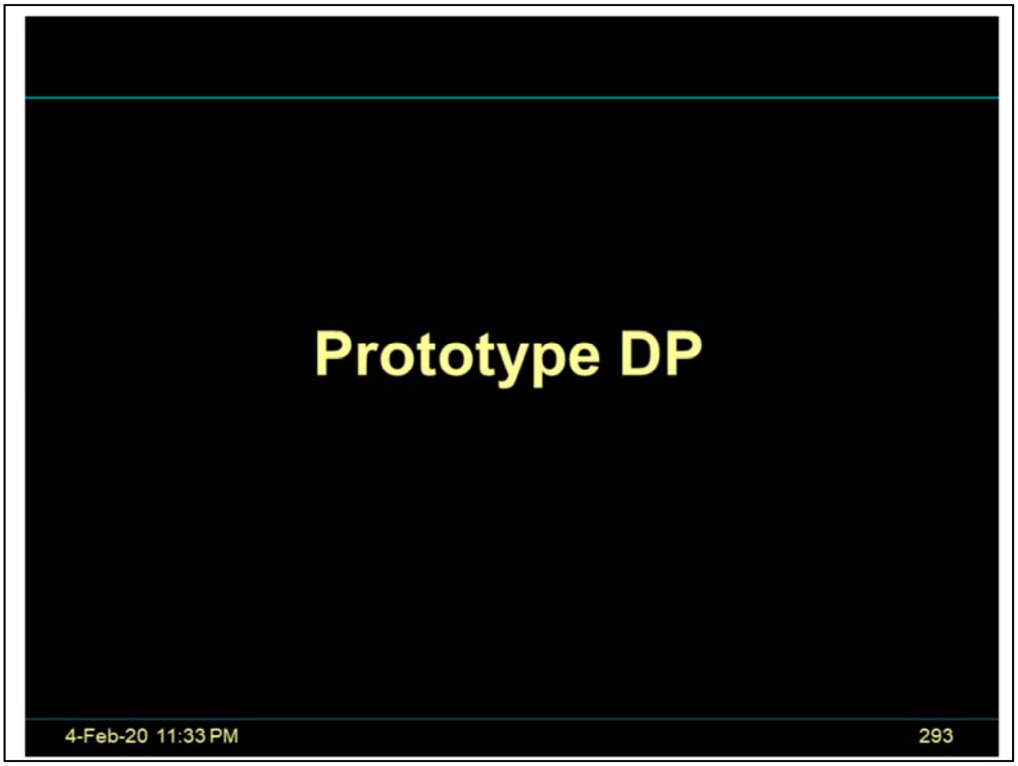
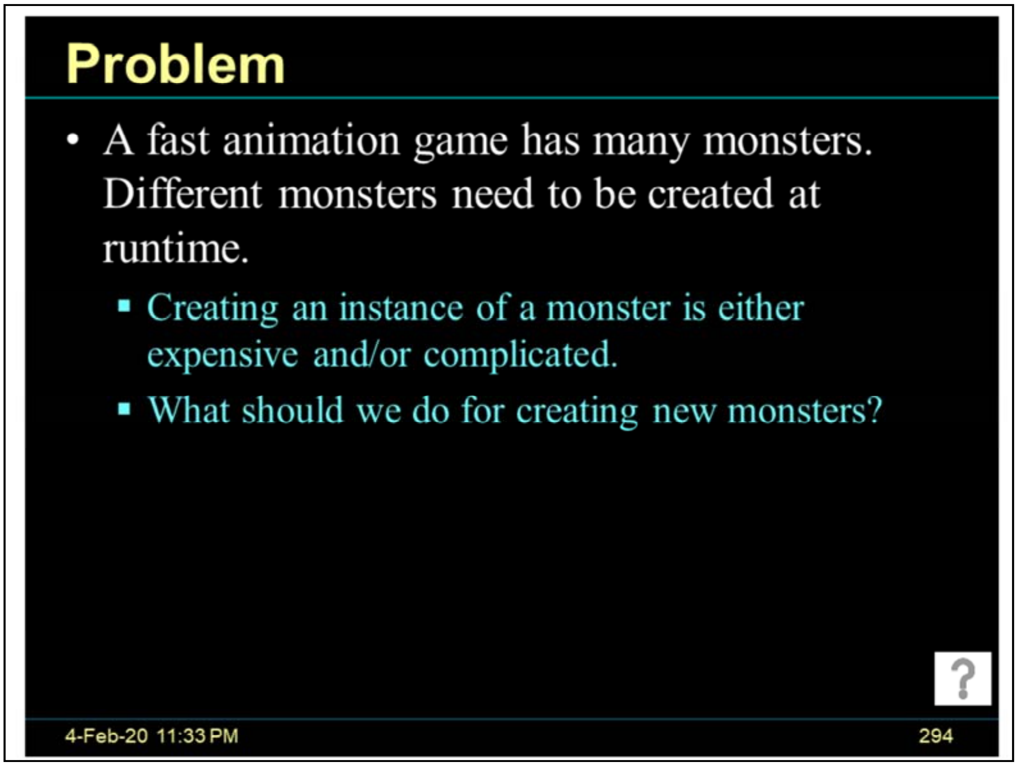
