There are 100 possible points. Write all answers on the answer sheets provided.

1. (4 pts) What is an operating system?

2. (4 pts) Describe the fetch-execute cycle.

3. (3 pts each) Briefly define/explain these terms:
   (a) kernel
   (b) process
   (c) quantum
   (d) context switching

4. (4 pts) Compare and contrast threads and processes.

5. (4 pts) Name the primary reason that cache’s are used. Name at least one problem that arises from the use of a cache.

6. (6 pts) Draw a diagram showing the (three) possible states of a process in a multiprogramming system.
   (a) (3 pts) Give a phrase or sentence describing each state in your diagram.
   (b) (4 pts) For each transition (i.e., arrow) in your diagram, give one example of an event that could cause such a transition.

7. (4 pts) On the axis provided show the approximate relationship between selection of a quantum size and response time.

8. (4 pts) Name two reasons that threads are used.

9. (4 pts) To what does the term “kernel mode” refer?
10. (6 pts) Give a detailed account of what happens when an interrupt occurs.

11. (4 pts) List at least four data you would expect to find in a process control block.

12. (3 pts) Describe a typical scheduling strategy used by a non-preemptive scheduling scheme.

13. (4 pts) Explain why a plain round-robin scheduling algorithm might not be the best approach in a modern, multi-user, interactive, networked operating system.

14. (16 pts) Write a complete Java program in which a parent process reads a list of names from a user (they enter a blank name to quit). After all the names have been entered, for each name, the parent process should create a child thread who will display its thread id and then capitalize the string (using the toUpperCase() method). The modified string should be returned to the parent process which will in turn display the received value. The parent should not wait on any child until all the children have been created.

You may assume there will not be more than 100 names entered.