**Which direction is the data flowing?**

**What does the parameter have to look like to represent that (how do i represent an input parameter if its a primitive type? If it’s a library type?)**

**A Parameter Checklist**

* Is the parameter a primitive type?
  + Does the function change the argument?
    - Define parameter as a non-const reference
    - **void swap(int& lhs, int& rhs);**
  + No change? Use pass by value
* Object types? **Always** use references
  + Want change? Use **non-const** reference
  + No change? Use **const** reference

**MEMORIZE THIS CHECKLIST**

**Declaring Parameter Review (in last notes too)**

* Out, in-out parameters
  + **Always** pass by reference (& after type)
  + Both object and primitive types
  + @param[in, out] or @param[out]
* In parameters depend on type
  + Object parameters (including string)
    - Always pass by **const** reference
    - This is for the memory/speed efficiency
  + Primitive parameters pass by value

**Just what is a stream?**

* Streams are abstract data flows
  + Data flows into program from a **source**
  + Your program processes the data
  + Information flows out to a **sink**
* Sending info **to** a sink is called **writing**
* Getting info **from** a source is called **reading**

**Stream classes**

* Specialized classes do input/output
  + **Istream:** “knows” how to read information
  + **Ostream**: “knows” how to write information
* Automatically created **global** stream objects
  + **Cin, cout, cerr, clog** for regular **chars**
  + **Wcin, wcout, wcerr, wclog** for wide **chars**

**Simplest Stream Input**

* Unformatted single character
  + **while(cin.get(ch))**
* Line-oriented **string** input
  + **While (getline(cin, str))**
  + Must remember.. - - - -- .

**Unformatted character I/O**

Exam 3 solution:

//heres what the code should look like

//substr length 2 at the end of the str, find how many times

//it appears in the string

int result = -1;

result = 0;

sizt\_t len = str.length();

if (len > 3) {

string last = str.substr(len - 2);

string front = str.substr(0, len - 2);

for (size\_t i = 0, flen = front.length(); i < flen; i++) {

if (last == front.substr(i, 2))

result++;

}

}

// mirror image

//abXYZba --> ab

//xxYxx --> xxYxx

size\_t i = 0;

for (size\_t len = str.length(); i < len; i++) {

if (str.at(i) != str.at(len - 1 - i)) { //if the mirrored chars arent the same

break;

}

}

result = str.substr(0, i); //so simple.. so elegant...