symbian Academy

Introduction to Symbian OS

Introduction Part 1a

Introduction

This lecture serves as general introduction to the course

- The background of Symbian the company
- Symbian OS C++ dialect
- The architecture of Symbian OS
- What is covered in the course
- Resources

Background

Before Symbian - Psion

Psion was established in 1980

- To develop games and software for the Z80-based Sinclair ZX81 and ZX Spectrum

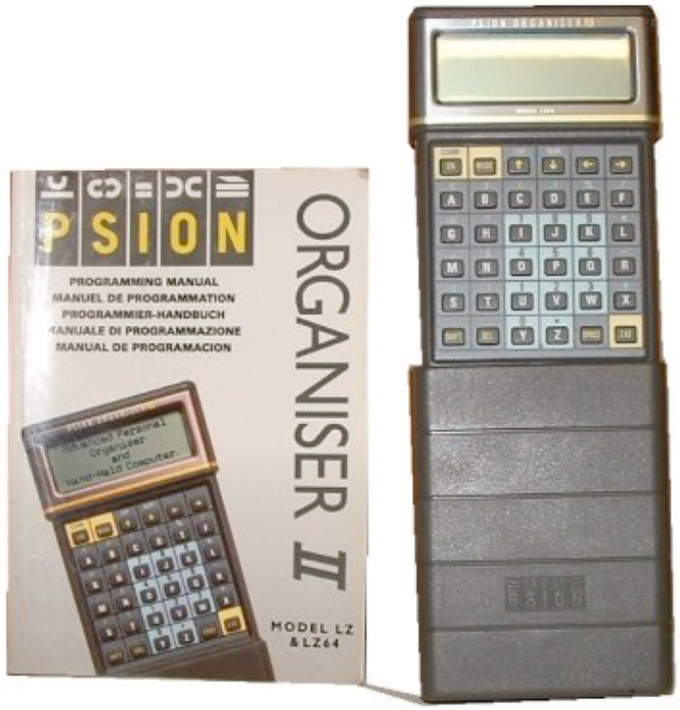
Products included

- Flight simulator, "Horace Goes Skiing", Psion Chess, Space Raiders and other games.
- Psion Chess for the ZX81 took 1K memory
- Office suite for the Sinclair QL (1983)

Before Symbian - Psion

Mid '80s - Hardware

- The 6301 based 8-bit Organiser (1984) - 14cm x 9cm, 2K RAM, 4K ROM, 8K datapak, 6 months battery life!
- Organiser II (1986) - the world's first proper PDA
- Organiser II variants were created: up to 64K r hardware
- Half a million sold over a decade



Before Symbian - Psion

1989 - MC400 laptop

- 16-bit "EPOC" Operating System - multi tasking, GUI, built-in apps
- Exceptional power management, screen technology

1991 - Series 3, 1993 - Series 3a, 1996 Series 3c/Siena, Series 3mx

- Clam shell organisers, built in apps, OPL, up to 2MB memory
- Built on EPOC
- Over 1.5 million units sold

1997 - Series 5

- 32 bit OS "EPOC32" (EPOC -> EPOC16 -> SIBO)
- Slide out QWERTY keyboard, touch screen, 4MB/8MB
- Implemented using C++

http://3lib.ukonline.co.uk/historyofpsion.htm

Symbian History

1998 - Symbian was formed by Psion, Nokia, Ericsson and Motorola

- Matsushita joined in 1999, Sony Ericsson & Siemens in 2002, Samsung in 2003
- Motorola share in Symbian were sold back in 2003
- Psion shares were sold back in 2004
- EPOC32 was later renamed to Symbian OS

First Symbian OS phone was released in 2000 (Ericsson R380)

- Nokia 9210 Communicator released in 2001 - this was the first "open" phone.

