

Application Development Framework

Application Express

Big Data

Business Intelligence

Cloud Computing

Communications

Database Performance & Availability

Data Warehousing

Database

.NET

Dynamic Scripting Languages

Embedded

Digital Experience

Enterprise Architecture

Enterprise Management

Identity & Security

Java

Linux

Mobile

Service-Oriented Architecture

Solaris

SQL & PL/SQL

Systems - All Articles

Virtualization

Teaching Mobile Computing to Generation C: A Conversation With Java Champion Qusay Mahmoud

By [Janice J. Heiss](#), July 2008

[Interviews Index](#)

This series of interviews spotlights [Java Champions](#), individuals who have received special recognition from Java developers across industry, academia, Java User Groups (JUGs), and the larger community.

Bio: Java Champion [Qusay H. Mahmoud](#) is an associate professor in the department of computing and information science and founding director of the Centre for Mobile Education Research at the University of Guelph in Canada. He holds a Ph.D. in computer science from Middlesex University (UK), as well as an M.Sc. in computer science and a B.Sc. in data analysis, both from the University of New Brunswick (Canada). Mahmoud is the author of [Learning Wireless Java](#) (O'Reilly) and [Distributed Programming With Java](#) (Manning Publications). He was named a [Java Champion](#) in 2007.

JavaSunCom (JSC): How are mobile devices currently used in the computer science curriculum?

Mahmoud: Few computer science departments offer undergraduate courses in mobile computing, and those that do, focus on the theory of wireless networks and protocols. But given the proliferation of mobile devices and wireless sensors, the CS curriculum needs to make mobile devices and mobile application development a core topic.

This is important because students in CS courses usually develop their programming assignments on a platform similar to the one they will be tested on. However, Java ME applications and services are developed on one platform, such as Windows or Linux, and deployed and run on a totally different platform, such as a cell phone. This multiplatform approach introduces students to different programming models.

In the summer of 2006, my department, with the help of colleague Allan Dyer, received some 200 secondhand BlackBerry wireless devices through the Toyota Canada Recycling Program. I borrowed 15 of them to use in two courses, Computer Programming I (CS1) and User-Centered Programming (CS3), which focuses on developing interactive graphical applications.

The devices had no operating system and didn't boot. After some research, I was able to download the latest BlackBerry operating system and upload it onto them. When I started one of them, I was thrilled that [Java ME MIDP/CLDC](#) is supported by the [BlackBerry](#).

JSC: What do you like about Java ME, and how have you used it in your courses?

Mahmoud: I've been working with Java ME since early 2000, and I published one of the first books on the topic, [Learning Wireless Java](#). I also developed an agent platform for mobile devices as part of my Ph.D. thesis.

I like the fact that [Java ME](#) is even simpler than [Java SE](#). I like its open-source license and perhaps most of all, its deployment on millions of phones and smart PDAs, including the BlackBerry. In fact, Java ME empowers 85 percent of today's mobile devices.

"Developing new applications for their own mobile devices motivates and inspires students. I see the results on my students' faces."

In CS1, I now spend six hours of lecture and six in labs working with integrated mobile devices. In CS3, I spend nine hours in each. I introduce students to the application-development life cycle for mobile devices using Java ME. Students learn about the BlackBerry device and Java ME MIDP/CLDC.

Developing new applications for their own mobile devices motivates and inspires students. I see the results on my students' faces. Allan Dyer and I have written extensively about this in a (PDF) paper, [Integrating BlackBerry Wireless Devices Into Computer Programming and Literacy Courses](#).

Qusay Mahmoud
Java Champion

For example, in my CS3 course, one of the first topics I cover is multithreading. I introduce students to the classic [producer-consumer problem](#), but they don't seem very excited about it -- the look on their faces tells me they want something more modern. When I show them how threads can be used to handle network connections and deal with user permissions on an actual physical device, they perk up, and it all begins to make sense.

Teaching With the Sun Java Wireless Toolkit

JSC: You also use the [Sun Java Wireless Toolkit \(WTK\)](#).

