|  |
| --- |
|  |

Top of Form



|  |  |
| --- | --- |
| 1. | Thread Z holds the lock on object A. Thread X is blocked inside a wait call on ObjectA. What allows thread X to become runnable? |

ANS : 3

|  |  |  |
| --- | --- | --- |
|  | Thread Z calls Thread.sleep(100); | |
|  | Thread X's wait() times out. |

|  |  |  |
| --- | --- | --- |
|  | Thread Z releases the lock on A and calls the notifyAll() method on objectA. | |
|  | Thread X is interrupted. |

Bottom of Form

Top of Form

Top of Form



|  |  |
| --- | --- |
| 2. | Given the following, 1. class MyThread extends Thread { 2. 3. public static void main(String [] args) { 4. MyThread t = new MyThread(); 5. t.start(); 6. System.out.print("one. "); 7. t.start(); 8. System.out.print("two. "); 9. } 10. 11. public void run() { 12. System.out.print("Thread "); 13. } 14. } what is the result of this code?  ANS : 3 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | The output cannot be determined | | |
|  | Compilation fails | |
|  | | An exception occurs at runtime. | | | |
|  | | Thread one. Thread two. | | |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

|  |
| --- |
|  |

Top of Form



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3. | which class is the wait() method defined? Select the one      correct answer | | | | | | |
| ANS : | 1 | | | | | | |
|  | | Runnable | | |
|  | | | | Frame | |
|  | | | Object | | | |
|  | | | Thread | | | |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

|  |
| --- |
|  |

Top of Form



**.**

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | What is the name of the interface that can be used to define a class that can execute within its own thread?  ANS : 2 | | |
|  | | Threadable |

|  |  |  |
| --- | --- | --- |
|  | Runnable | |
|  | Thread |

|  |  |
| --- | --- |
|  | Run |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

|  |
| --- |
|  |

Top of Form



**.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5. | Which method must be defined by a class implementing the java.lang.Runnable  interface?  ANS : 4 | | | | | |
|  | | public void start() | | | |
|  | | | public void run(int priority) | |
|  | | | void run(int priority) |

|  |  |  |
| --- | --- | --- |
|  | public void run() | |
|  | void run() |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6. | You have an application that executes the following line: Thread myT = new Thread() ; Which statements are correct?  ANS : 2 | | | | |
|  | | | | If myT.stop() is called, the Thread can later be started with myT.start() and will execute the run method in the Thread class. | | |
|  | | | "The Thread myT has the priority of the Thread that executed the construction statement." | | | |
| "The Thread myT is now in a runnable state." | | | | |
|  | | If myT.start() is called, the run method in the class where the construction statement appears will be executed. | | | | |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7. | Given:  1. class MyThread extends Thread {  2. public void run() { System.out.println(“AAA”); }  3. public void run(Runnable r) { System.out.println(“BBB”); }  4.  5. public static void main(String[] args) {  6. new Thread(new MyThread()).start();  7. }  8. } What is the result?  ANS : 4 | | | |
|  | | The code runs with no output. | |
|  | | Compilation fails. |

|  |  |
| --- | --- |
|  | BBB |

|  |  |
| --- | --- |
|  | AAA |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form

|  |  |  |
| --- | --- | --- |
| 8. | What is the use of the synchronized keyword?  ANS : 1 | |
|  | | Ensures only one thread at a time may access a method or object | |
|  | | Ensures that two or more Threads will start and end at the same time | | |

|  |  |
| --- | --- |
|  | Ensures that two or more processes will start and end at the same time |
|  | Allows two process to run in parallel but to communicate with each other | |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9. | Which one CANNOT directly cause a thread to stop executing? | | | | | | | |
| ANS : | 4 | | | | | | | |
|  | | Calling the SetPriority method on a Thread object. | | | | | |
|  | | | Calling read method on an InputStream object. | | | |
|  | | | Calling the wait method on an object. | | |
|  | | | | Existing from a synchronized block. |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10. | What is the data type for the parameter of the sleep() method? | | | | | |
| ANS : | 1 | | | | | |
|  | | long | | | |
|  | | | int |
|  | | | short | |

|  |  |
| --- | --- |
|  | byte |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form

|  |  |
| --- | --- |
| 11. | What will happen when you attempt to compile and run the following code?             public class Bground extends Thread {                           public static void main(String argv[])      {                                  Bground b = new Bground(); b.run();                              }                           public void start()      {                                   for (int i = 0; i<10; i++)            {                                       System.out.println("Value of i = " + i);                                    }       } } |

ANS : 1

|  |  |
| --- | --- |
|  | Clean compile but no output at runtime |
|  | A compile time error indicating that no run method is defined for the Thread class | |

|  |  |  |
| --- | --- | --- |
|  | A run time error indicating that no run method is defined for the Thread class | |
|  | Clean compile and at run time the values 0 to 9 are printed out |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Top of Form