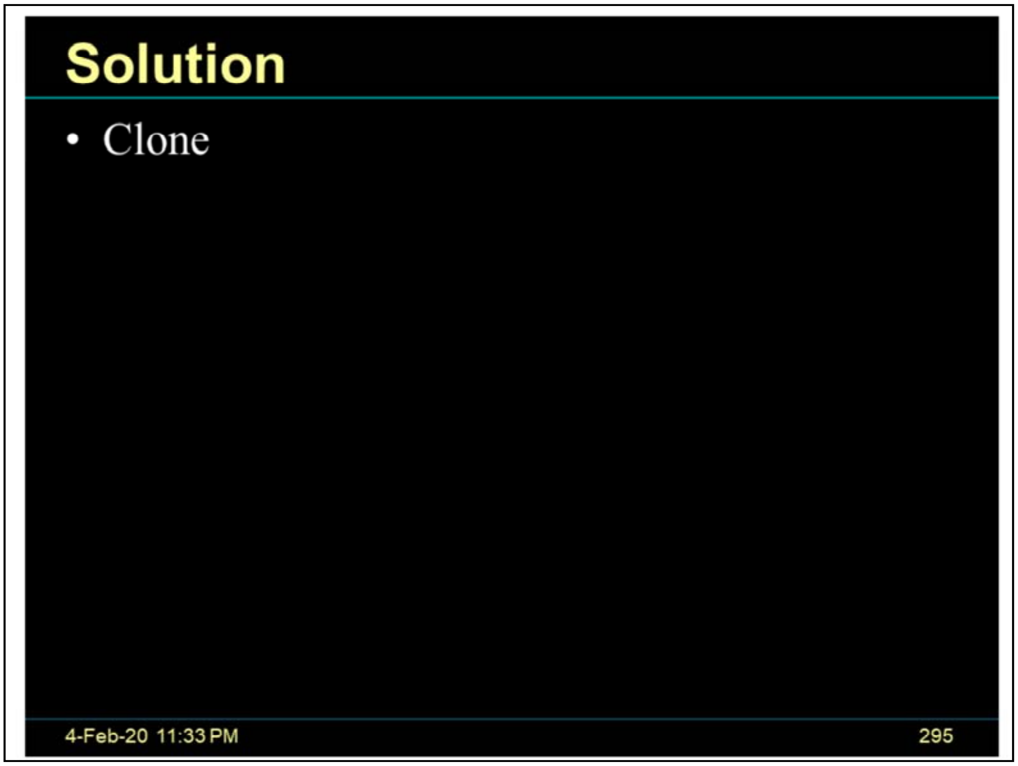


