Computer Engineering Department
PhD Qualifier Exam

13 September, 2013

- Time: 09:30 - 12:30
- The exam contains 8 questions in total. You are to choose 6 of these questions and answer only to them.
- By submitting this exam, you agree to fully comply with Koç University Student Code of Conduct, and accept any punishment in case of failure to comply. If you do not submit this exam on time, you will directly fail the qualifier.
- The exam is open-book and open-notes.
- You are not allowed to use any electronic equipment such as computers and mobile phones.
- In general, you are not allowed to ask questions during the exam.
- Good luck!

1. **a)** Draw the result of inserting the keys (10, 26, 32, 55, 12, 20, 28, 40, 60, 22, 11) into an initially empty red-black tree using top-down insertion (that is, the insertion algorithm that performs only one pass from root to the leaf). Show your work step by step (that is, every insertion).

b) Give an **non-recursive** algorithm to search for a certain element x in a stack S, using only a queue Q and a constant number of **reference** variables. Your method should return true if x is in the stack. Complete the following Java-like method in pseudo-code.

```
Input:  Stack S, Item x
Output: returns true if x is in S,   S is unchanged

booleean searchStack(Stack S, Item x) {
    Queue Q;









}
```

2. Let X and Y be independent continuous random variables with PDFs f_X and f_Y, and CDFs F_X and F_Y, respectively. Remember the definition of CDF:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(t)dt$$

Let $S = \min(X, Y), L = \max(X, Y)$.

(a) Express the following probability in terms of the CDFs F_X and F_Y.

$$P(S \geq 1)$$

(b) Let a and d be two numbers with $a < d$. Express the following probability in terms of the CDFs F_X and F_Y.

$$P(a \leq S \text{ and } L \leq d)$$

(c) Consider four numbers $a < b < c < d$. Express the following probability in terms of the CDFs F_X and F_Y.

$$P(a \leq S \leq b \text{ and } c \leq L \leq d)$$

3. Consider the following pseudocode:

function Compute (a, n: integers)
if $n = 0$ **then**
    $y = 1$
else if n mod $2 = 0$ **then**
    $y = (\text{Compute}(a, n/2))^2$
else
    $y = a{*}(\text{Compute}(a, \lfloor n/2 \rfloor))^2$
return y

a) Show by strong induction that the above algorithm computes a^n.

b) Write down the worst-case complexity function $T(n)$ as a divide and conquer relation.

c) Give a big-O estimate for the worst-case complexity. Explain your answer.

Note that you can solve parts (b) and (c) without solving part (a).

4. **a) Process synchronization.**

Consider three processes P, Q and S that start executing concurrently. Process P updates shared data item d1 (code part A), process Q reads the data item d1 (code part B), and process S updates the data item d1 (code part C). You are asked to describe how these processes can use **semaphores** to ensure that the code parts A, B, and C are always executed in the following order: 1. A, 2. C, 3. B

(i) Indicate the semaphores that you use and their initial values.

(ii) Fill in the pseudo-codes for these processes by showing how they use the semaphores.

```
P {

update.d1(); // code part A

}
Q {

read.d1(); // code part B

}
S {

update.d1(); // code part C

}
```

b) Processor scheduling.

You are given the following processes to execute on a single processor with **round-robin scheduling**. All processes are assumed to arrive at time 0 and they enter to the process ready queue in the order P1, P2, P3, P4. Processes' CPU burst times (in msec) are given below. Recall that CPU burst time refers to the amount of time the process uses the processor.

Process	CPU Burst time (msec)
P1	3
P2	1
P3	4
P4	3

(i) Calculate **average turnaround time** for the time quantum value q = 2. You can take the context-switch time as 0. Recall that turnaround time of a process is the total time taken between the arrival of the process for execution and the completion of the process execution.

(ii) For the time quantum value q = 2, show the **preemptions on the Gantt chart**, and give the **number of preemptions** for the schedule. Recall that a Gantt chart illustrates processes' schedule in time, and a preemption refers to the interruption of a process involuntarily (that is, without its cooperation) in order to execute another process.

5. Given an array A of n numbers, we call a number x the *frequent element* of A if it occurs at least $0.7n$ times in A (i.e., for at least 70% of values for $1 \leq i \leq n$ we have $A[i] = x$). Note that not all arrays have frequent elements and each array can have at most one frequent element. The *mode* of A is the value that occurs most frequently in A. For these algorithms, you do not need to provide a fully-working pseudocode. Instead, you can coarsely describe the algorithm.

   a) Give an algorithm that finds the *mode* of A in time $O(n \log n)$.

b) Give an *expected* $O(n)$ time *deterministic* algorithm that finds the *frequent element* of A or reports that none exists, but always returns the correct answer.

