CS F213 - Object Oriented Programming

J. Jennifer Ranjani

email: jennifer.ranjani@pilani.bits-pilani.ac.in

Chamber: 6121 P, NAB

Consultation: Fridays 4 – 5 p.m.

https://github.com/JenniferRanjani/Object-Oriented-Programming-with-Java

BITS Pilani

Pilani Campus

Context Switching w.r.t. Processes and Threads

- Threads
  - Values of registers, PC, stack pointer must be changed
  - Memory management information need not be changed because they share the same virtual address space
  - No system calls are required for cooperating among threads
- Processes
  - Requires all the above.

Thread class

Method	Meaning
getName	obtain a thread's name
getPriority	obtain a thread's priority
IsAlive	Determine whether the thread still running
join	Wait for the thread to terminate
run	Entry point for the thread
sleep	Suspend a thread for a period of time
start	Start a thread by calling its run method

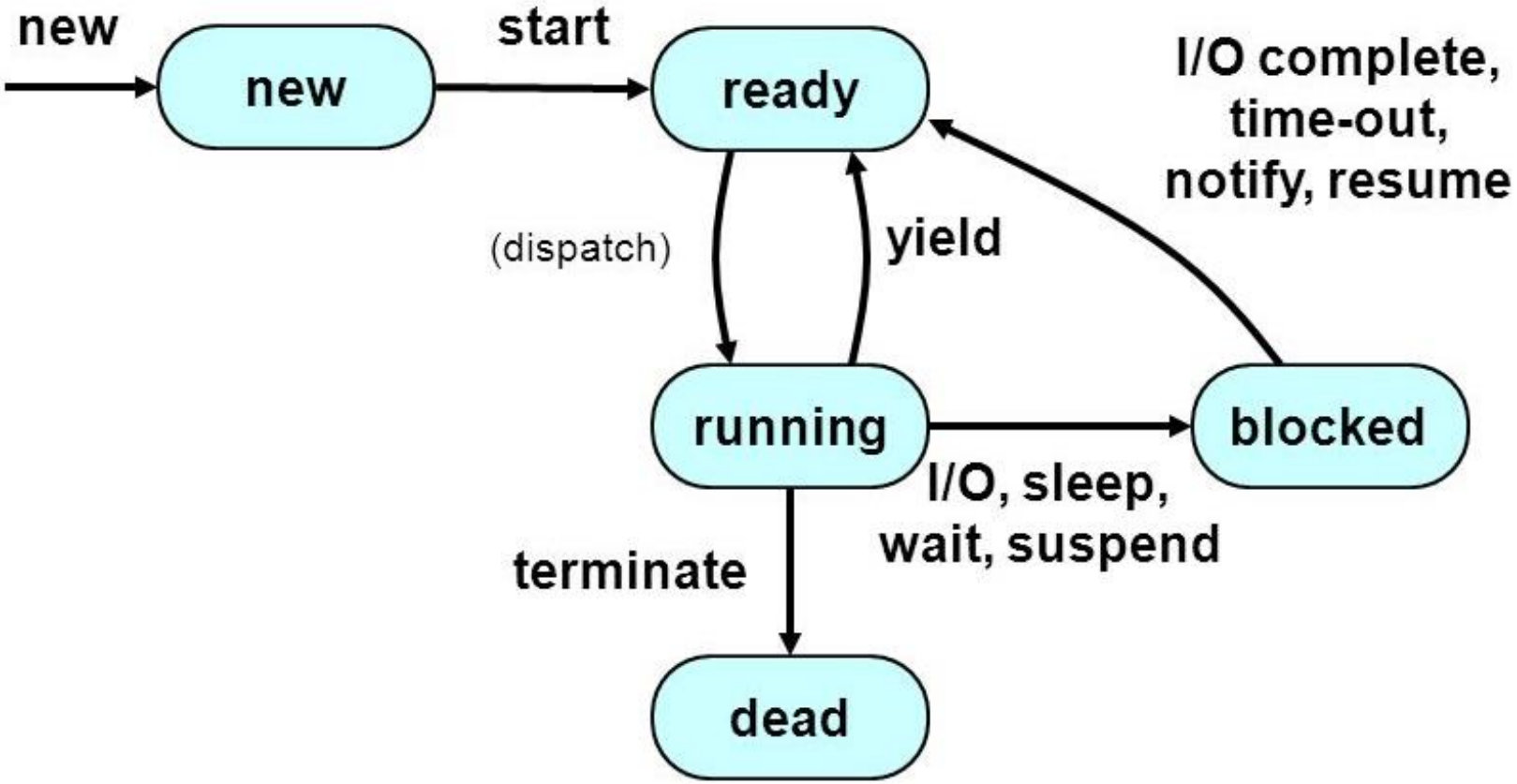
Main Thread

- When Java program starts, the main thread starts running immediately
  - It is the thread from which other threads are spawned
  - Often, it is the last thread to finish execution, because it performs various shutdown actions.
  - It is created automatically
    - but it can also be controlled through a Thread object
    - **currentThread()** can be used to obtain a reference to the main thread

Thread.sleep()

- Causes the current thread to suspend execution.
- This makes the processor time available for other threads waiting for the CPU.
- Two overloaded sleep methods
  1. sleep(long millis)
  2. sleep(long millis, int nanos)
- Sleep period can be terminated by interrupts.
  - you cannot assume that invoking sleep will suspend the thread for precisely the time period specified.

Thread States



Creating a Thread

It can be created in two ways

- Implementing the **Runnable** Interface
  - By creating a class that implements the Runnable interface
  - Only a single method **run()** needs to be implemented
  - Inside run(), define the code that the thread needs to perform
  - run() method can call other methods, use other classes and declare variables

- Thread(Runnable threadOb, String tName)
  - Defines where the execution of the thread will begin.

- New thread will begin execution only after we call the **start()**