2.5G phones followed in early 2002

- Nokia 7650

3G phones in December 2002

- 3G FOMA F2051 from Fujitsu

Symbian History

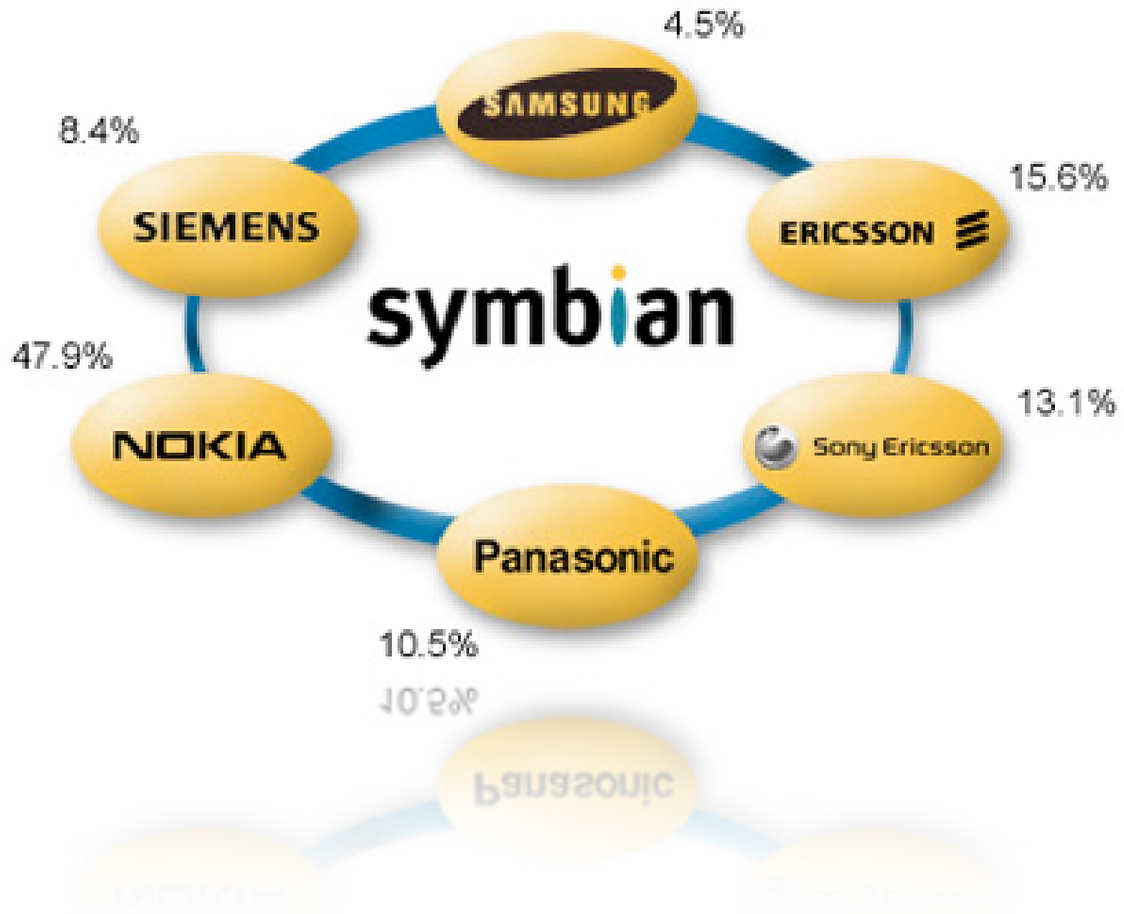
Other notable milestones

- EKA2 (EPOC Kernel Architecture 2) was announced in 2004
- Secure Platform introduced in Symbian OS v9 in 2005
- Symbian OS opens to non-mobile developer community through P.I.P.S in 2007

  P.I.P.S stands for "**P**.I.P.S **I**s **P**OSIX on **S**ymbian OS" - a POSIX compliant API

Symbian Ownership

As of the Summer 2007 Symbian shareholders:



Some Stats

As of 31 March 2007:

- A growing company - 15.9m Symbian smartphones shipped by licensees in Q1 2007, a 35.9% increase on Q1 2006 (Q1 2006 - 11.7m)
- 126 million cumulative Symbian smartphone unit shipments
- 20 million cumulative Symbian smartphone unit shipments milestone reached in Japan since first 3G Symbian model shipped in 2003 all models shipped since have been 3G
- The demand for Symbian OS C++ developers outstrips the supply!
- More stats at http://www.symbian.com/about/fastfacts/fastfacts.html

Symbian OS C++

Symbian OS C++

Symbian OS C++ is referred to as "the C++ domain specific dialect and accompanying frameworks that is used to build Symbian OS and the software that runs on it"

Evolution of Symbian C++

Symbian OS design and C++ considerations were driven by factors such as:

- Power sources (battery lifetimes)
- User responsiveness
- "Always on" nature of mobile devices
- Limited resources (RAM, ROM)
- Re-use and customization through frameworks.
- Reliable data storage
- Openness

Evolution of Symbian C++

C++ for Symbian OS was also influenced by the point in time at which the OS was developed:

- In 1994 - C++ was still evolving
- It had not yet been standardized

This affected the take-up of some of its later features:

- Templates
- Exceptions
- Namespaces
- New casts
- Boolean types
- Some were not supported by the tool chain or were immature.

