2007 Paper 6 Question 4

Concurrent Systems and Applications

(*a*) For *each* of the following tasks, give a code fragment that achieves as much of the task as is possible using the introspection API of the Java programming language and state which aspect(s) of the requirement is/are impossible in Java.

(*i*) Given a (non-null) object reference, determine whether or not the object has a *public*, *static* method named `myMethod` which takes no arguments. [4 marks]

(*ii*) Invoke the static method `public foo(java.lang.Integer x)` on a class definition named `MyClass` with argument `myInt` when the overloaded, static method `foo(java.lang.Number x)` is also defined on `MyClass`. [4 marks]

(*iii*) Given an object reference to an instance of a class named `Rocket`, set the value of its public field `numberOfEngines` to the (primitive) `int` value `5` and make the method `launch()` into a synchronized method. [4 marks]

(*b*) You are porting the JVM to a new processor that does not have a compare-and-swap (CAS) instruction but does offer test-and-clear (TAC): `tac(addr)` atomically reads the value stored at memory address `addr`, overwrites it with zero, and returns the value that was seen. Construct a Java-style *re-entrant* mutex using TAC. [4 marks]

(*c*) A server daemon has an object of type `Client` for each currently-active connection. Instances of `Client` each contain an object reference to a `java.lang.Socket` which must be closed (by calling `close()`) when the `Client` object is garbage collected. Show, by means of Java code fragments, how Phantom References and Reference Queues can be used to invoke the `close()` method in a timely fashion following an instance of `Client` becoming unreachable. [4 marks]