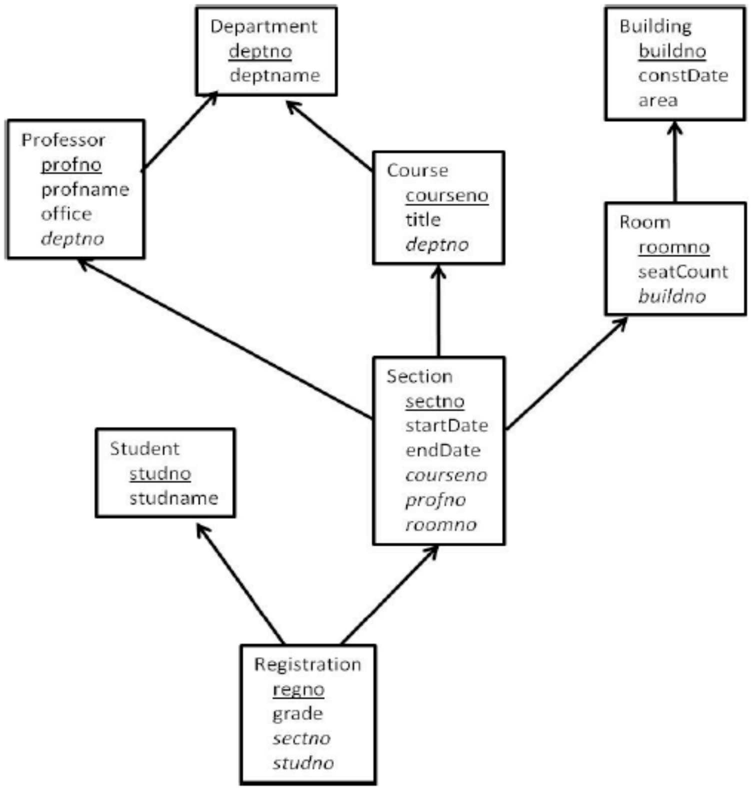
6. Suppose that you are given a class `Car` that extends class `Vehicle` and a class `Apartment` that extends class `RealEstate`. Neither `Vehicle` nor `RealEstate` have a method named `getTax()`. But classes `Car` and `Apartment` both have a method named `getTax()` that returns a `double`.

   We would like to have a class `TaxUtil` with a static method `getTotalTaxes()`. This method takes an argument that is an array of `SomeType` objects, such that when we call `TaxUtil.getTotalTaxes(SomeType[] taxableItems)`, it returns the total of the quantities returned by the `getTax()` method of each object in `taxableItems`.

   Use a diagram to arrange `Car, Vehicle, Apartment, RealEstate,` and `SomeType` into a type hierarchy. Show the important parts of the code for `Car`, `RealEstate` and `SomeType` that make the above possible. Show the implementation of the method `TaxUtil.getTotalTaxes()`.

7. We have enormous amount of code that was written by a software engineer who thought the language used call by value, but now he has found out that the language uses call by reference. Now he wants to write a translator that will convert his code to produce the intended behavior. Describe how the translator should work. In particular describe what should be modified in the source language, and how the translator can accomplish the desired conversion. Make sure your explanation is not a mere restatement of the task, but includes specifics of what changes should occur.

8. Consider the university database schema below; in which primary keys are underlined and foreign keys are italic.



a) The following query finds a weighted average of seat counts by course, in which a given room's seatCount enters multiple times in the computation, depending on how many times a section of the course has been taught in the room. Modify the query to give a simple average, in which each room's seatCount enters just once in the computation, even though the room may have hosted several sections of the course.

```
select C.courseno, C.title, avg(R.seatCount) as AvgSeats
from Course C, Section S, Room R
where C.courseno = S.courseno and S.roomno = R.roomno
group by C.courseno, C.title
```

b) Draw an ER diagram corresponding to the professor, course, room, section, registration tables of the university schema diagram, in which section appears as a weak entity. You need not construct a diagram including all the university tables, just the five mentioned here. Place arrows and participation limits as appropriate on all relationships. Choose one of the participation limits and defend it by stating a reasonable application constraint that requires those particular limits.

c) In the university schema diagram, consider just the portion dealing with students, sections, and registrations. Represent the depicted relationships in an ER diagram that does not break out registration as a separate entity. Instead, the ER diagram should show a many-to-many relationship between student and section with the appropriate information from registration recorded as attributes of that relationship. In your translation, also assume that the grade attribute is now multivalued. That is, a student receives a set of grades associated with his participation in the section. Reflect this generalization as necessary on the ER diagram. Your ER diagram should show participation limits that imply total participation of section and partial participation of student in the many-to-many relationships.

d) Map your ER diagram from the preceding question into relational tables and show the corresponding schema diagram. Account properly for the multivalued attribute.