Evolution of Symbian C++

Idioms were invented to fix omissions e.g.

- Leaves
- The cleanup stack
- Descriptors

Certain patterns of use were not adopted as C++ was explicitly intended as a general purpose systems language

- It was not optimized for small, low memory and low power devices

Common Paradigms

Common Symbian OS paradigms

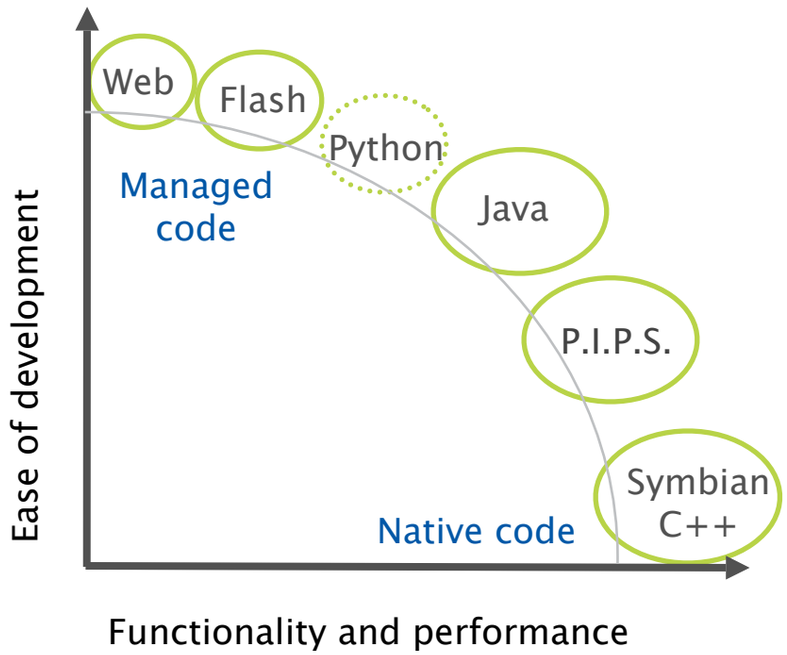
- Multithreading and pre-emptive multitasking
- Lightweight micro-kernel OS design
- Client-Server, session based IPC (among other mechanisms)
- Asynchronous services, Active Objects
- Cleanup Stack, Leaves, Traps for exception handling
- Re-usable frameworks for apps, middleware, GUIs
- Descriptors
- Naming Conventions

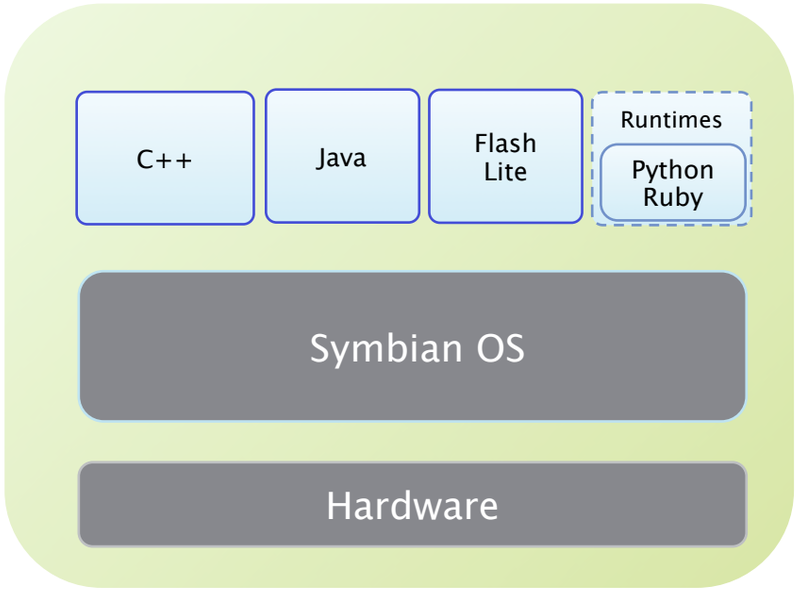
We'll discuss these, and more, in the course

