# Singleton

Linda Marshall

Department of Computer Science
University of Pretoria

8 November 2021

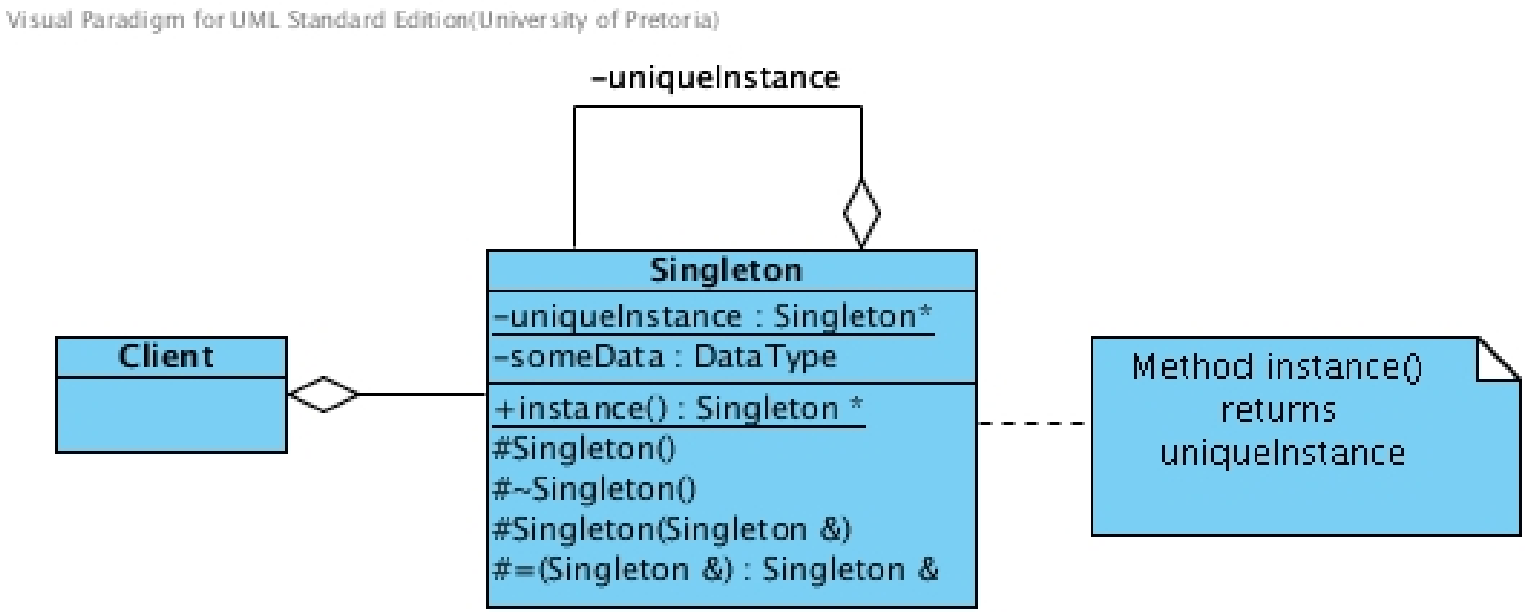
**Name and Classification:** Singleton (Object Creational)

**Intent:** "Ensure a class only has one instance, and provide a global point of access to it." GoF(127)

"Ensure a class only has one instance, and provide a global point of access to it."

GoF(127)

