**Chapter-30, Thread**

**1.** Which of the following class that you enable to create and control thread?

a) java.io.thread

b) **java.lang.thread**

c) java.util.\*

d) java.lang.system

**2.** How many main parts of thread or execution context?

a) 4 b)5

c)**3** d)2

**3.** Which of the following main parts of thread?

a)A virtual CPU

b) the data on which the code works

c)the code that the CPU execute

d) **above all**

**4.** Two thread shared the same data when they share access to a common\_\_\_\_\_\_\_.

a) class b) method

c) **object** d) interface

**5.** A thread constructor takes an argument that is an instance of\_\_\_\_\_.

**a)** Running b) New

**c)** Dead d) **Runnable**

**6.** To create a newly thread you must call which method.

a) close() b) **start ()**

c) sleep() d) wait ()

**7.** The model of preemptive scheduler is that many threads might be runnable but how many thread is running?

a) two b) three

c) **one** d) four

**8.** When a thread complete execution and terminates, it can’t run again?

a) **True**

b) False

**9.** Which method is to used to determine if a thread is still visible?

a)alive b) **isAlive**

c) runnable d) dead

10. The sleep method is one way to\_\_\_a thread for a period of time.

a) moving b)**halt**

c) running d) none

11. Join methods also depands on

a) operating system timers

b) schedulers

c) **a+b** d) none

**12.** Join also responds to Nan interrupt an exit with a/an

a)i/oException b)ArithematicException

c)NullPointerException

d)**InterruptedException**

**13.** Which method we use to give other runnable threads a chance to execute?

a) **Thread.yield()** b) Thread.wait()

c) Thread.sleep() d) none

**14.** A mechanism that enables a programmer to control thread that are sharing data is called

a) thread b) **synchronize**

c) wait d) deadlock

**15.** Which of the following serial of lifecycle method of a thread?

a) Runnable –New—Dead—Running--Nonrunnable

b) **New—Runnable—Running—Nonrunable—Dead**

c) Running—Dead—Nonrunnable—New--Runnable

d) New—Running—Runnable—Nonrunnable—Dead

**16.** If two Thread instance of same class the can share same code when they execute.

a. True

b. False

**17.** An instance of Runnable is made from a\_\_\_\_\_\_\_\_\_\_\_

a. Thread Object.

b. Thread mathod.

c. Object.

d. Class.

**18.** Multithreaded programming environment enables you to create multiple thread based on the\_\_\_\_\_\_\_\_\_\_

a. Different Runnable instance.

b. Same Runnable instance.

c. Two Runnable instance.

d. Three Runnable instance.

**19.** Which method run newly created Thread automatically?

a. begin();

b. stop();

c. trim();

d. start();

**20.** Preemptive and time-sliced are similer?

a. True

b. False

**21.** How many different states Thread object lifetime?

a. Two

b. Three

c. Four

d. Five(New,Runnable, Running, Blocked,Dead)

**22.** By which method can push Thread for preiod of time?

a. Thread.sleep();

b. Thread.start();

c. Thread.start-sleep();

d. Thread.sleepthread();

**23.** How many Thread Priority in java ?

a. One

b. Two

c. Three(Thread.MIN\_PRIORITY, Thread.NORM\_PRIORITY, Thread.MAX\_PRIORITY)

d. Four

**24.** What is the default priority in java Thread ?

a. Thread.MIN\_PRIORITY

b. Thread.NORM\_PRIORITY

c. Thread.MAX\_PRIORITY

**25.** What does Thread.yield() method do ?

a. stop Thread

b. start Thread

c. gives other runnable thread a chance to excute.

d. gives same runnable thread a chance to excute.

**26.** Which class is enables to create and control threads?

a. Java.swing.thread

b. Java.awt.thread

c. Java.lang.thread

d. Javax.swing.thread

**27.** Which one is true?

a. 2 threads can share the same data when they share access to a common object

b. 2 threads can share the same data when they share access to a different object

c. 2 threads can share the same data when they execute code from instance of the different class

**28.** Which one is true?

a. A newly created thread can be run automatically

b. A newly created thread cannot be run automatically

c. A newly created thread may be run automatically

**29.** Generally In java technology threads are \_\_\_\_\_\_\_?

a. Primitive

b. Boolean

c. Preemptive

d. Characteristics

30. The word preemptive means ---

a. Previously it was empty

b. Not primitive

c. Time-slicing

d. None of these

31. Which method is used to pausing a thread for some time?

a. Thread.pause ()

b. Thread.stop ()

c. Thread.sleep ()