Other Language Support

Symbian OS also has support for

- Java
- Flash Lite
- Runtimes such as Python, Ruby, Visual Basic
- OPL (until v9)

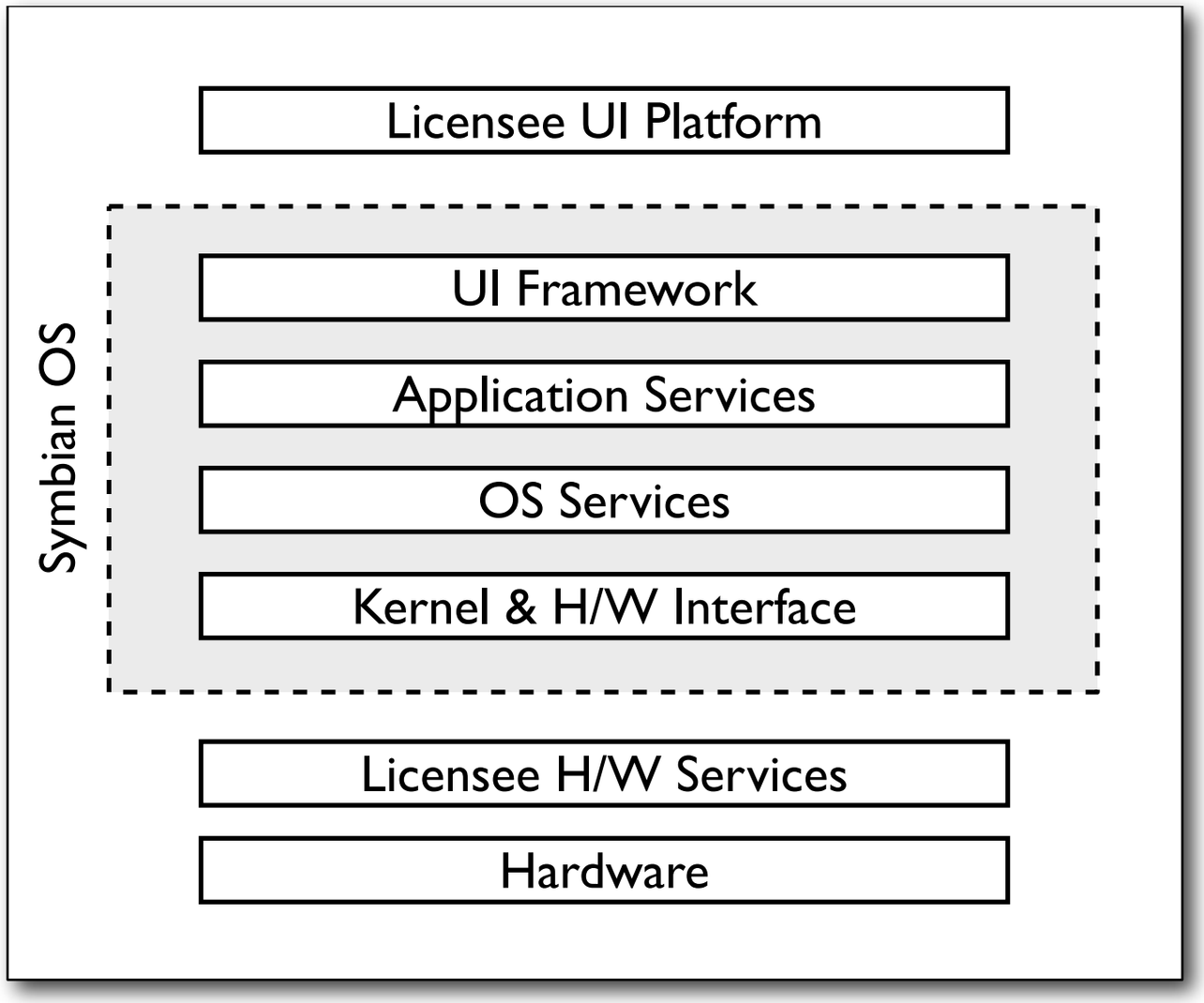




Architecture

