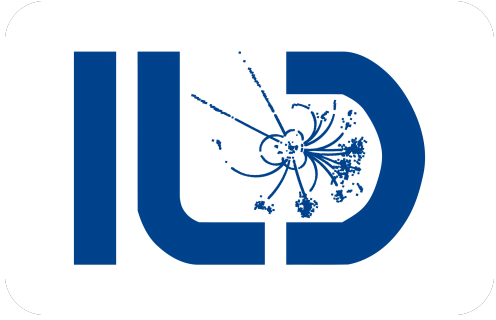
# Programming tips.

## Smart pointers

Rémi Ete

**DESY**

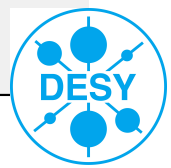
**November 5, 2018**

# Programming tips

## A typical ownership problem: who owns what ?

```
1  #include <Particle.h>
2
3  typedef Particle* ParticlePtr;
4
5  class Jet {
6  public:
7    Jet(const std::vector<ParticlePtr> &particles) :
8      m_particles(particles) {}
9    void print();
10 private:
11   std::vector<ParticlePtr> m_particles{};
12 };
13
14 int main() {
15   std::vector<ParticlePtr> particles;
16   particles.push_back(new Particle("electron"));
17   particles.push_back(new Particle("muon"));
18   Jet jet(particles);
19   jet.print();
20   return 0;
21 }
22
```



# Programming tips

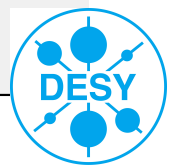
## Smart pointers

- `std::shared_ptr` and `std::unique_ptr`
- Owns a pointer for you
  - No new done by user
  - No delete done by user
  - Done internally
- std::shared_ptr
  - Owns a pointer for you
  - Internal pointer shared by many std::shared_ptr instances
  - delete called when no one owns it
- std::unique_ptr
  - Owns a pointer for you
  - Not copiable !
  - delete called at end of scope

# Programming tips

## A typical ownership problem: who owns what ?

```
1  #include <Particle.h>
2
3  typedef Particle* ParticlePtr;
4
5  class Jet {
6  public:
7    Jet(const std::vector<ParticlePtr> &particles) :
8      m_particles(particles) {}
9    void print();
10 private:
11   std::vector<ParticlePtr> m_particles{};
12 };
13
14 int main() {
15   std::vector<ParticlePtr> particles;
16   particles.push_back(new Particle("electron"));
17   particles.push_back(new Particle("muon"));
18   Jet jet(particles);
19   jet.print();
20   return 0;
21 }
22
```



# Programming tips

## `std::shared_ptr` to the rescue !

```
1  #include <memory>
2  #include <Particle.h>
3
4  typedef std::shared_ptr<Particle> ParticlePtr;
5
6  class Jet {
7  public:
8    Jet(const std::vector<ParticlePtr> &particles) :
9      m_particles(particles) {}
10   void print();
11 private:
12   std::vector<ParticlePtr> m_particles{};
13 };
14
15 int main() {
16   std::vector<ParticlePtr> particles;
17   particles.push_back(std::make_shared<Particle>("electron"));
18   particles.push_back(std::make_shared<Particle>("muon"));
19   Jet jet(particles);
20   jet.print();
21   return 0;
22 }
23
```

# Programming tips

## Some references

- Reference documentation:
  - `https://en.cppreference.com/w/cpp/memory/shared_ptr`
  - `http://www.cplusplus.com/reference/memory/shared_ptr`
- Videos
  - `https://www.youtube.com/watch?v=_Sk9JT_gTV4`
  - `https://www.youtube.com/watch?v=ENj37HvptgU`