Contents

1. State chart components
2. Practice on state chart

State chart

A state diagram, also known as a state machine diagram, is a way of expressing dynamic information about a system. It is used to describe the behavior of a system or of an individual object. At any given point in time, the system or object is said to be in a certain state. It remains in this state until an event occurs that causes it to change state.

Several different types of event can cause the system to change from one state to another. In each state, the system behaves in a different way. A transition represents a change of state in response to an event.

**Basic components of a state chart diagram:**

1. **Start state:** a black filled circle represent the initial state of a system or a class.

UML-State-Diagram

1. **State:** a rounded rectangle represents a state. A state represents the conditions of an object at an instant of time.

UML-State-Diagram

1. **Transition:** asolid arrow represents the transition or change from one state to another. The arrow is labelled with the event which causes the change in state.

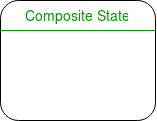
UML-State-Diagram

* Transitions from one state to the next are denoted by lines with arrowheads. A transition may have a trigger, a guard and an effect.

### Guard: a boolean condition that allows or stops a transition, written above the transition arrow.

### 

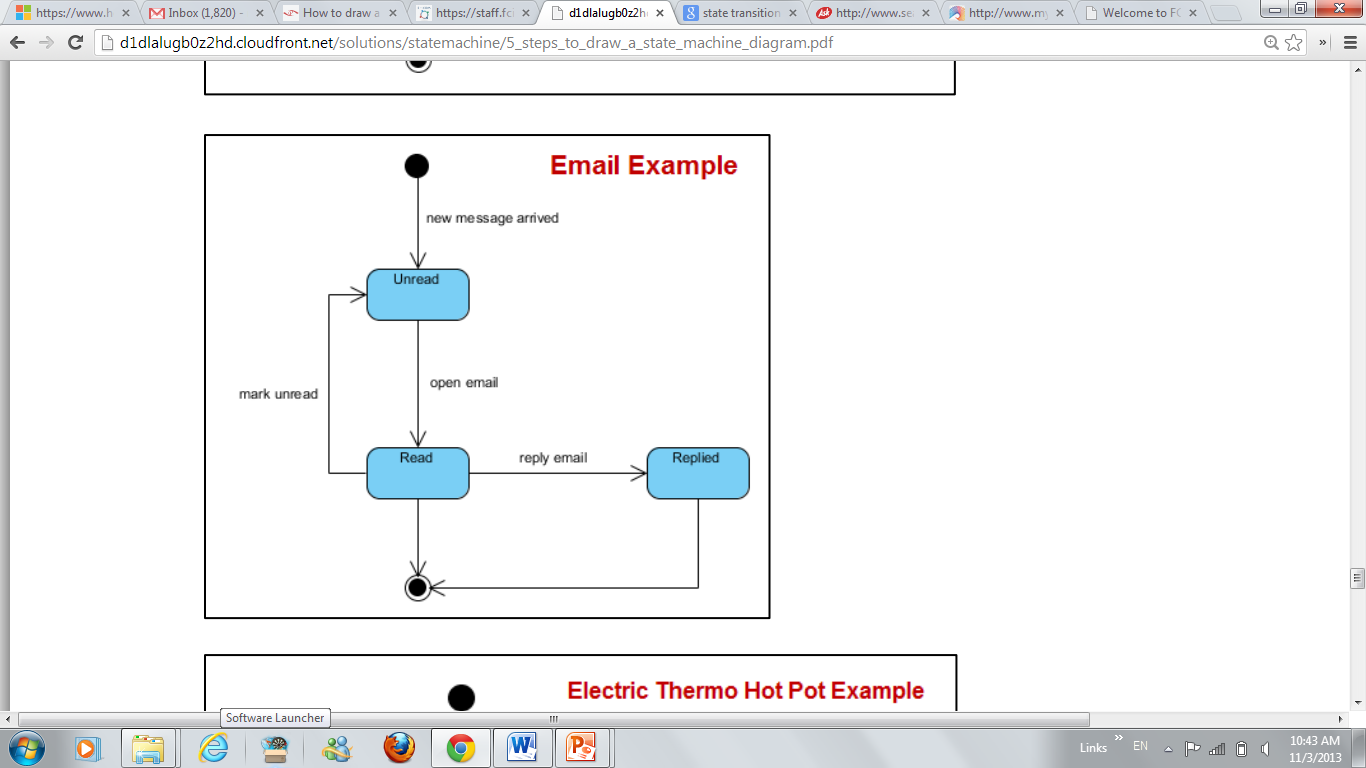
1. **Self-transition:** a solid arrow pointing back to the state itself represent a self-transition. There might be scenarios when the state of the object does not change.
2. **Composite state:** alsoa rounded rectangle represents a composite state. We represent a state with internal activities using a composite state.



