Qualifier Exam Rules

The Ph.D. qualifier exam at Koç University Computer Science and Engineering Ph.D. Program has two steps. The goal of the first step is to test if the candidate has enough skills and background knowledge to perform computer science and engineering research. The goal of the second step is to test if the candidate can perform high-quality research and effectively communicate results, properly representing the University.

The qualifier exam must be taken within 5 semesters of the beginning of the candidate's Ph.D study and within 7 semesters if the student does not hold a master degree. The actual date may depend on the student and her/his advisor and it is highly recommended to take the exam within first 2 years of the program. Students who fail at their first attempts are given another exam on the part/parts in which they failed in the next semester. If a student fails in her/his second attempt, then the student will be expelled from the program.

Written Exam:
The written exam comprises the core computer science and engineering topics. All Ph.D. students are required to take the same exam at any given semester. The written exam will include 2 questions from each core area (8 in total). The student needs to choose and answer 6 questions to get full credit (only 6 questions will be graded). The written exam will be open book and open notes (but no digital equipment). Past questions will be made available to students as sample exams.

The written exam is generally held in the first week of December in the fall semester and first week of May in the spring semester.

Oral Exam:
In the oral exam, the candidate must present his/her research orally, and answer any questions by the committee members. Before this oral presentation, the candidate must have a paper as approved by her/his advisor, preferably either accepted or submitted to a top-quality conference. This paper must be presented to the exam subcommittee at least one week before the exam date. The oral presentation must meet the quality standards of a presentation at a top-quality conference. The oral exam involves roughly 30 minutes of student presentation, and 30 minutes of committee questions to the student. The questions asked by the committee need to be related to the student's research, but may include other topics as long as they are relevant. In particular, they may question the student's background on areas that are necessary to perform research in the student's field.

The oral exam must be taken within the same semester of the written exam upon successful completion of the written exam. It is student's responsibility to arrange locations/ date / time of the oral exam.

Exam Committee:
The exam committee involves five faculty members, including: the student's Ph.D. advisor, two faculty from Koç University Computer Science and Engineering Ph.D. Program, and two other faculty from a university other than the one they are affiliated with. The other internal members are selected by the Graduate Coordinators. Two external members should be selected by the student's advisor.

PhD Qualification Exam Process:

1. To pass the written exam, the student's score must be greater than or equal to 60.
2. To pass the oral exam, the exam committee of the student must approve by simple majority.
3. The qualifier exam has two parts. Students who are successful in the written section attend the oral section. If the student fails any of the two parts, must repeat the part/parts in which they failed in the following semester.
4. If the student fails the exam twice, the student is expelled from the program.
5. In any case, a student's exam committee may decide on additional requirements, such as additional courses.
6. Written exam topics are listed below.

Core Areas:
Below is a list of core areas for the written exam. The student must be aware that this is not a comprehensive list, but just serves as a mere starting point. Students may seek further pointers from their own advisors.

Software and Programming Concepts (related KU courses: COMP 132, COMP 301)

- Abstraction techniques
- Inheritance and polymorphism, type hierarchies
- Recursion and iteration
- Representation strategies
- Expressions, AST
- Compiler vs. interpreter, and virtual machine concepts
- Scope of a variable
- Parameter passing
- Garbage collection idea
- Stack vs. Heap storage
- Object-oriented programming: classes, objects, interfaces, abstract classes
- Arrays, utilities and collections libraries, generics

Mathematical Foundations (related KU courses: COMP 106, ENGR 200)

- Mathematical Logic (propositional and predicate logic)
- Formal Proof techniques
- Mathematical Induction
- Sets and Functions (1-1 functions, bijections, etc.)
- Relations, closure of relations, equivalence relations, connectivity relation

- Recurrence relations
- Basic number theory, modular arithmetic, Fundamental Theorem of Arithmetic

- Theory of Computation (Finite State Automata, Turing Machines)
- Probabilistic Models, Conditional Probability, Total Probability Theorem, Independence, Counting
- Discrete Random Variables: Probability Mass Functions, Functions of Random Variables, Expectation, Mean, and Variance
- Continuous Random Variables: Probability Distribution Functions, Cumulative Distribution Functions, Normal Random Variables, The Continuous and Discrete Bayes' Rule
- Joint PMFs/PDFs of Multiple Random Variables, Derived Distributions, Covariance and Correlation, Applications of the Central Limit Theorem
- Basics of Bernoulli, Poisson and Markov Processes
- Statistics (Bayesian Inference and the Posterior Distribution, Classical Parameter Estimation, Basics of Linear Regression, Binary Hypothesis Testing, Significance Testing)

Algorithms and Data Structures (related KU courses: COMP 202, COMP 446)

- Algorithm Big-Oh complexity analysis (time and space)
- Advantages and disadvantages of basic data structures (linked list, stack, queue, tree, heap, hash table, graph).
- Using graphs to represent data. Using elementary graph algorithms (BFS, DFS, shortest path, spanning tree) to answer questions on data represented as a graph. BFS vs. DFS.
- Advantages and disadvantages of various sorting (quicksort, insertion sort, heapsort, mergesort) and searching (linear search, binary search) algorithms.
- Modifying an existing algorithm or data structure to serve an additional purpose.
- Complexity classes P and NP. NP-completeness.
-

Systems (related KU courses: COMP 304, COMP 306)

- Process Management: Process concept, multiprogramming, concurrent processes, producer-consumer shared-memory solution.
- CPU Scheduling: concepts, criteria, CPU scheduling algorithms, algorithm evaluation and performance.
- Process Synchronization: race conditions, critical section problem, solutions, semaphores, deadlock and starvation, classical synchronization problems.
- Memory Management: contiguous memory allocation, internal and external fragmentation, paging, segmentation.
- Virtual Memory: demand paging, page fault, page replacement algorithms, allocation of frames, thrashing, working set model.
- ER Model Conceptual Design.
- Relational Model, SQL.

- Relational Algebra and Calculus.
- Schema Refinement and Normalization.

Additional degree requirements:

For successful graduation, at a bare minimum, each Ph.D. student must publish at least 1 high quality journal or top quality conference paper. Without being a hard limit, expected number of top quality papers is 4-5.

For successful graduation, at a bare minimum, each M.Sc. student must publish at least 1 high quality conference paper. Without being a hard limit, expected number of top quality papers is 2.