- `in` is an **input-output** parameter. The function depends on the stream's initial state (formatting, etc.) and it is changed by calling the function (setting the error value).
- `str` is an **output only** parameter; it makes no difference what is inside `str` when you call the function—data **only flows out**.

The Java concept of data flow—parameters are input, return statements are output—is too simplistic for C++. In C++ (as in many other languages), parameters can be used as input, as output, or as a combination of both.

Use this checklist to determine the direction of data flow:

- ☐ Argument not modified by function: **input parameter**
- ☐ Argument modified, input value not used: **output** parameter
- ☐ Argument used and changed by function: **input-output** parameter

Use this checklist to determine **how to declare** the parameter variable:

- ☐ Output and Input-Output parameters: **by reference**
- ☐ Input primitive (built-in and enumerated) types: **by value**
- ☐ Input library and class types: by **const reference**

> ***Never*** *pass by value for class or library types.*