1. **Final state:** a filled circle within a circle notation.

UML-State-Diagram

**Example:** state diagram for an email.

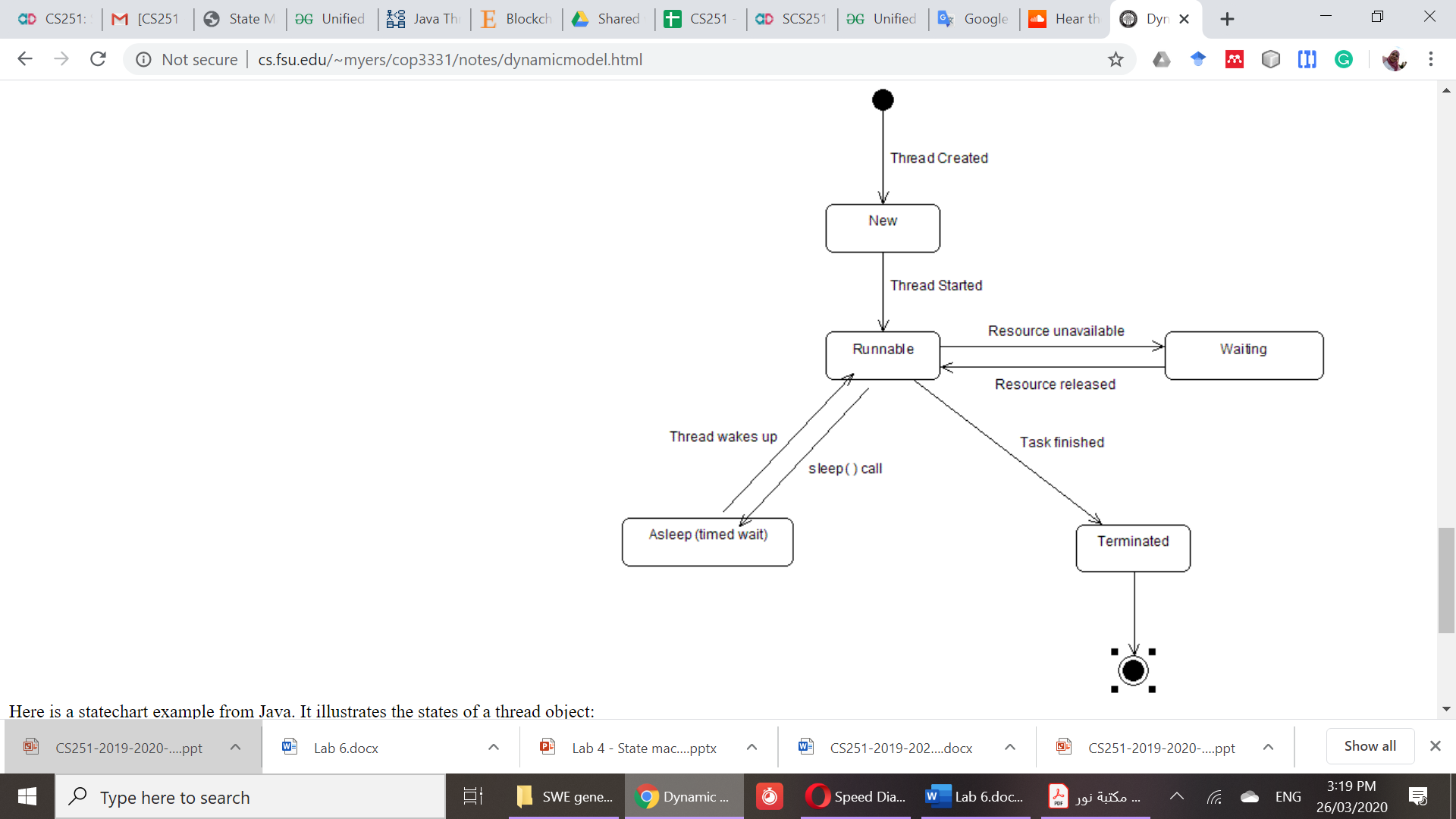
****

**Lab task**

**Draw the state chart diagram for a java thread object.**

**It has several states, these states are as follows:**

* new -- represents a newly created Thread object. Not yet running a task
* runnable -- a thread is now executing a task
  + Note: this is referred to as runnable instead of running because the processor still has to schedule threads and processes to run and take turns
* asleep (timed waiting) -- a thread has been put to sleep with a sleep() function call
* waiting -- a thread is blocked, waiting on a resource to become available
* terminated -- a thread has finished its task and has been shut down



References

1. Object-Oriented Software Engineering, Practical Software Development using UML and Java, Second edition
2. <https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/>
3. <https://sparxsystems.com/resources/tutorials/uml2/state-diagram.html>
4. <http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm>