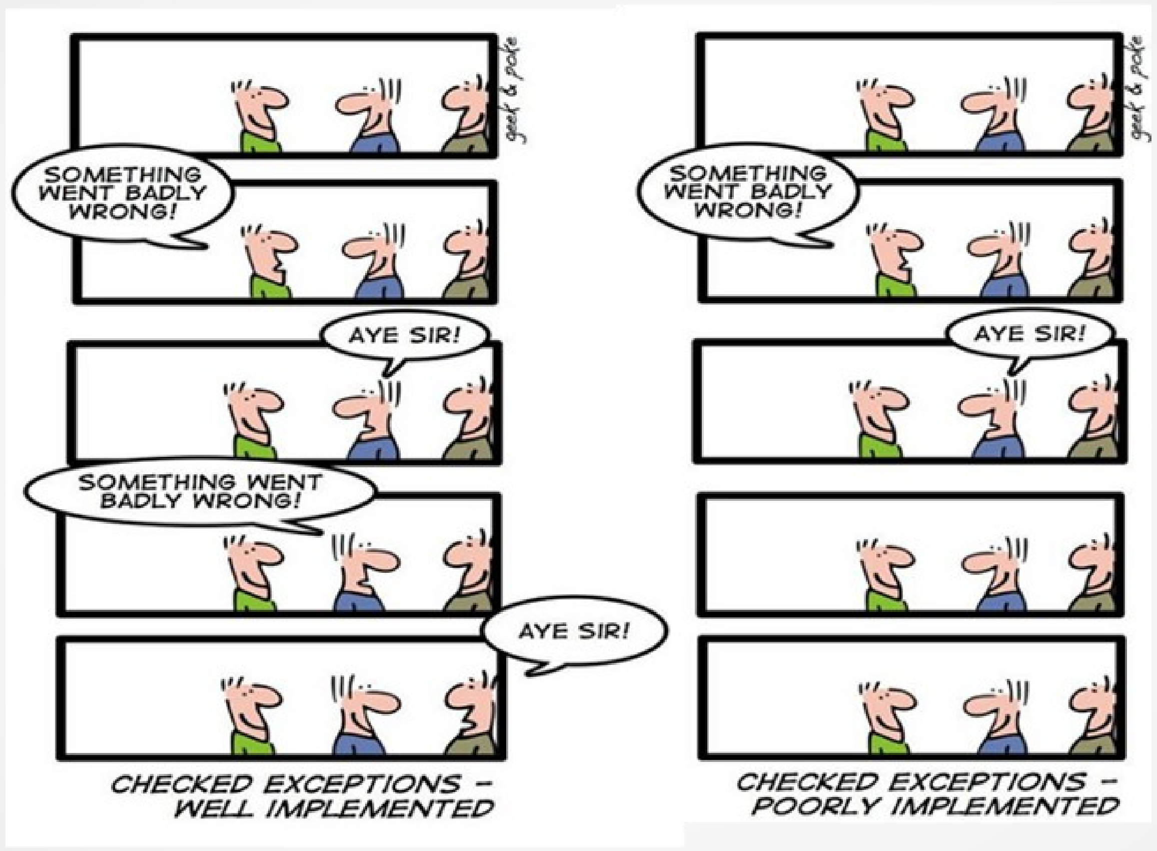
C++ Exception Handling


geek & poke
SOMETHING WENT BADLY WRONG!
AYE SIR!
SOMETHING WENT BADLY WRONG!
AYE SIR!
CHECKED EXCEPTIONS - WELL IMPLEMENTED
geek & poke
SOMETHING WENT BADLY WRONG!
AYE SIR!
CHECKED EXCEPTIONS - POORLY IMPLEMENTED

http://geekandpoke.typepad.com/geekandpoke/2009/06/simply-explained-checked-exceptions.html

Krishna Kumar

Range checking

```
int main () {
    std::vector<int> myvector (10);   // 10
    zero-initialized ints
    // assign some values:
    for (unsigned i=0; i<myvector.size(); i++)
        myvector[i]=i;
    std::cout << "myvector contains:";
    for (unsigned i=0; i<=myvector.size(); i++)
        std::cout << ' ' << myvector[i];
    std::cout << '\n';
    return 0;
}
```

- Output
  - My vector contains:
  - 0 1 2 3 4 5 6 7 8 9 0

Range checking ::at

```
int main () {
std::vector<int> myvector (10);   // 10
zero-initialized ints
 // assign some values:
for (unsigned i=0; i<myvector.size(); i++)
    myvector.at(i)=i;
std::cout << "myvector contains:";
for (unsigned i=0; i<=myvector.size(); i++)
    std::cout <<  myvector.at(i);
std::cout << '\n';
return 0;
}
```

- Output

```
terminating with uncaught exception of
type std::out_of_range: vector
```

  - My vector contains:
  - 0 1 2 3 4 5 6 7 8 9 Abort trap

Traditional Error Handling

- Special class to handle error

```
struct Range_error {};
void f (int n) {
    if (n < 0 || max <n) throw Range_error{};
}
```

- **terminate a program:**

```
if (something_wrong) exit(1);
if (argc != 2) {
std::cerr << "Incorrect number of arguments" << std::endl;
std::cerr << "Usage: ./factory course  (C++/Java)" << std::endl;
std::exit(EXIT_FAILURE);
}
```

Traditional Error Handling (cont...)

- Return an error value:
  - int get_int(); // get next integer when no acceptable “error value is present”.
- Return a legal value and leave program in error state:

  double d = sqrt(-1.0)
- Call an error handler:
  - if (something_wrong) something_handler; // possibly continue

Exception Handling (Simple Example)

- Catching Exception:
- void f() {

```
    try {
        throw E {};
    }
    catch(H) {
        // when do we get here?
    }
}
```