

Course No : SWE 4412 **College of Engineering** **Student No:** _____

Course Title: Concurrent and real-time programming **Final Exam First Term 2015** **Student Name:** _____

Exam Time: 2 Hours **Eng. Ibrahim Al-Sharif** **Date: 21-01-2015**

No of Questions: 6 **Total Grade: 50** **Try to Answer: All**

1- State if the following statements True Or False: (20 marks)

1. The instance method of the immortal memory class creates a new immortal memory area when called ().
2. When there are schedulable objects active inside scoped memory area, the allocated memory can be reclaimed ().
3. JVM responsible for managing the heap memory ()
4. The RationalTime class is the parent class of the RelativeTime class().
5. Absolute time is a duration measured by a particular clock. ().
6. The real time clock is unique. ().
7. Timers are RTSJ clocks.().
8. Scheduling parameters are used by a scheduler to determine which object to execute()
9. The heap is typically used for storing variables of basic data types that are local to a method ().
10. Immortal and scoped memory are physically outside the heap().
11. A reference count indicates the number of times the scoped memory has entered and existed.()
12. A dangling reference is a reference to an object that has been collected when scope memory is reclaimed.()
13. A reference from the Immortal to the scoped memory is allowed.()
14. A quantity semaphore can be used for mutual exclusion.()
15. It two schedulable object are using a scoped memory competitively, they will be active in the scoped memory area simultaneously. ().
16. A persistence signal remains set until all waiting threads received it.()
17. Garbage collection performed only when the heap is full.().
18. Running garbage collector may have a significant impact on the response time of time-critical thread.().
19. Java wall clock accuracy depends on the host system.().
20. If Java thread called one of the enter methods for memory area an exception is thrown().
21. Size estimator returns only the size of the created object.()

Course No : SWE 4412 College of Engineering Student No: _____

Course Title: Concurrent and real-time programming Final Exam First Term 2015 Student Name: _____

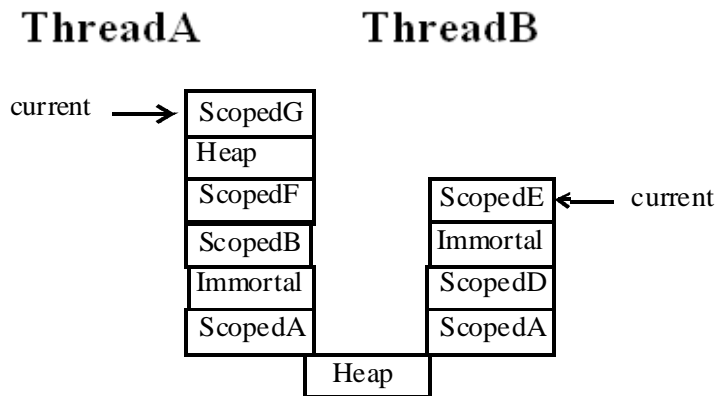
Exam Time: 2 Hours Eng. Ibrahim Al-Sharif Date: 21-01-2015

No of Questions: 6 Total Grade: 50 Try to Answer: All

2- Answer the following questions. (15 marks)

- 1- List and explain the memory areas types in RTSJ.
- 2- Why do we need memory management in real-time systems?
- 3- When you advise to use semaphore and barriers communication paradigm?
- 4- Release parameters affects the scheduling of real-time objects, Identify and explain the three types of release parameter.
- 5- What are the two types of schedulable objects in RTSJ?

3- Draw the new stack.(5 marks)



Draw the cactus stack when ThreadB wishes to enter ScopedF.

Course No : SWE 4412 College of Engineering Student No: _____

Course Title: Concurrent and Final Exam First Term Student Name: _____
real-time programming 2015

Exam Time: 2 Hours Eng. Ibrahim Al-Sharif Date: 21-01-2015

No of Questions: 6 Total Grade: 50 Try to Answer: All

6- Consider the following class: (Bonus 10 marks)

```
import javax.realtime.*;
public class RtThread extends RealtimeThread {
    ScopedMemory A;
    ScopedMemory B;
    ScopedMemory C;
    ScopedMemory D;

    Runnable r1 = new Runnable() {
        public void run() {
            HeapMemory.instance().enter(r5); B.enter(r2); } };

    Runnable r2 = new Runnable() { public void run() {
    C.enter(r3);
    ImmortalMemory.instance().enter(r6); } };

    Runnable r3 = new Runnable() { public void run() {
    D.enter(r4);
    try {
    A.executeInArea(r7); } catch (Exception e) { };
    } };

    Runnable r4 = new Runnable() { public void run() {
    // Create a new Schedulable object W here with immortal memory as the initial memory area.
    RealtimeThread W = new RealtimeThread(null, null, null, ImmortalMemory.instance(), null,
    null); } };
    Runnable r5 = new Runnable() {
    public void run() {
    // Create a new Schedulable object X here with A as the initial memory area.
    RealtimeThread X = new RealtimeThread(null, null, null, A, null, null); } };

    Runnable r6 = new Runnable() {
    public void run() {
    // Create a new Schedulable object Y here with heap as the initial memory area.
    RealtimeThread Y = new RealtimeThread(null, null, null, HeapMemory.instance(), null, null);
    } };

    Runnable r7 = new Runnable() {
    public void run() {
    // Create a new Schedulable object Z here with B/ as the initial memory area.
    RealtimeThread Z = new RealtimeThread(null, null, null, B, null, null); } };

    // Constructor
    public RtThread(SchedulingParameters sp, ReleaseParameters rp, MemoryParameters mp,
    MemoryArea ma, ProcessingGroupParameters pgg, ScopedMemory one, ScopedMemory two,
    ScopedMemory three, ScopedMemory four) {
```

Course No : SWE 4412 **College of Engineering** **Student No:** _____

Course Title: Concurrent and real-time programming **Final Exam First Term 2015** **Student Name:** _____

Exam Time: 2 Hours **Eng. Ibrahim Al-Sharif** **Date: 21-01-2015**

No of Questions: 6 **Total Grade: 50** **Try to Answer: All**

```
super(sp, rp, mp, ma, pggp, null);
A = one; B = two; C = three; D = four;}

// run method of the RtThread thread class
public void run() {
    // Create a new schedulable object V here with the heap as the initial memory area.
    RealtimeThread V = new RealtimeThread(null, null, null, HeapMemory.instance(), null, null);
    A.enter(r1); }} // End of class RtThread
```

Assume that an instance of RtThread is created with the heap as the current memory area and with the heap as the memory parameter (that is, the initial memory area). Illustrate the memory stacks for the created real-time threads V, W, X, Y and Z.