# Programming tips.

## Smart pointers

Rémi Ete

DESY

November 5, 2018

# Programming tips

## A typical ownership problem: who owns what ?

```cpp
#include <Particle.h>

typedef Particle* ParticlePtr;

class Jet {
public:
  Jet(const std::vector<ParticlePtr> &particles) :
    m_particles(particles) {}
  void print();
private:
  std::vector<ParticlePtr> m_particles{};
};

int main() {
  std::vector<ParticlePtr> particles;
  particles.push_back(new Particle("electron"));
  particles.push_back(new Particle("muon"));
  Jet jet(particles);
  jet.print();
  return 0;
}

```

# 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 ?

```cpp
#include <Particle.h>

typedef Particle* ParticlePtr;

class Jet {
public:
  Jet(const std::vector<ParticlePtr> &particles) :
    m_particles(particles) {}
  void print();
private:
  std::vector<ParticlePtr> m_particles{};
};

int main() {
  std::vector<ParticlePtr> particles;
  particles.push_back(new Particle("electron"));
  particles.push_back(new Particle("muon"));
  Jet jet(particles);
  jet.print();
  return 0;
}

```

# Programming tips

## std::shared_ptr to the rescue !

```cpp
#include <memory>
#include <Particle.h>

typedef std::shared_ptr<Particle> ParticlePtr;

class Jet {
public:
  Jet(const std::vector<ParticlePtr> &particles) :
    m_particles(particles) {}
  void print();
private:
  std::vector<ParticlePtr> m_particles{};
};

int main() {
  std::vector<ParticlePtr> particles;
  particles.push_back(std::make_shared<Particle>("electron"));
  particles.push_back(std::make_shared<Particle>("muon"));
  Jet jet(particles);
  jet.print();
  return 0;
}

```

# 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`