Mahmoud: I chose the Sun Java Wireless Toolkit for its simplicity. I love it, and I have been using it since its first public release in 2000. Before that, I had to preverify classes using a command-line tool and write Java Application Descriptor (JAD) files by hand! The WTK provided the simple intuitive interface I was looking for in a tool for beginners.

In my experience, the high failure rate in CS1 is not just related to problem solving and programming skills -- the wrong programming tools can themselves cause some students to fail. For this reason, I only use the standard [Java SE SDK](#), with command-line tools and a simple editor, in my CS1 course. I also make use of [DrJava](#), which is a great tool for interactive expression evaluation, among other things.

How Students Develop Java ME Apps on a BlackBerry

JSC: Take us through the steps that students follow in developing Java ME apps for the BlackBerry.

Mahmoud: First, they use the WTK to develop their Java ME applications, and build and test their applications into the emulator. Then they use the `rapc` tool that comes with the RIM Java Development Environment (JDE) to convert the Java ME application, `.jar` and `.jad`, into a BlackBerry application, `.cod` file. Next, they load the generated `.cod` file into the BlackBerry simulator and test the application. Then they load the generated `.cod` file using a USB cable into an active BlackBerry physical device and test it.

In addition to the WTK emulator and the BlackBerry simulator, I provide students with BlackBerry devices to experiment on, but their experience is not limited to BlackBerry devices. The Java applications they develop can run on any Java-enabled device.

Once students have learned the basics of Java ME using the WTK, it's safe to let them experiment with a full-blown IDE such as NetBeans and the NetBeans Visual Mobile Designer, and RIM's Java Development Environment.

The Centre for Mobile Education Research

JSC: What are the goals of the [Centre for Mobile Education Research \(CMER\)](#), which you helped create?

Mahmoud: CMER is generously funded, in part, by Research In Motion (RIM). Its mission is to engage in cutting-edge research to develop state-of-the-art applications and services to facilitate and enhance mobile education and learning, and to provide leadership in integrating mobile devices into the CS curriculum.

We do this through research projects, partnerships, and technology transfer. We want to offer leadership in revitalizing computer science education and provide practical improvements in teaching. We are designing and developing educational tools, pedagogical methods, and an academic kit that will be available in the fall of 2008 to make it easier for instructors to prepare material and reduce the time and energy investment required. The academic kit will provide a collection of instructor resources, such as tutorials, sample code, presentation slides, lab description and experiments, sample quizzes and assignments with solutions, and case studies.

We'll also be organizing design and discovery events for high school students and engaging teachers in the process. We'll offer workshops for CS and engineering educators and organize outreach activities. Put simply, CMER will define what happens next in mobile devices in the CS curriculum.

JSC: What advice do you have for computer science departments that might want to integrate mobile devices into their curriculum?

Mahmoud: It's a challenge because it depends on the availability of faculty to teach it, but the CMER academic kit will help. Here's some advice for integration strategies:

Introduce mobile devices and the basics of mobile application development as early as possible in introductory programming courses.

Introduce the technology as a lab component in any core CS course.

Introduce the technology into core CS courses at the intermediate and advanced level, such as courses in data structures, operating systems, and database systems. All of these courses provide opportunities to explore mobile devices and their applications from design and practical points of view.

The variety of operating systems available on mobile devices makes for a good use case for a course on operating systems. In a database course, students could explore how to access remote databases securely, how to store information in a local database on the device, and how to interact with existing database-driven applications, such as personal information management (PIM) data.

The discussion of application design and services for mobile devices fits nicely into software engineering and human-computer interaction courses. Many software engineering courses include projects where students working in teams design and develop large software systems. Students need to be aware that if their software systems are going to support mobile devices, then design needs to be integrated into the project at the beginning and not as an add-on.

Introduce the technology into project-based courses in network programming, distributed systems, and so on. Such courses could be at the second-, third-, or fourth-year levels.

Introduce the technology in senior capstone projects for graduating computer science majors.

So far, I have integrated mobile devices and mobile application development in CS1, CS3, Distributed Programming II, which is a project-based fourth-year course, and capstone projects.

