Unique Pointers : Unique pointers are containers of raw pointers. It prevents copying of its contained pointers. That means it allows only one owner of the underlying pointer. If we try to make more than one copy of any resource then compile time error will occur.

Abstract Class : We use abstract class to provide an appropriate base class from which other classes inherit the behaviour of this abstract class. Abstract classes provide the structure of the class. It's method should be implemented by the class which inherits the abstract class. We cannot instantiate an abstract class. It helps in code reusability and abstraction.

In the Unique pointers example we created a Car class and assigned its object in an unique point. So, next time we can't reuse this pointer. Without using the unique pointer, we have to create a Car type pointer and assign the object in it. If we want to free this memory then we have to set this pointer as NULL value.

In abstract class example, We created a Vehicle abstract class which has a virtual method getSpeed. This abstract has a function which name is getMileage() that returns an integer value mileage and a protected variable mileage which value could be accessed by its base class.

Then we created a base class called Bus and inherited the Vehicle abstract class. Since we inherit an abstract class so we have to implement its virtual method.

Without using an abstract class later we could create a Bus class, In which we wrote all the implementation. So, if we have ten classes then we have to use the same code in all the base classes if we don't use abstract classes.