نظام تشغيل CS222

كويز 4 ترم 443

السؤال 1

0.5 درجات — تم الحفظ

:The point of using **wait()** method instead of **sleep()** method because

- The wait() method add functionalities to control the thread more than the sleep() method
- The wait() method releases the lock prior to waiting, and reacquires the lock prior to returning from the wait() method
- The wait() method force the thread to sleep and give control to other threads
- The wait() method does not release the lock and keep waiting until the specify condition is met

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 1 من 6

السؤال 2

0.5 درجات — تم الحفظ

.The **notifyAll()** method unblocks/wakes up all threads waiting for the lock object

- صواب
- خطأ

- `notify()` توقظ خيطًا واحدًا فقط، بينما `notifyAll()` توقظ جميع الخيوط.

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 2 من 6

السؤال 3

0.5 درجات — تم الحفظ

.Race condition occurs if the effect of multiple threads on shared data depends on the order in which the threads are scheduled

يكون فيه سباق اذا كان فيه ترتيب

- صواب
- خطأ

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 3 من 6

السؤال 4

0.5 درجات — تم الحفظ

.Deadlock occurs if no thread can proceed because each thread is waiting for another to do some work first

جمود

يكون فيه جمود اذا ما ارتبط بشيء

- صواب
- خطأ

يمنع الانتقال إلى السؤال التالي إجراء تغييرات على هذه الإجابة.

السؤال 4 من 6

السؤال 6

0.5 درجات — تم الحفظ

.The lock object is used to terminate a thread before accessing a shared resource

لحل مشاكل التحكم في الموارد المشتركة بين الخيوط (threads)، تستخدم كائن القفل (Lock). هذا القفل يضمن أن خيطًا واحدًا فقط يمكنه الوصول إلى المورد في وقت واحد، مما يمنع التداخل والمشاكل.

لانهاء

★ يستخدمون القفل عشان انهاء

- صواب
- خطأ

انقر فوق إرسال لإكمال هذا التقييم.

السؤال 6 من 6

السؤال 5

0.5 درجات تم الحفظ

:Suppose that the following function manipulate a shared resource

```
}public void func(double value)
...
{
```

:and you want to use the Java build-in lock, which of the following is correct

- }public void func(double value)
  }lock
  ...
  {
  {
- }public lock void func(double value)
  ...
  {
- }synchronized public void func(double value)
  ...
  {
- }public synchronized void func(double value)
  ...
  {