**32.** Is it possible to make some actions at a time on a machine with one CPU by using thread?

a. Yes

b. No

**33.** The sleep is a \_\_\_\_ method in the thread class.

a. Dynamic

b. Static

c. Different

d. None of these

**34.** The word in thread “isAlive” means the thread is still \_\_\_\_\_\_\_?

a. Running

b. Alive

c. Not destroy

d. Viable

**35.** The term “isAlive” means is details \_\_\_\_\_\_\_?

a. The thread has been started and its task has been finished

b. The thread has been started but its task has not been completed

c. The thread has been started and already completed its job

d. The thread has been started and still it continues

**36.** In thread class “getPriority” method is a/an \_\_\_\_\_ type value.

a. Floating

b. Double

c. Int

d. Point

**37.** In thread Priority method default priority is -------

a. DEF\_PRIORITY

b. SET\_DEF\_PRIORITY

c. NORM\_PRIORITY

d. MIN\_PRIORITY

e. MAX\_PRIORITY

**38.** Which methods are responds to an interrupted method?

a. Sleep

b. InterruptedException

c. Join

d. None of the above

e. A & C

f. B & C

g. A & B

**39.** Why we use thread.yield () method---

a. To stop other runnable threads

b. To give other runnable threads a chance to execute

c. To pause other runnable threads and a chance to restart

d. All are false

**40.** Which keyword we used to stop corrupting data when more than single thread is running ---

a. Sleep

b. Break

c. Synchronized

d. Nothing of these

**41.** In java technology is there any “flag” option when creating object?

a. Yes

b. No

**42.** How many methods provide the “java.lang.Object” class?

a. 2

b. 3

c. 4

d. 1

**43.** Which are the methods of “java.lang.Object” class?

a. Wait

b. Notify

c. Break

a. A & C

b. A & B

c. B & C

**Chapter 30 Check Point Questions**

**1**. Why is multithreading needed?

A. Multithreading can make your program more responsive and interactive, and enhance the performance.

B. Multithreading is needed in many situations, such as animation and client/server computing.

C**. Both A and B**

D. None of the above.

2. What is a runnable object?

**A. An instance of Runnable is a runnable object.**

B. An instance of Running state is a runnable object.

C. An instance of sleep state is a runnable object.

**3. What is a thread?**

A. **A thread is wrapper object for a runnable object for executing a runnable task.**

B. A thread is a runnable task.

C. A thread is an Interface

D. None of the above.

4. **If a loop contains a method that throws an InterruptedException, why should the loop be placed inside a try-catch block?**

A. It will produce compile error

B. **If the loop is outside the try-catch block, the thread may continue to execute even though it is being interrupted.**

**C.** None of the above.

**5. How do you set a priority for a thread?**

A. You use the putPriority() method to set the priority for a thread

B. You use the addPriority() method to set the priority for a thread

**C. You use the setPriority() method to set the priority for a thread**

D. You use the getPriority() method to set the priority for a thread

6.What is the default priority?

A. The default priority of the thread is Thread.NORM\_PRIORITY (1).

**B. The default priority of the thread is Thread.NORM\_PRIORITY (5).**

C. The default priority of the thread is Thread.NORM\_PRIORITY (10).

D. The default priority of the thread is Thread.NORM\_PRIORITY (0).

7. Is an instance of FlashText a runnable object?

A. Yes. Because it implements the Runnable interface.

**B. No. Because it does not implement the Runnable interface.**

8. What is the purpose of using Platform.runLater?

A. **Invoking Platform.runLater(Runnable r) tells the system to run a task in the JavaFX application thread.**

B. Invoking Platform.runLater(Runnable r) does not tell the system to run a task in the JavaFX application thread.

9. Can the wait(), notify(), and notifyAll() be invoked from any object?

**A. Yes.**

B. No

**10. Can the read and write methods in the Buffer class be executed concurrently?**

A. Yes.

**B. No**

11. When invoking the read method, what happens if the queue is empty?

**A. It will wait for a signal for the queue to be not empty.**

B. It will sleep.

C. It will be dead.

**Written Question:**

2. How can multiple threads run simultaneously in a single-processor system?

Ans. Because most of time the CPU is idle--for example, the CPU is doing nothing while the user enters data--it is practical for multiple threads to share the CPU time in single-processor systems.

