IMACS Curriculum Guide

Computer Programming Enrichment & University Computer Science

Table of contents

Background

In the late 1980s, as part of a government-funded program, members of the IMACS Curriculum Development Group (CDG) were in the process of designing a special computer science curriculum based on new developments in academic computer science. A unique functional programming language called "Scheme" had recently been invented at MIT and was starting to change the way academics were looking at certain areas of computer science education. With this advancement, the IMACS CDG saw great opportunities to refine this approach for talented secondary school students at a time when almost no secondary schools were offering rigorous coursework in foundational computer science. With support from eminent American computer scientists, CDG researchers realized that these exciting developments could be used to teach advanced computer science concepts to much younger students than had ever been possible before. The resulting curriculum forms the core of the university-level IMACS Computer Science curriculum, and it was soon adopted by special programs for talented secondary school students such as Duke University's Talent Identification Program (TIP) and the Johns Hopkins Center for Talented Youth (CTY) program.

However, IMACS developers realized that the rigor of this new curriculum put it beyond the capabilities of students who are too young to have developed sufficient study skills. Yet young students of elementary and middle school age are often just as enthusiastic about computers and computer science as their older counterparts. So the IMACS CDG set about developing a curriculum that introduced the important concepts of computer science in a way that younger students would find irresistable. The result is a curriculum that introduces students to programming using the pedagogical language Logo. Through a series of carefully designed activities using exciting teaching tools, students are introduced to the essential concepts in computer science. Because each activity is so engaging, students hardly realize the power of what they are learning. Yet on completion of the IMACS Computer Enrichment series, students

have the necessary foundation to tackle more advanced programming courses, as well as academic university-level computer science courses such as the IMACS Computer Science program.

Our courses

Computer Programming Enrichment

In the IMACS Computer Enrichment program, students are introduced to the programming language Logo. Like Scheme, one of the languages used in the IMACS Computer Science curriculum, Logo was developed at MIT. It allows students to control a vector graphics device called a "turtle", which students program to draw a variety of geometric shapes.

While Logo's graphical output is both intuitive and fun for students, it simultaneously provides a simple but powerful framework for building up to advanced concepts from ground level. As students work through self-paced activities involving Logo and the turtle, they are introduced to increasingly sophisticated concepts from modern computer science, including variables, libraries, generality, and control constructs like iteration and recursion. From the very beginning of the course, students are encouraged to think like professional computer scientists as they write their own programs to solve complex, multi-stage problems. Through their work, students are also introduced to essential mathematical concepts in areas such as algebra and geometry that are useful in a wide variety of applications.

Benefits to Students

The IMACS Computer Enrichment program is a computer programming course, teaching the students how to control the computer and how to program it to do their bidding. The course was created by recognized experts in the field, backed by years of experience of providing an up-to-date, world class computer science curriculum to talented students.

The course also provides much more than simple "coding" classes most often offered to younger students. While these coding classes typically simply teach students how to use premade libraries in a particular programming language, IMACS computer programming classes teach broader conceptual skills that will provide a foundation for any language a student may go on to study.

While coding is increasingly taught to students in secondary school, most students must wait until college to be introduced to the fascinating world of modern computer science. By introducing these ideas early, young students are given the thinking and problem solving tools that set them apart from their peers.

We teach computational thinking and reasoning that students can always rely on to find solutions, no matter which programming language they happen to be using. These timeless

skills are extremely helpful when learning more advanced languages. Consequently, the program is the ideal introduction to the more rigorous IMACS Computer Science program, in which students as young as sixth grade are introduced to college-level material.

Computer Science

The IMACS Computer Science program consists of a series of university-level computer programming courses designed to take young students into the field of modern academic computer science. High school students completing the advanced courses typically place directly into upper level computer science classes at the very best universities.

University Computer Science

During the late 1980's and early 1990's, developers in the IMACS Curriculum Development Group (CDG) along with academics from across the country were investigating the educational possibilities of new developments in the theory of programming languages. While universities across the world were considering the effects of the new developments on their college curriculums, the CDG focused its attention on the opportunity this offered to middle and high school students.

The result of this research effort was a seminal textbook written by senior members of the CDG, called *The Schemer's Guide*. First published in its English language edition in 1990, the textbook has been used at high schools and at the college freshman level across the world and has been adopted by many programs designed to introduce talented students to modern computer science, programs such as Duke University's Talent Identification Program (TIP) and at the various Centers for Talented Youth (CTY) programs directed by the Johns Hopkins University. At IMACS, we have kept the core of the original program, which is powerful precisely because of its generality, while updating the curriculum as needed to reflect the latest developments in the field.

Together with supplementary material written for IMACS by the authors, *The Schemer's Guide* forms the core component of the IMACS University Computer Science program. Its approach to computer science pedagogy is to utilize a syntactically simple programming language with very few built-in functions, guiding students to build their own more complex functions and extensive libraries based on these. This provides students a much deeper understanding of how computer science functions at its core than most other coding courses for secondary school students.

So often, the study of computer science is made especially difficult by the requirement that students first learn a programming language such as Python, Java, or C that are widely used in industry. Such languages are syntax-heavy, filled with arbitrary rules and regulations that deliberately constrain the programmer and make learning the language a challenging but often tedious task. IMACS instructors use the Scheme language's mathematical simplicity to shortcut this process. Using techniques invented by the IMACS CDG, the task of teaching the IMACS student to program is achieved very quickly. This allows students to focus on the broader

theoretical underpinnings of computer science, applicable to any programming language they might later learn.

In fact, the IMACS student is able to progress so quickly by these methods that following the first UCS course they are well prepared for the second course, which guides students to quickly and easily translate the foundational knowledge learned in Scheme to other languages, such as Haskell and Python. The third course in our sequence is AP Computer Science A, which teaches Java. Our students typically find this course to be significantly easier than the first two and are well prepared for the relevant AP exam.

Benefits to Students

Unlike the vast majority of courses offered to students of secondary school age, the IMACS Computer Science courses provide students with a deep understanding of the computer science concepts that underpin computer programming. This is achieved through a curriculum refined over many years designed and taught by a faculty with advanced qualifications and extensive experience. Consequently, graduates of the IMACS university-level Computer Science courses are characterized by their ability to learn any programming language in very short order.

Many years teaching computer science by this means have produced a wealth of IMACS graduates who have gone on to great achievements in computer science. Such students have been heavily recruited by the best universities for many years. Some have earned Master's degrees during their four years of undergraduate study. Several IMACS graduates now work for Google and Microsoft, others have worked for NASA and similar highly respected institutions. When students who have completed a significant portion of the advanced computer science program reach the stage of applying to colleges, the Institute for Mathematics and Computer Science sends transcripts from our courses directly to schools. For students who have excelled in these classes, we will also provide strong letters of recommendation on request.

For testimonials from former IMACS students who have gone on to highly successful tech careers, please check out these blog posts written by tech entrepreneur Steve Krouse and computer science professor Ryan Newton.