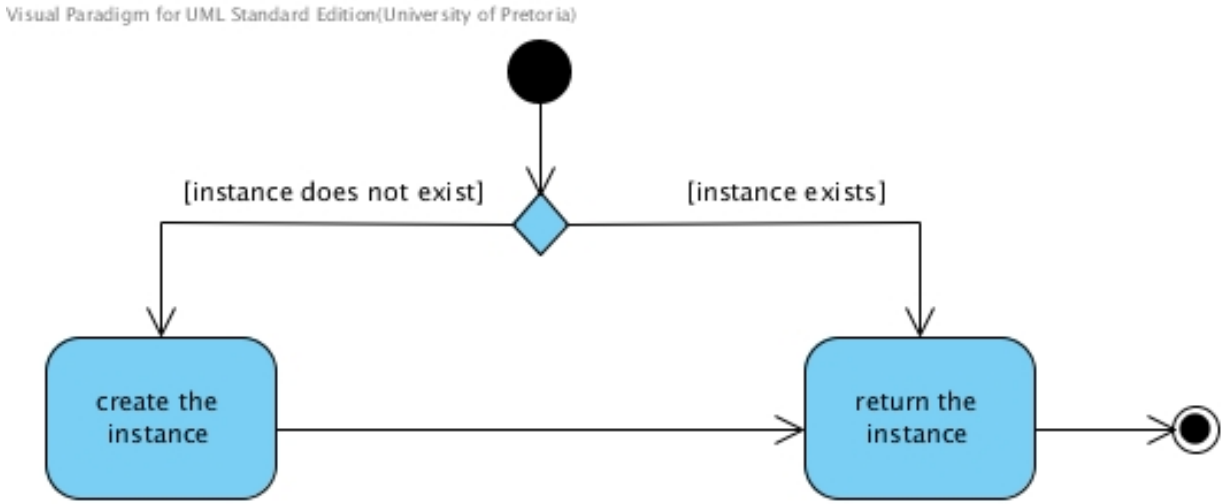
Take note of the following




T Muldner, C++ Programming with Design Patterns Revealed, Addison-Wesley,2002

## Take note of the following

- Functionality of the `instance` method.