In the Distributed Programming II course, for example, a team of students created a multiplayer game for the BlackBerry. In the capstone projects, one student designed and developed a Java ME game -- a derivative of the Hexxagon game, whose motto is "It takes a minute to learn and a lifetime to master" -- and another student designed and developed a puzzles suite -- including Sudoku, crossword, and word search -- for the BlackBerry.

Teaching Computer Science Concepts Using Mobile Devices

JSC: What kinds of topics related to mobile devices should be covered in the computer science curriculum?

Mahmoud: This is an excellent question. Mobile devices and application development represent a multidisciplinary area that includes programming, design, software engineering, human-computer interaction, web programming, security, networking, and a number of other traditional computer science areas.

Here are some key topics that could be taught:

- Basic understanding of mobile application development
- Mobile platforms and desired features of future platforms
- The mobile device user interface
- Software architecture issues
- Software engineering for mobility
- Networking and security issues
- Databases for mobile applications
- Software adaptability
- Games for mobile devices
- Location-based services
- Service discovery and interaction
- Provisioning of mobile applications and services
- Business models and billing
- Mobile devices in vertical markets such as healthcare

Generation C: Content, Cash, Choice, and Control

JSC: What can you say about how the use of mobile devices is influencing the minds and assumptions of a whole generation of students?

"This generation of students is becoming known as Generation C -- it's about content, cash, choice, and control."

Mahmoud: Today's generation of students is becoming known as [Generation C](#) -- it's about content, cash, choice, and control.

Surveys show that SMS (Short Message Service) is the most widely used nonvoice application on mobile devices, but games are the most widely downloaded applications. Students are attentive to their mobile devices simply because they are inherently personal, with ring tones and themes.

Qusay Mahmoud
Java Champion

CS students will benefit from this experience as it enables them to design mobile applications that are friendly and useful, and ones that may generate cash for them too.

Don't Lecture Too Much

JSC: When I asked fellow Java Champion [Cay Horstmann](#), a professor of computer science at San Jose State University, the biggest mistake that computer science teachers make, [he said](#), "They lecture too much. I think it's horribly ineffective to lecture for 50 or even 75 minutes without giving the students a chance to try out what they're learning. These days, all my students have laptops and we have a 15- to 20-minute lecture, a lab, another mini-lecture and lab, and a five-minute wrap-up." Your response?

"I'm beginning to see that computer programming will be taught in mainstream secondary and even elementary schools, just like reading and writing."

Mahmoud: I agree. For the past six years, I've taught at the [University of Guelph-Humber](#), where we emphasize hands-on learning, which enhances experiential learning. Every course in our curriculum has a lab component, which is essential in preparing students for jobs in the IT market, as well as programs with cooperative education.

JSC: Do you have any visionary ideas about how computer programming might be taught in the future as technology evolves?

Qusay Mahmoud
Java Champion

Mahmoud: I'm beginning to see that computer programming will be taught in mainstream secondary and even elementary schools, just like reading and writing. In fact, we are already starting to see tools that help teachers with this task, such as [Scratch](#), [Alice](#), and [Greenfoot](#).

We have recently developed a Greenfoot scenario, [Going to the Moon](#), to teach computer programming to high school students as early as grade 10. But tools such as Scratch can be used with grade 1 students -- it's great for storytelling.

I demonstrated Scratch to my son, who is six and in first grade, and he was able to create a simulation -- with animations and voice messages -- of the tag game that he plays with his friends at school in less than 20 minutes.

Faculty Summit on Mobile Devices in CS Education

JSC: You organized the first [Faculty Summit on Mobile Devices in CS Education](#), which took place on June 12, 2008. How did it go?

Mahmoud: First, I'm grateful to RIM and Sun Microsystems for sponsoring this event. Though the target audience was CS professors from universities in the greater Toronto (Ontario) area, we had attendees from the U.S. as well as high schools. The event was a success, with students offering demos of their work. The presentations, posters, and photos from the event are [now available](#).

We hope it will become a premier event for academics interested in integrating mobile devices in the CS curriculum. We plan to repeat it again next year as well as at various locations around the world.

JSC: What are some fallacies or misconceptions that you encounter regarding Java ME?

