CAAM 420 Daily Note

Xinyi Wang

Nov.4, 2013

1 Announcement

Project 5 Due Nov.11

2 Function Overloading

Different functions can't have same name in C while in C++, this can be achieved.
C: Functions identified by names; functions with same name are identified as same function.
C++: Functions identified by signatures: name+number and type of arguments. (look out for default arguments!)
!!Notice that C++ cannot distinguish functions with same name by their return types because we may never use the return value of a function.
For example:
gcc ex62_function_overloads.c conflicting types for 'foo'
g++ ex62_function_overloads.c cool!
What's happening:
gcc treats .c files as c source (gcc treats .cc files as c++ source)
g++ treats both .c and .cc (and .cpp...) files as c++ source

3 C++ Exceptions

```
foo{
```

```
    foo1{
       foo2{
           foo3{...}
           }
       }
   }
```

what if foo3 blows up?
We can use exit(1).
Now, we have an alternative:
foo3 throws to catch(...){...}

See new version of vecppbase.hh and ex49_vecpp_add_try.cc for examples.

Now, let's try it in a simple example stuff.cc:

```
#include "cppstd.hh"
#include "CAAM420Exception.hh"
int main(int argc, char** argv){
      try{
          if(argc!=2){
                 CAAM420Exception e;
                 e<<"are you sure\n";
                 throw e;
                 }
      cerr<<argc[1]<<endl;
      }
      catch(CAAM420Exception & e{
          e.write(cerr);
          cerr<<"if yes, enter\"y\"\n";
          string s;
          cin>>s;
          if(s=="y"){
                cerr<<"blah blah blah\n";
          }
          else{
                exit(1);
```

```
        }
        exit(0);
    }
}
```

CAAM420Exception.hh is a provided exception header file.

4 Namespace

Namespace is another level of identification; It gives additional piece of name to a function.
Look at ex67_namespace.hh for an example.
How to use:
main function in ex67_namespace.cc:

```
int main() {
using A::foo;
using B::foo;
A::foo a;
B::foo<int> b;
C::bar c;
a.what();
b.what(1);
c.what();
}
```

We have to specify the namespaces of functions with same name in order to call them in main function.