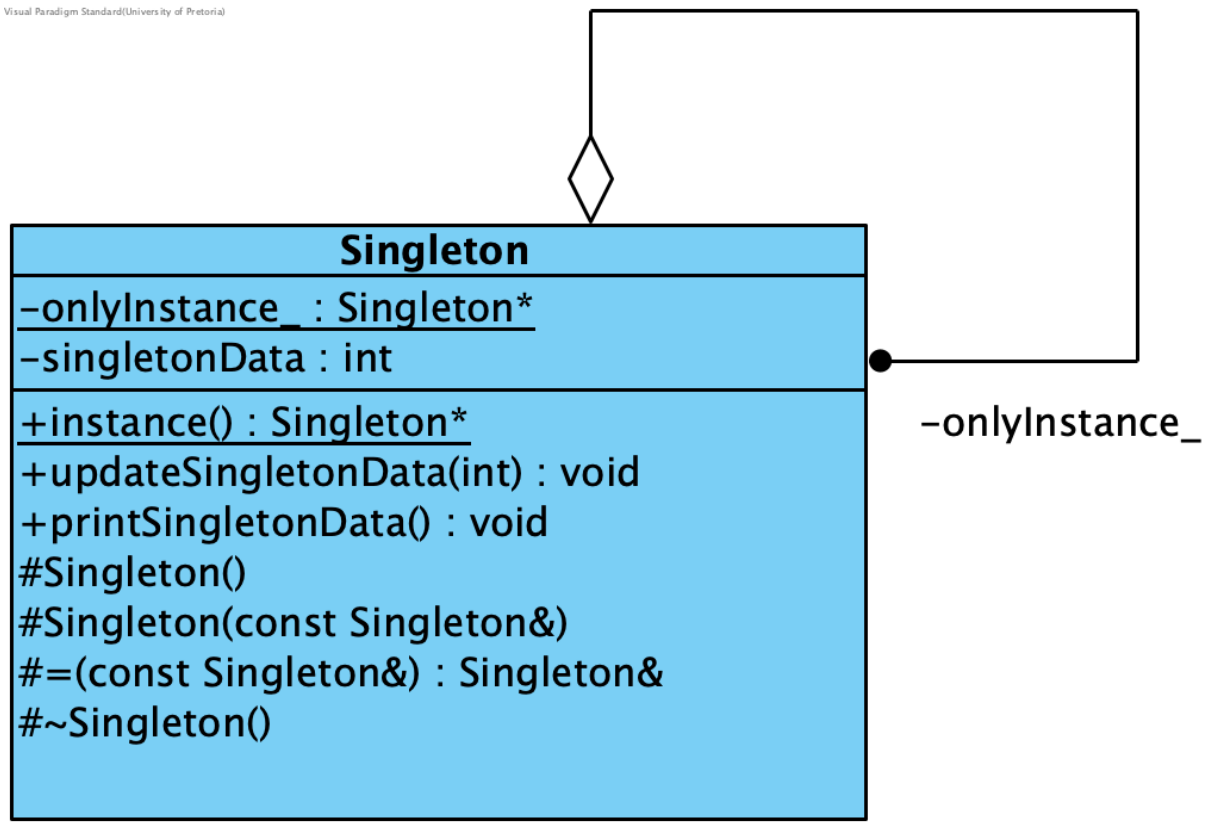



- Note the visibility of the constructor, Copy Constructor, Assignment operator and Destructor

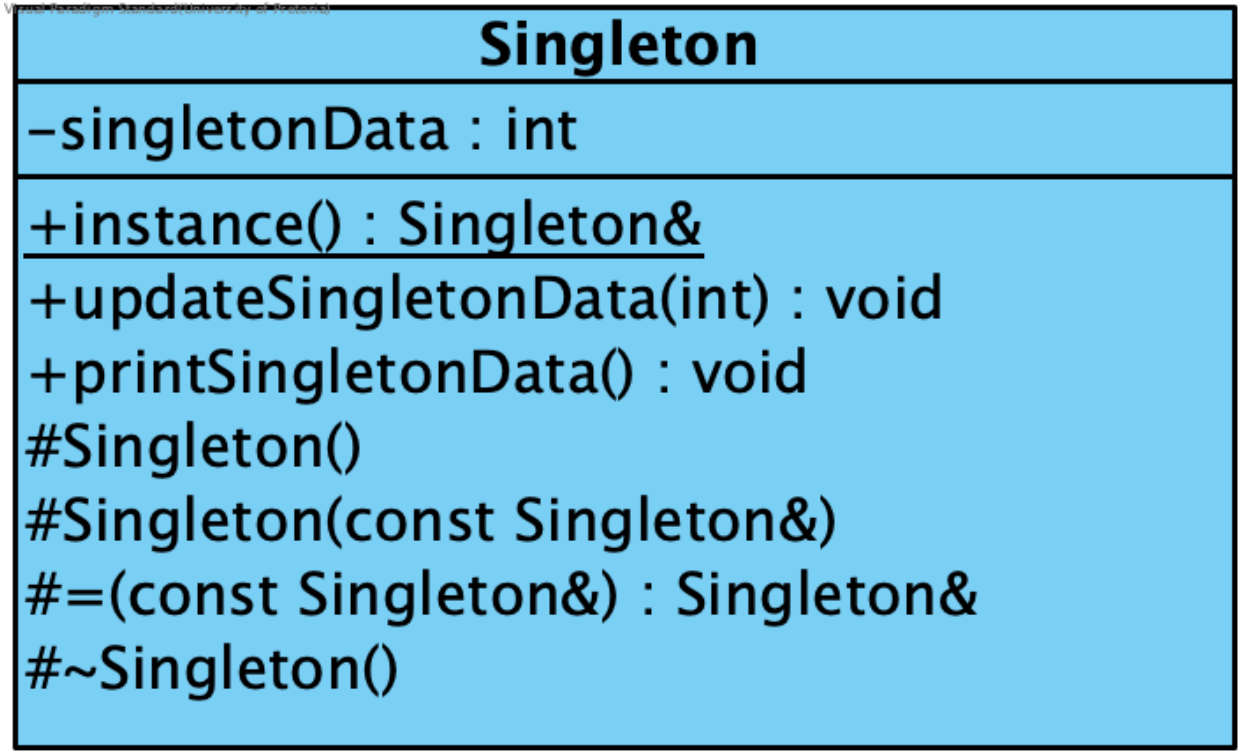
## Singleton

- defines an instance operation and ensures that the object is only constructable via this operation.