Prototype DP



Problem

- A fast animation game has many monsters. Different monsters need to be created at runtime.
  - Creating an instance of a monster is either expensive and/or complicated.
  - What should we do for creating new monsters?



Solution

- Clone

4-Feb-20 11:33 PM 295

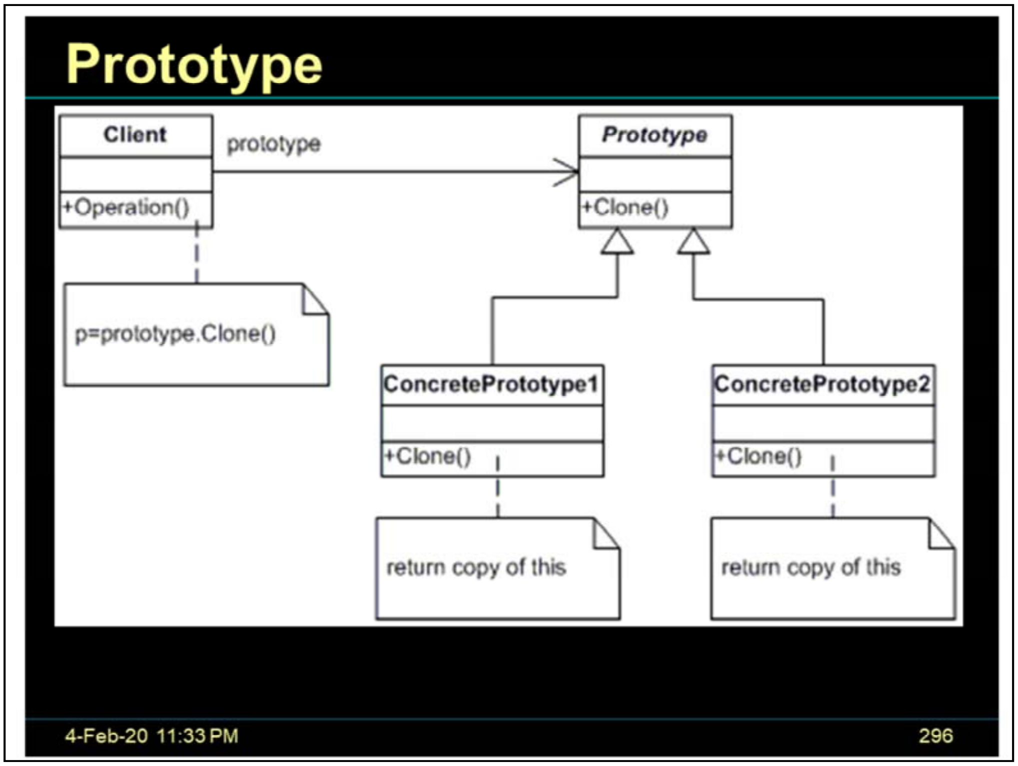
Solution: Make copies of existing objects and then modify as required.

Use clone method for shallow copies.

Use (de)serialization for deep copies.

The client instead of “new” operator calls a "clone" operation on the abstract base class.

The client can supply a enumerated data type that designates the particular concrete derived class desired.



