A simple event loop

- Audience: people willing to write a little C++ code to do analysis of GLAST data
- Objectives:
  - Syntax is easy to use, remember
  - Same code works with either ROOT or FITS files, extensible to other tabular representations
  - Uses setup defaults whenever possible – specify just a file name, let the system do the rest

Illustrate with an example from the test program

```
class Example {
public:

    Example( tuple::ITable & t,
                 std::string label,
                 std::string ra_name, std::string dec_name)
    {
        double ra, dec, ra_sum=0, dec_sum=0;

        t.selectElement(ra_name, ra);
        t.selectElement(dec_name, dec);

        for( tuple::Iterator it = t.begin(); it!=t.end(); ++it){
            ra_sum += ra; dec_sum+=dec;
         }
        std::cout << label<< t.size() << " entries, average ra, dec: "
            << ra_sum/t.size() << ", "<< dec_sum/t.size() << std::endl;
    }
};
```

Notes on the example

- The "selectElement" assigns a pointer to a named column or branch, exactly like ROOT's TTRee::SetBranchAddress(). An exception is thrown if the table does not have a column or branch with the given name.

- Creating the iterator with the begin() and the increment operator actually saves the selected elements.
  - Somewhat contrary to STL philosophy, better to have that happen with a dereference operator.

Example cont.

```
// setup and test the FITS implemetation with a local test file,
// extracted using the GSSC web interface
tuple::FitsTable ft(DC1_galcenter.fits" );
Example one(ft, "FITS file: ", "ra", "dec");

// now check root.
tuple::RootTable rt(DC1_galcenter.root");
Example two(rt, "ROOT file: ", "FT1Ra", "FT1Dec");
```

Notes:

- Constructors find the first table (FITS) or TTree (ROOT); one can specify an different HDU, or an alternate TTree by name.
- Numeric types are converted to double (KISS)

Concluding remarks

- How to get it: just check out the package "tuple".
- Suggestions (and help!) welcome: some needed upgrades:
  - Convert ROOT numeric formats (requires double now)
  - Add an interface to ascii tables
  - Perhaps use a "factory" to select the concrete ITable
  - Fill arrays (only scalars at the moment)