Mahmoud: First, I recommend that Java ME developers become familiar with the [eight fallacies](#) of distributed computing that were developed by people at Sun in the mid-1990s. These are common but flawed assumptions made by distributed application developers that are very relevant in distributed mobile applications:

The network is reliable.
Latency is zero.
Bandwidth is infinite.
The network is secure.
Topology doesn't change.
There is one administrator.
Transport cost is zero.
The network is homogeneous.

As for Java ME-specific fallacies, the first fallacy is that Java ME is meant to compete with or replace browser-based mobile applications. Developing browser-based mobile applications using a markup language such as WML or XHTML is easier than developing with Java ME, but Java ME is meant to open up mobile devices and let consumers download more applications. Java ME is perfect for interactive applications.

Second fallacy: Java ME is complex. Just because it allows you to develop applications for mobile devices, that doesn't mean you have to be an expert at embedded application development.

The third fallacy is that Java ME is an operating system. Just because it powers 85 percent of the mobile devices on the market today doesn't make it an operating system. It's the most ubiquitous mobile application platform.

Finally, as with any standard, different interpretations of the specifications exist, despite the availability of testing compatibility kits. The various Java ME implementations, devices, and device capabilities have resulted in device fragmentation problems, whereby a Java ME application doesn't behave the same on all Java ME devices -- even on multiple devices from the same vendor. I don't see this problem disappearing anytime soon. Developers just have to do more testing.

Go for It!

JSC: Do you have any final advice for computer science professors who are considering using mobile devices in their teaching?

Mahmoud: Go for it -- because your students will love you for doing it, and it will help them with their careers.

Every computer science career path today requires some knowledge of mobile devices. System administrators need to configure email servers for mobile users. Programmers need to build applications to run on mobile devices. Web developers need to port an interface for mobile users. And project managers must understand the cost and benefits of porting a system to a mobile device.

"Don't be afraid of Java ME if you have no experience with it -- it's just Java with a very limited set of APIs."

And as mobile devices become more powerful and service providers feel pressured to offer compelling applications beyond games, there's a growing need for innovation.

Qusay Mahmoud
Java Champion

There are costs involved in using mobile devices in courses: In addition to the device, you need SIM cards, activation fees, and data plans. But as WiFi-enabled mobile devices become more available, CS departments won't need to worry about these.

Don't be afraid of Java ME if you have no experience with it -- it's just Java with a very limited set of APIs.

See Also

[Qusay H. Mahmoud's Blog](#)
[Qusay H. Mahmoud](#)
[The Centre for Mobile Education Research \(CMER\)](#)
[Faculty Summit on Mobile Devices in CS Education](#)
[Integrating BlackBerry Wireless Devices Into Computer Programming and Literacy Courses \(PDF\)](#)
[Eight Fallacies of Distributed Computing](#)
[Java Champions Project](#)
[Java Champion Dr. Heinz Kabutz](#)
[Java Champion Cay Horstmann](#)
[Java Champion Adam Bien](#)
[Java Champion Kirk Pepperdine](#)
[Mobile & Embedded Community Blogs](#)
[Mobility Tech Tips](#)

Rate and Review

Tell us what you think of the content of this page.

Excellent Good Fair Poor

Comments:

Your email address (no reply is possible without an address):

[Sun Privacy Policy](#)

Note: We are not able to respond to all submitted comments.

Contact Us

US Sales: +1.800.633.0738

Have Oracle Call You

Global Contacts

Support Directory

About Oracle

Company Information

Communities

Careers

Cloud

Overview of Cloud Solutions

Software (SaaS)

Platform (PaaS)

Infrastructure (IaaS)

Data (DaaS)

Free Cloud Trial

Events

Oracle OpenWorld

Oracle Code

JavaOne

All Oracle Events

Top Actions

Download Java

Download Java for Developers

Try Oracle Cloud

Subscribe to Emails

News

Newsroom

Magazines

Customer Success Stories

Blogs

Key Topics

ERP, EPM (Finance)

HCM (HR, Talent)

Marketing

CX (Sales, Service, Commerce)

Supply Chain

Industry Solutions

Database

MySQL

Middleware

Java

Engineered Systems