Prototype

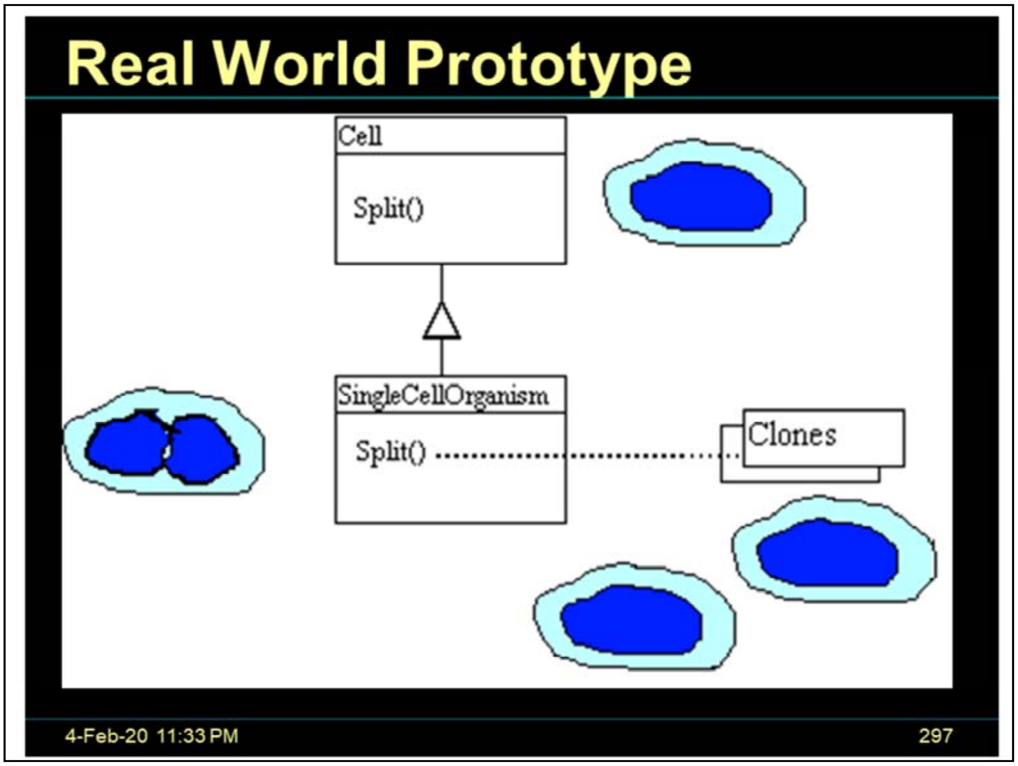
Specify the kinds of objects to create using prototypical instance and create new objects by copying this prototype.

================================

It hides the complexities of making new instances from the client.

Provides the option for the client to generate objects whose type is not known.

In some circumstances, copying an object can be more efficient than creating a new object.

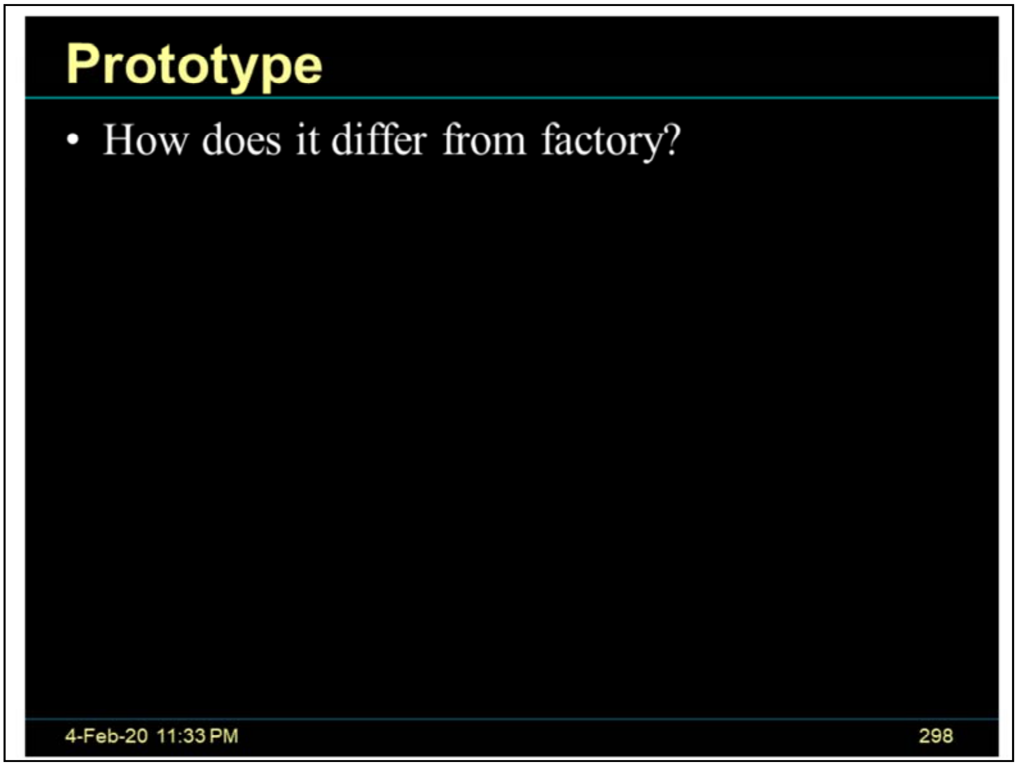


Real World Prototype

4-Feb-20 11:33 PM

297

The *Prototype* pattern specifies the kind of objects to create using a prototypical instance. Prototypes of new products are often built prior to full production, but in this example, the prototype is passive, and does not participate in copying itself. The mitotic division of a cell, resulting in two identical cells, is an example of a prototype that plays an active role in copying itself and thus, demonstrates the *Prototype* pattern. When a cell splits, two cells of identical genotype result. In other words, the cell clones itself.



