BOSTON UNIVERSITY
COLLEGE AND GRADUATE SCHOOL OF ARTS AND SCIENCES

```
dprintf("mb_delete_mob: c_get rtn: %d\n", st);
if (st == DB_LOCK_DEADLOCK)
goto retry;
if (st != 0) { /* NZ means some kind of error */
if (st != DB_NOTFOUND)
(*dcb.dc_db_err_rtn)(fn, dcb.dc_filename, st);
dprintf("mb_delete_mob: getcurrent: %d\n", st);
st1 = csr_mob->c_close(csr_mob); /* Discard the cursor. */
txn_abort(tid);
LOCK_RELEASE();
errno = st;
return ((st == DB_NOTFOUND) ? 0 : -2);
```

Department of Computer Science

What is Computer Science?

Computer Science is the systematic study of the feasibility, structure, expression, and mechanization of the methodical processes (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information, whether such information is encoded in bits and bytes in a computer memory or transcribed in genes and protein structures in a human cell. The fundamental question underlying all of computing is: *what computational processes can be efficiently automated and implemented?*

To tackle this seemingly simple question, computer scientists work in many complementary areas. They study the very nature of computing to determine which problems are (or are not) computable. They compare various algorithms to determine if they provide a correct and efficient solution to a concrete problem. They design programming languages to enable the specification and expression of such algorithms. They design, evaluate, and build computer systems that can efficiently execute such specifications. And, they apply such algorithms to important application domains.

What Computer Science Is Not…

Computer Science is not *just* about building computers or writing computer programs! Computer Science is no more about building computers and developing software than astronomy is about building telescopes, biology is about building microscopes, and music is about building musical instruments! Computer science is not about the tools we use to carry out computation. It is about how we use such tools, and what we find out when we do. The solution of many computer science problems may not even require the use of computers—just pencil and paper. As a matter of fact, problems in computer science have been tackled decades before computers were even built. That said, the design and implementation of computing system hardware and software is replete with formidable challenges and fundamental problems that keep computer scientists busy. Computer Science is about building computers and writing computer programs, *and much much more!*

Why Computer Science?

In 1943, Thomas J. Watson, Chairman of IBM declared: *"I think there is a world market for maybe five computers."* A few billion computers later, there is a temptation to fall into Watson's embarrassing underestimation of the potential that computing may have on our society. Indeed, in a few decades, *"one computer per capita"* may sound as outrageous as a *"world market of five computers"* sounds today. Computer scientists envision a world in which computing is pervasive and seamless. The golden age of computing (and of computer scientists) has barely begun.

Students choose to major in computer science for a variety of reasons. Many of our students graduate to rewarding computer-related careers in software engineering, system administration and management, research and development in industrial and governmental laboratories. And, since computer technology has transformed almost all disciplines, many of our graduates use their computer science major (and the analytical skills it instills) to prepare them for a career in other disciplines such as medicine, law, education, physical and life sciences, social sciences, and humanities. Demand for graduates well-versed in computer science is high and is expected to continue to grow as the information age comes of age!

over

How Does Computer Science Relate To Scientific Computing?

Computers and software artifacts have become indispensable tools for the pursuit of pretty much *every* scientific discipline. The use of computers has enabled biologists to comprehend genetics, has enabled astrophysicists to get within femtoseconds of the big bang's initial conditions, and has enabled geologists to predict earthquakes. It is not surprising, then, for scientists in these disciplines to increasingly rely on a computational methodology (in addition to traditional mathematical or empirical methodologies) to make advances in their respective fields of study. Such scientists are often referred to as computational scientists. So, a computational chemist is a scientist who uses computers to make contribution to chemistry, just as a mathematical physicist uses mathematics to model atomic dynamics, or an empirical biologist uses a microscope to observe cellular behaviors. And, just like all of these scientific disciplines, advances in computer science itself often rely on the use of computers and computational processes. In that sense, among all scientific disciplines, Computer Science is unique. It is the only discipline which fuels its own advancement. Indeed it is a *recursive* discipline!

How Does Computer Science Relate To Computer Engineering?

The realization of a computing system, subject to various physical and technological constraints, is a challenging undertaking that requires a great deal of knowledge about the functionality and characteristics of the building blocks available at our disposal using today's technologies (e.g., semiconductor technologies, optical communication technologies, wireless signaling technologies, etc.) Computer engineering concerns itself with current practices in assembling hardware and software components to erect computing engines with the best cost-performance characteristics. In contrast, computer scientists worry about the feasibility and efficiency of solutions to problems in a manner that is less dependent on current technologies. As such, computer scientists work on abstractions that hide details of underlying implementations to enable the construction and comprehension of yet more complex systems. The creative process of developing, implementing, and evaluating computing abstractions is what pushes the frontiers of what computers and computations can do. For example, the pervasive use of the Web in our society is a direct result of our ability to free Internet application developers from the lower-level implementation details of moving bits and bytes over wires from one point to another. Similarly, the tremendous advances in the use of computer animation are a direct result of our ability to free programmers from having to worry about lower-level digital signal processing techniques.

What Does It Take To Be A Successful Computer Scientist?

Computer Science is about problem solving. Thus, the qualities of a good computer scientist include a passion for finding elegant solutions, an ability to use mathematical analysis and logical rigor to evaluate such solutions, creativity in modeling complex problems through the use of abstractions, attention to details and hidden assumptions, an ability to recognize variants of the same problem in different settings, and being able to retarget known efficient solutions to problems in new settings. If you like to solve puzzles, then computer science is for you!

Department Contact Information

Ms. Jennifer Streubel, Program Coordinator
Department of Computer Science
Boston University
111 Cummington Street
Boston, MA 02215

Tel: 617-353-8919
Fax: 617-353-6457
Email: csdept@cs.bu.edu
Http://www.cs.bu.edu

Spring 2003