C++ code 2.14.22: Sub-problem (2-14.t): Implementation of CRFeSpace, ↪ GitLab

```cpp
class CRFeSpace : public lf::uscalfe::UniformScalarFESpace<double> {
 public:
  /** @brief no default constructors*/
  CRFeSpace() = delete;
  CRFeSpace(const CRFeSpace &) = delete;
  CRFeSpace(CRFeSpace &&) noexcept = default;
  CRFeSpace &operator=(const CRFeSpace &) = delete;
  CRFeSpace &operator=(CRFeSpace &&) noexcept = default;

  /** Main constructor that sets up the local-to-global index mapping
   * by calling the constructor if its base class
   * The parameter mesh_p is a shared pointer to underlying mesh (immutable) */
  explicit CRFeSpace(std::shared_ptr<const lf::mesh::Mesh> mesh_p)
      : lf::uscalfe::UniformScalarFESpace<double>(
            std::move(mesh_p), std::make_shared<CRReferenceFiniteElement>(),
            nullptr, nullptr) {}
  // You might have wondered what to do, but there is nothing to do!
  // The base class takes care of everything

  ~CRFeSpace() override = default;
};
```