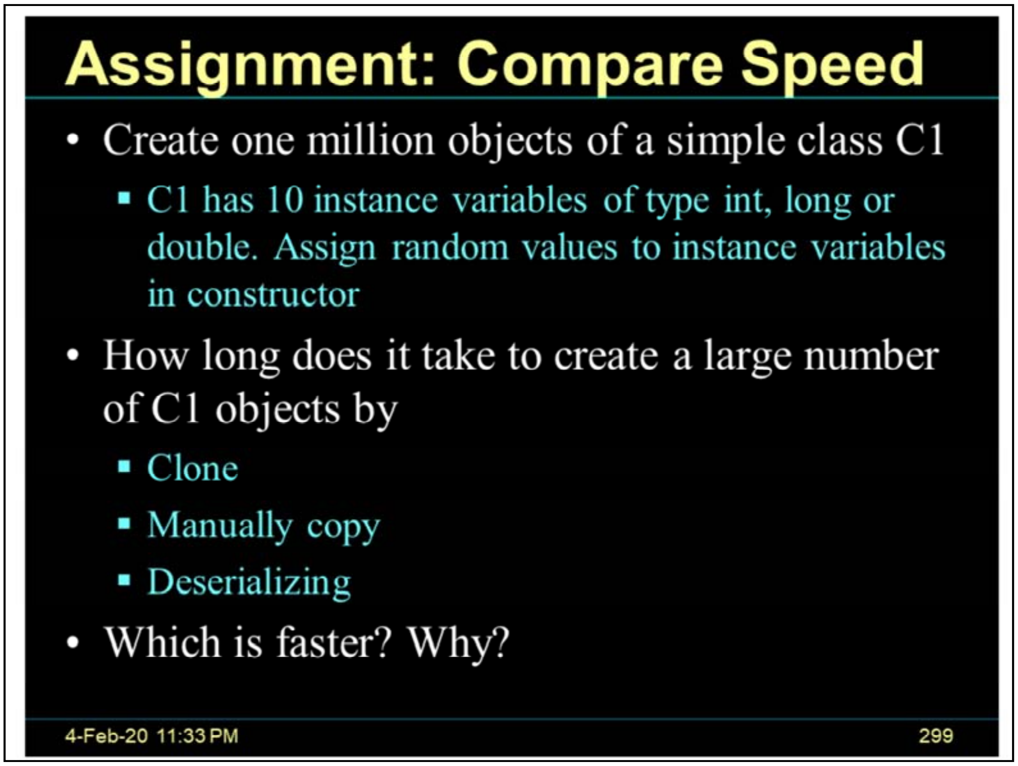
Prototype

- How does it differ from factory?

4-Feb-20 11:33 PM 298

Prototype vs. Factory

Prototype doesn't require sub classing, but it does require an "initialize" operation.
Factory Method requires sub classing, but doesn't require Initialize.



Assignment: Compare Speed

- Create one million objects of a simple class C1
  - C1 has 10 instance variables of type int, long or double. Assign random values to instance variables in constructor
- How long does it take to create a large number of C1 objects by
  - Clone
  - Manually copy
  - Deserializing
- Which is faster? Why?

4-Feb-20 11:33 PM 299

Java Solution is present in Benchmark project, under bnch.cloning

C# solution is under ooadAndpatterns solution.

Java - One result: Cloning takes 30ms, manual copy takes 29ms, Serialize/Deserialize takes 770ms

C# - One result: Cloning takes 230ms, manual copy takes 266ms, Serailize/Deserialize takes 40,000ms