**3.** How do you define a task class? How do you create a thread for a task?

You can create a task classes by implementing the Runnable interface and create a thread for a task using the constructor new Thread(task).

9. What causes the text to flash?

You see the text flashing, because the program displays a text on the label and then display nothing on the label. This alternates to cause the flashing effect.

15. What are the benefits of using a thread pool?

You can create a thread for each task. This approach is convenient for a single task execution, but it is not efficient for a large number of tasks, because you have to create a thread for each task. Starting a new thread for each task could limit throughput and cause poor performance. A thread pool is ideal to manage the number of tasks executing concurrently.

16. How do you create a thread pool with three fixed threads? How do you submit a task to a thread pool? How do you know that all the tasks are finished?

To create a thread pool with three threads, use

ExecutorService executor = Executors.newFixedThreadPool(3);

To submit a task, use

executor.execute(task);

To check whether all tasks in a the pool are finished, invoke isTerminated() method.

20. How do you create a condition on a lock? What are the await(), signal(), and signalAll() methods for?

A condition on a lock can be created using lock.newCondition(). The await() method causes the current thread to wait until the condition is signaled. The signal() method wakes up one waiting thread, and the signalAll() method wakes up all waiting threads.

21. What would happen if the while loop in line 58 of Listing 32.6 was changed to an if statement?

**if** (balance < amount)

When a thread notify a waiting thread, you cannot assume the balance >= amount for the waiting thread. The condition (balance < amount) may be still true when the thread is awakened.

22. Why does the following class have a syntax error?

**public** **class** Test **implements** Runnable {

**public** **static** **void** main(String[] args) {

**new** Test();

}

**public** Test() **throws** InterruptedException {

Thread thread = **new** Thread(**this**);

thread.sleep(1000);

}

**public** **synchronized** **void** run() {

}

}

To override the init() method defined in the Applet class, you have to use the exact signature. The init() method does not claim throwing exceptions.

23. What is a possible cause for IllegalMonitorStateException?

If you invoke the methods on a condition without first acquiring a lock for the condition, an IllegalMonitorStateException would be thrown.

34. What is a deadlock? How can you avoid deadlock?

Deadlock occurs in the case that two or more threads acquire locks on multiple objects and each has the lock on one object and is waiting for the lock on the other object. The resource ordering technique can be used to avoid deadlock.

35. What is a thread state? Describe the states for a thread.

A thread state indicates the status of thread. Tasks are executed in threads. Threads can be in one of five states: New, Ready, Running, Blocked, or Finished.

When a thread is newly created, it enters the New state. After a thread is started by calling its start () method, it enters the Readystate. A ready thread is runnable but may not be running yet. The operating system has to allocate CPU time to it. When a ready thread begins executing, it enters the Running state. A running thread can enter the Ready state if it’s given CPU time expires or its yield () method is called.

36. What is a synchronized collection? Is ArrayList synchronized? How do you make it synchronized?

A synchronized collection is thread-safe, i.e., it can be accessed by multiple threads concurrently without being corrupted. ArrayList is not thread-safe. There are several ways to make it safe. You may obtain a lock before accessing it, or use the Collections.synchronizedList(list) to return a synchronized list.

37. Explain why an iterator is fail-fast.

An iterator is fail-fast. This means that if you are using an iterator to traverse a collection while the underlying collection is being modified by another thread, then the iterator will immediately fail by throwing java.util.ConcurrentModificationException.

38. How do you define a ForkJoinTask? What are the differences between RecursiveAction and RecursiveTask?

To define a ForkJoinTask, define a class that extends RecursiveAction or RecursiveTask. RecursiveAction is for a task that doesn't return a value and RecursiveTask is for a task that returns a value.

39. How do you tell the system to execute a task?

Invoke the invoke method on a ForkJoinTask object to execute a task.

40. What method can you use to test if a task has been completed?

Invoke the isDone method to test if the task is completed.

41. How do you create a ForkJoinPool? How do you place a task into a ForkJoinPool?

You can create a fork join pool using new ForkJoinPool(). To place a task to the pool, use the invoke method to add a task.