

Advanced Java SIM Applet Development

Learn how to design, develop, optimize & validate real VAS Toolkit applications



Creating new services has until now required long development cycles and proprietary implementation. Using Java Card based SIM cards and Gemalto **Development & Simulation Tools**, based on the latest ETSI standards; you can now work in-house, and transform ideas into applications in a matter of days. This training course will allow you to benefit from Gemalto's wide experience in **designing and deploying applications on the field**, and rapidly begin developing and validating your own services.



At the end of the training you will

- > Be able to design, write & validate 'interoperable' STK applets.
- > Understand the constraints of designing applications in Java for a smart card
- > Be able to optimize your source code to gain in time and speed and robustness
- > Optionally:
Be able to design & run a Test Plan before large scale deployment. To validate application-handset-network operation.

Who should attend

- > Development Staff

Pre-requisites:

- > GSM standards 11.11, 11.14 (SIM Toolkit), 23.048.
- > A working knowledge of object-oriented programming.
- > It is strongly recommended that you follow the "Java SIM Card Administration" course before attending this course

This course is held in English

Key topics

- > class conversion
- > jar and cap files
- > 43.019 api
- > handlers
- > application design process

Course Schedule



Day 1	Practice
<p>Reminder on Java card architecture</p> <ul style="list-style-type: none"> > Introduction > Architecture > JCRE, VM <p>Description of Java card 2.1 API</p> <ul style="list-style-type: none"> > javacard.framework > java.lang > javacard.security > javacardx.crypto <p>Description of Toolkit API</p> <ul style="list-style-type: none"> > sim.access > sim.toolkit > Developing toolkit applications > Examples of coding 	<ul style="list-style-type: none"> - Modifying source code of pre-written javacard applet. Loading & Testing in a smart card using tools - Add a new STK menu to an applet implementing a proactive command, compile / convert / load / install / test - Add a new service in an applet using the event SMS_PP and other proactive commands. - Configure an OTA communication chain to test and debug your applet.

Day 2&3	Practice
<p>Development guidelines</p> <ul style="list-style-type: none"> > Constraints of smart card environment > Optimisation of code in size and speed > Advanced Debugging techniques <p>Dummy project (1,5 days)</p> <ul style="list-style-type: none"> > The scope of this dummy project is to focus on the 43.019 API (java framework and SIM toolkit commands) > Starting from a Functional Requirement Specifications > Development of the application > Debugging using Developer Suite tool 	<ul style="list-style-type: none"> - Optimise your applet code. - Development of the application - Debugging using Developer Suite tool & advanced techniques

Optional - Day 4	Practice
<p>Introduction to validation method for SIM card</p> <ul style="list-style-type: none"> > How to build a Test Policy Plan <p>Integration Test Method for SIM Toolkit application</p> <ul style="list-style-type: none"> > How to build an Integration Test Plan 	<ul style="list-style-type: none"> - Define a validation strategy for a SIM toolkit application - Defining a test policy plan and a Integration test plan - Usage of Views Detective tool for integration tests

