|Welcome back programmer. Are you ready to learn the prototype design pattern? First what is the prototype design pattern. Prototype is a creational design pattern that lets you copy existing objects without making your code dependent on their classes.|Say you have an object, and you want to create an exact copy of it. How would you do it? First, you have to create a new object of the same class. Then you have to go through all the fields of the original object and copy their values over to the new object. | But there’s a catch. Not all objects can be copied that way because some of the object’s fields may be private and not visible from outside of the object itself. There’s one more problem with the direct approach. Since you have to know the object’s class to create a duplicate, your code becomes dependent on that class. If the extra dependency doesn’t scare you, there’s another catch. Sometimes you only know the interface that the object follows, but not its concrete class, when, for example, a parameter in a method accepts any objects that follow some interface.|So what should you do? You should implement the cloneable interface in you class while overriding the clone method. And making a new constructer that takes the instance you want you clone and Carrie over all of the field values of the old object into the new one.|Good job keeping up till here now you have a brief knowledge about the prototype design pattern you can check the resources we provide for more explanation. Also don’t forget to take the quiz to receive your badge.