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12. | In order for there to be an efficient multi-threaded solution to a problem, what is a key characteristic that the problem must have?  ANS : 1 | | | | | | | | |
|  | | There are multiple tasks to be performed that involve very little cooperation | | | | | | |
|  | | | The program is to be run on a machine with multiple CPUs | | | | |
|  | | | It interacts with devices and networks | | | |
|  | | | | The program is to be used by multiple people | |
|  | | | | A problem takes a long time to solve |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Top of Form



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13. | What will be the outcome when the following application is executed? public class Test  {     public void newThread()         {               Thread t = new Thread()                  {                    public void run()                        {                          System.out.println("Going to sleep");                          try                            {                             sleep(5000);                           }                          catch (InterruptedException e)                             {                            }                         System.out.println("Waking up");                     }              };              t.start();             System.out.println("All done");  } public static void main(String[] args)     {       new Test().newThread();    } }  ANS : 3 | | | |
|  | | The code prints “Going to sleep” and then “Waking up.” | |
|  | | The code prints “All done” only. |
|  | | The code prints “All done,” then “Going to sleep,” and then “Waking up.” | | | |
|  | | The code prints “Going to sleep,” then “Waking up,” and then “All done.” | | | |

|  |  |
| --- | --- |
|  | The code does not compile. |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 14. | Which method actually schedules a thread for execution?  ANS : 1 | | | | | |
|  |  | | | | | |
|  |  | | | | | |
|  | | start( ); | | | |
|  | | | execute( ); | |
|  | | | init( ); |

|  |  |
| --- | --- |
|  | run( ); |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15. | Which method in the Thread class is used to create and launch a new thread of  execution?  ANS : 5 | | | |
|  | | run(Runnable r); | |
|  | | execute(); |

|  |  |  |
| --- | --- | --- |
|  | start( Runnable r); | |
|  | run(); |

|  |  |
| --- | --- |
|  | start( ); |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |  |  |  |
| --- | --- | --- | --- |
| 16. | Which of the following is true about the ‘not runnable’ state of a thread?  ANS : 4 | | |
|  | | A thread is ‘not runnable’ if it is dead | | |
|  | | | A sleeping thead enters the not runnable state when the specified time has elapsed | | | |
|  | | | When the start() method is invoked, the thread enters the ‘not runnable’ state | | |

|  |  |
| --- | --- |
|  | When a thread is blocked by another thread, is sleeping or is waiting, it enters the not runnable state |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17. | What is the default priority of a newly created thread?  ANS : 1 | | | |
|  | | A thread inherits the priority of its parent thread | |
|  | | | MAX\_PRIORITY (which is defined as 10 in the Thread class.) | | | |
|  | | | MIN\_PRIORITY (which is defined as 1 in the Thread class.) | | |

|  |  |
| --- | --- |
|  | NORM\_PRIORITY (which is defined as 5 in the Thread class.) |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Top of Form



|  |  |
| --- | --- |
| 18. | Which of the following is the only method provided by the Runnable interface? |

ANS : 3

|  |  |  |
| --- | --- | --- |
|  | init( ); | |
|  | | sleep( ); | |

|  |  |
| --- | --- |
|  | run( ); |
|  | start( ); | |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



|  |
| --- |
|  |

Top of Form

|  |  |  |  |
| --- | --- | --- | --- |
| 19. | Given: 1.   public class Test extends Thread  2.        { 3.            public static void main(String[] args)  4.                 { 5.                     Test t = new Test(); 6.                     Thread tt = new Thread(t); 7.                     tt.start();   8.                  } 9.             public void run()  10.                  { 11.                    System.out.println("wow! what an output"); 12.                 } 13.     } What will be the result of compiling and running above code?  ANS : 2 | | |
|  | | Exception is thrown at runtime |

|  |  |  |
| --- | --- | --- |
|  | "Program prints:wow! what an output" | |
|  | Compilation suceeds, but No Output |

|  |  |
| --- | --- |
|  | Compilation Fails |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

|  |  |
| --- | --- |
| 20. | What is the o/p of the following program ?  1. class MyThread extends Thread { 2. 3. public static void main(String [] args) { 4. MyThread t = new MyThread(); 5. Thread x = new Thread(t); 6. x.start(); 7. } 8. 9. public void run() { 10. for(int i=0;i<3;++i) { 11. System.out.print(i + ".."); 12. } 13. } 14. }  ANS : 2 |

|  |  |
| --- | --- |
|  | 0..1..2..3.. |

|  |  |
| --- | --- |
|  | 0..1..2.. |
|  | Compilation fails. | |

|  |  |  |
| --- | --- | --- |
|  | An exception occurs at runtime. | |
|  | 1..2..3.. |

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form



Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form