

Course Proposal: CMSC 436 - Programming Handheld Systems

Evan Golub and Adam Porter

Background: Over the last 2 semesters we and others have developed and offered special topics courses on Programming the iPhone (CMSC 498I) and on Programming the Android platform (CMSC 498G). We are now proposing to broaden this course to general handheld systems and to add it to our official course catalog. Several faculty have expressed interest in teaching this course, including Ashok Agrawala, Ben Bederson, Evan Golub and Adam Porter.

Catalog Description:

CMSC 436: Programming Handheld Systems, (3) Prerequisites: CMSC 330

This course teaches fundamental principles and concepts that underly the programming of handheld systems, such as mobile phones, personal digital assistants, and tablet computers. In particular, it emphasizes areas, such as limited display size, limited power, memory and CPU speed, and new input modalities, where handheld systems differ substantially from non-handheld systems, and, thus require special programming tools and approaches. Students will apply these concepts and principles in the context of an existing handset programming platform.

The following sample syllabus lists key topics covered in this course. Implementation details will differ depending on the subject handheld programming platform, such as iPhone, Android, Windows Mobile or BlackBerry, used.

Sample Syllabus:

1. Introduction to handheld systems. What are handsets? How are they different from desktop devices? What design constraints do application developers face?
2. Execution platforms. Hardware/Software Stack. Overview of Operating system, runtime environment, core & application libraries and device characteristics.
3. Development environment & tools. Key programming tools and hardware including integrated development environments, emulators, debuggers and device integration support.
4. Application programming components. Discussion of the key software building blocks used in programming the subject platform.
5. Connectors for application programming components. Strategies and protocols for integrating components into complete applications.
6. User interface elements and design principles for handheld systems. User interface layout, classes and usability principles.
7. Event-based programming. Asynchronous programming & event handling. Examples include touch event management, event delegation, notifications & alarms.
8. Data Management. On-device data bases, data persistence, file systems & XML.
9. Multi-threading. Creating applications using multiple threads & accompanying performance issues.
10. Graphics. Classes to support 2D & 3D graphics and animation.
11. Networking. Network services and protocols, including TCP sockets, web-based applications, and Bluetooth support.
12. Sensors. Using device-based sensors including accelerometers, gyroscopes, location services and maps.
13. Multimedia: Audio, video and image processing.