```cpp
// Header file Particle.hpp - Luca Morelli 2021

#ifndef PARTICLE_HPP
#define PARTICLE_HPP

#include "ParticleType.hpp"
#include "ResonanceType.hpp"

#include <iostream>
#include <string>
#include <vector>

class Particle {
  // Data members
  // Static
  static std::vector<ParticleType *> particleType_;  // Types of particles
  static int NParticleType_;  // Number of Types of particles
  static constexpr int maxNumParticleType{
      10};  // Max number of Types of particles
  // Non static
  int index_;
  double Px_, Py_, Pz_;

  // Private member functions
  static int findParticle(std::string pName);
  void boost(double bx, double by, double bz);

 public:
  // Constructors
  Particle() : index_{-1}, Px_{0}, Py_{0}, Pz_{0} {}  // Default
  Particle(std::string name, double Px = 0, double Py = 0, double Pz = 0);

  // Public member functions
  // Static
  static void addParticleType(std::string name, double mass, int charge,
, double width = 0);
  static void printParticleTypes();
  // Non static
  int getIndex() const;                // Returns the index of the type
  void setParticle(int index);         // Sets the type by index
  void setParticle(std::string name);  // Sets the type by name
  void printDetails() const;
  double getPx() const;
  double getPy() const;
  double getPz() const;
  double getMass() const;
  int getCharge() const;
  double getEnergy() const;
  double invMass(Particle const &particle2) const;
  void setP(double Px, double Py, double Pz);
  int decay2body(Particle &dau1, Particle &dau2) const;
};

// Operator overload declaration
std::ostream &operator<<(std::ostream &os, Particle const &particle);
#endif
```