- Many patterns can be made a Singleton


Visual Paradigm Standard(University of Pretoria)
Singleton
-onlyInstance_ : Singleton*
-singletonData : int
+instance() : Singleton*
+updateSingletonData(int) : void
+printSingletonData() : void
#Singleton()
#Singleton(const Singleton&)
#=(const Singleton&) : Singleton&
#~Singleton()
-onlyInstance_

- Where is the instance variable `onlyInstance_` initialised?
- What about the memory leak?

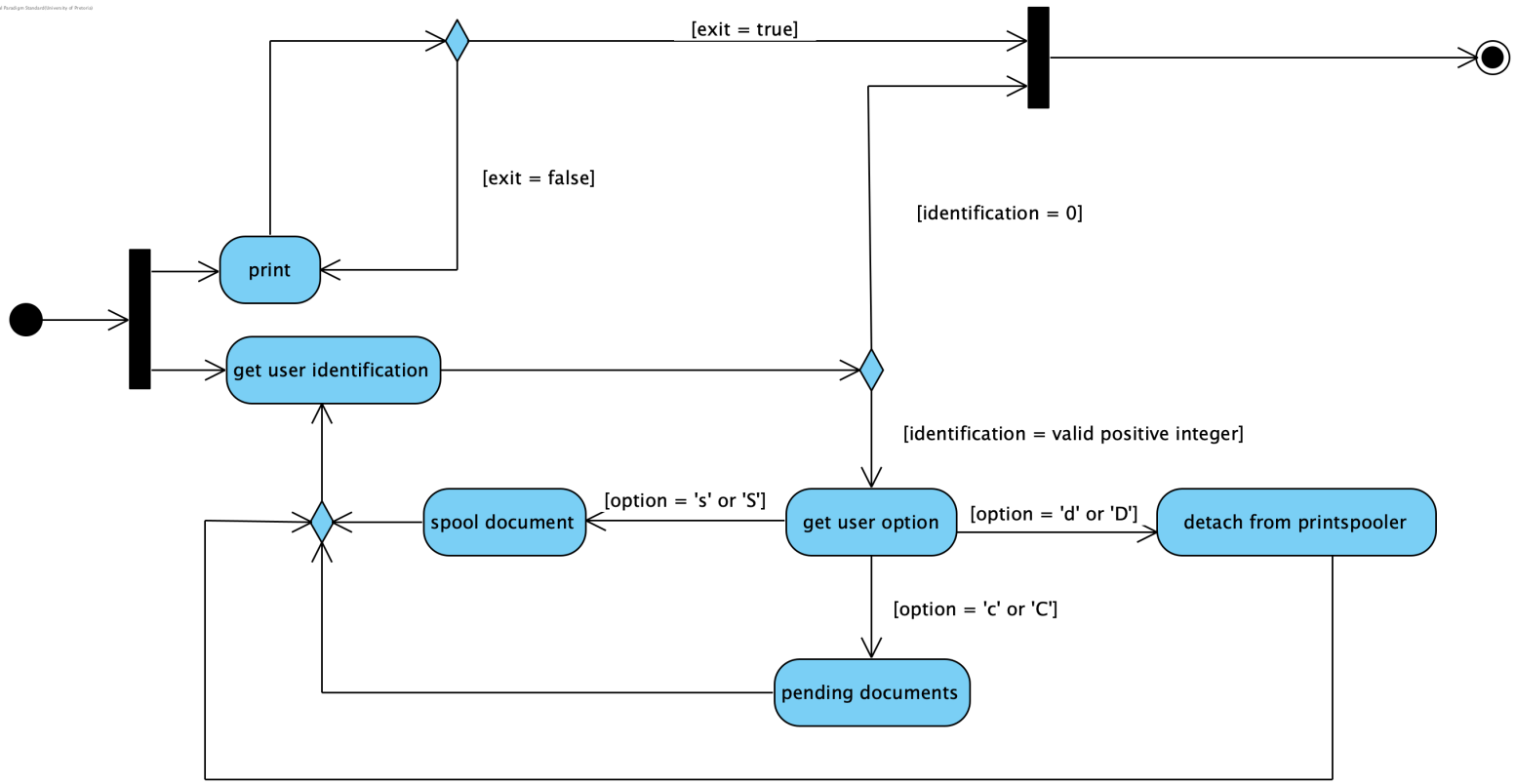

Singleton
-singletonData : int
+instance() : Singleton&
+updateSingletonData(int) : void
+printSingletonData() : void
#Singleton()
#Singleton(const Singleton&)
#=(const Singleton&) : Singleton&
#~Singleton()

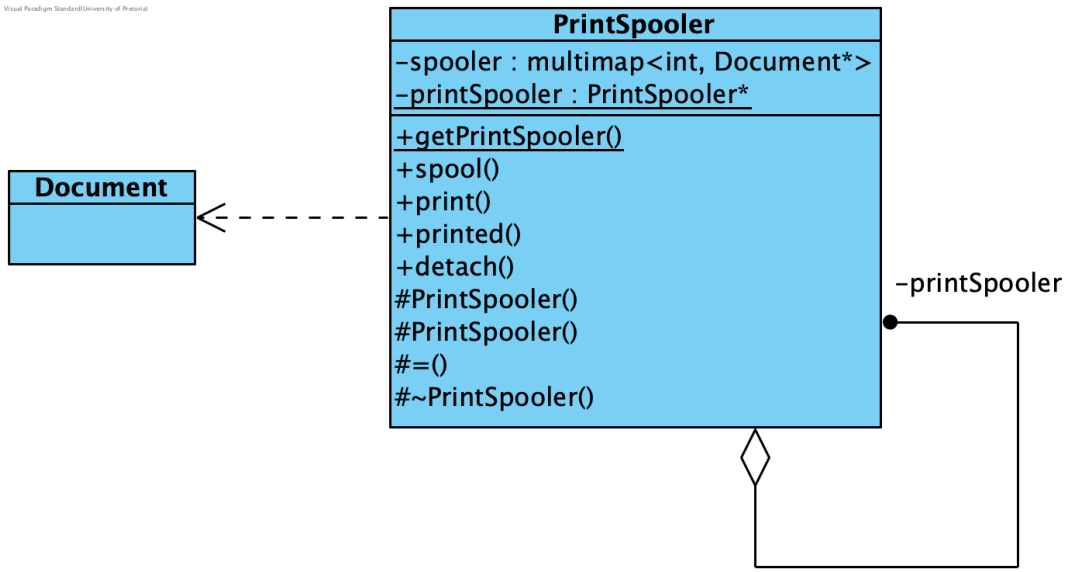
- creates a static Singleton object on the stack and returns a reference to the object
- the Singleton object is created when the `instance` method is called for the first time

- Why must the Constructor, Copy Constructor, Assignment operator and Destructor be protected or private?
- What are the implications of having the Constructor, Copy Constructor, Assignment operator and Destructor private vs protected?

- Would a virtual destructor in the Standard example solve the memory leak problem?
- The Meyers solution solves the memory leak problem!

- Do I know how many calls to Singleton `instance` have been made?
- If not, how do I make sure I know?


[exit = true]
[exit = false]
[identification = 0]
print
get user identification
[identification = valid positive integer]
[option = 's' or 'S']
spool document
get user option
[option = 'd' or 'D']
detach from printspooler
[option = 'c' or 'C']
pending documents


Visual Paradigm Standard(University of Pretoria)
PrintSpooler
-spooler : multimap<int, Document*>
-printSpooler : PrintSpooler*
+getPrintSpooler()
+spool()
+print()
+printed()
+detach()
#PrintSpooler()
#PrintSpooler()
#=()
#~PrintSpooler()
Document
-printSpooler