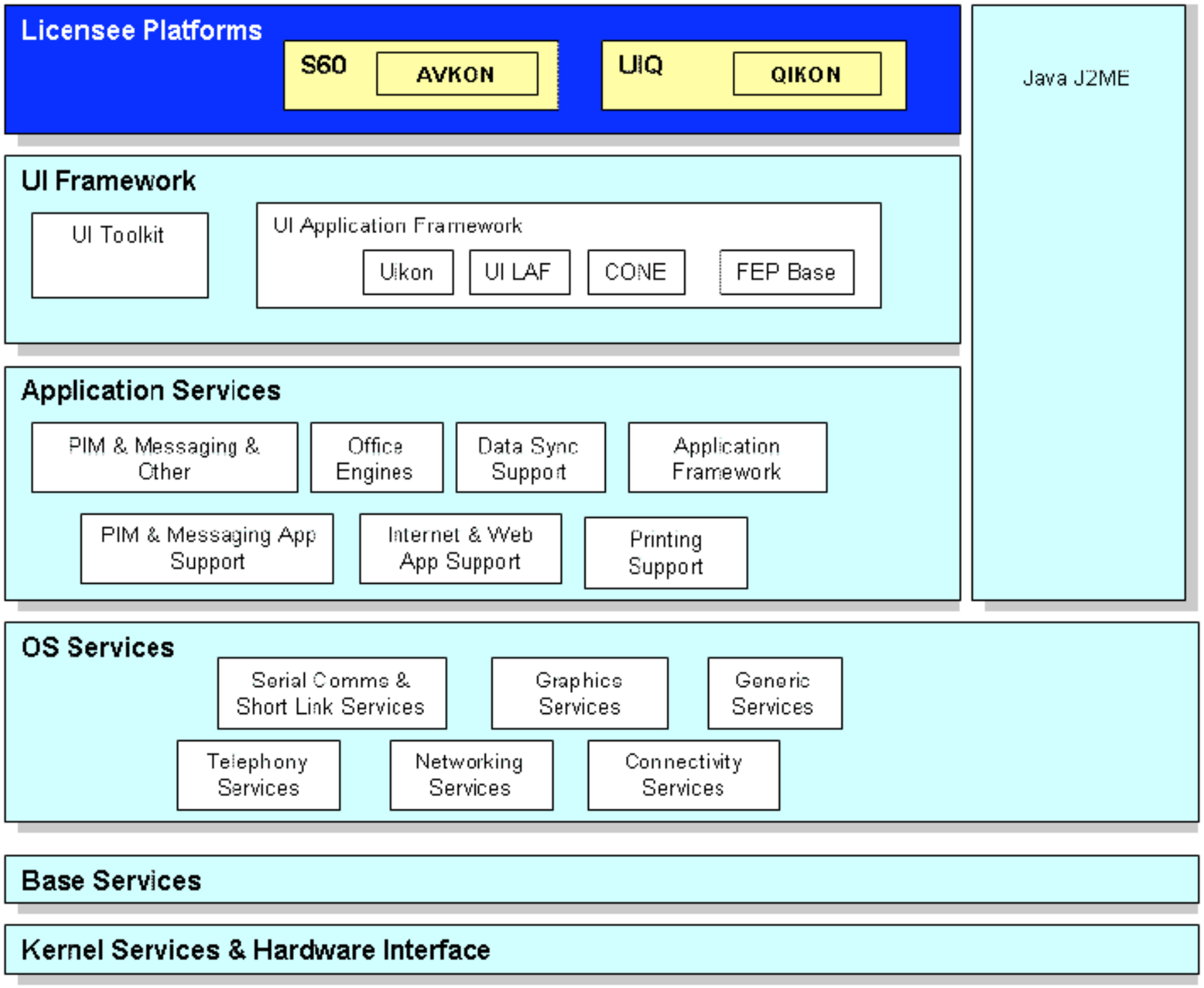
Symbian OS Architecture

Symbian OS separates the UI from the engines and services allowing licensees (like Nokia) to develop their own UIs for the phones



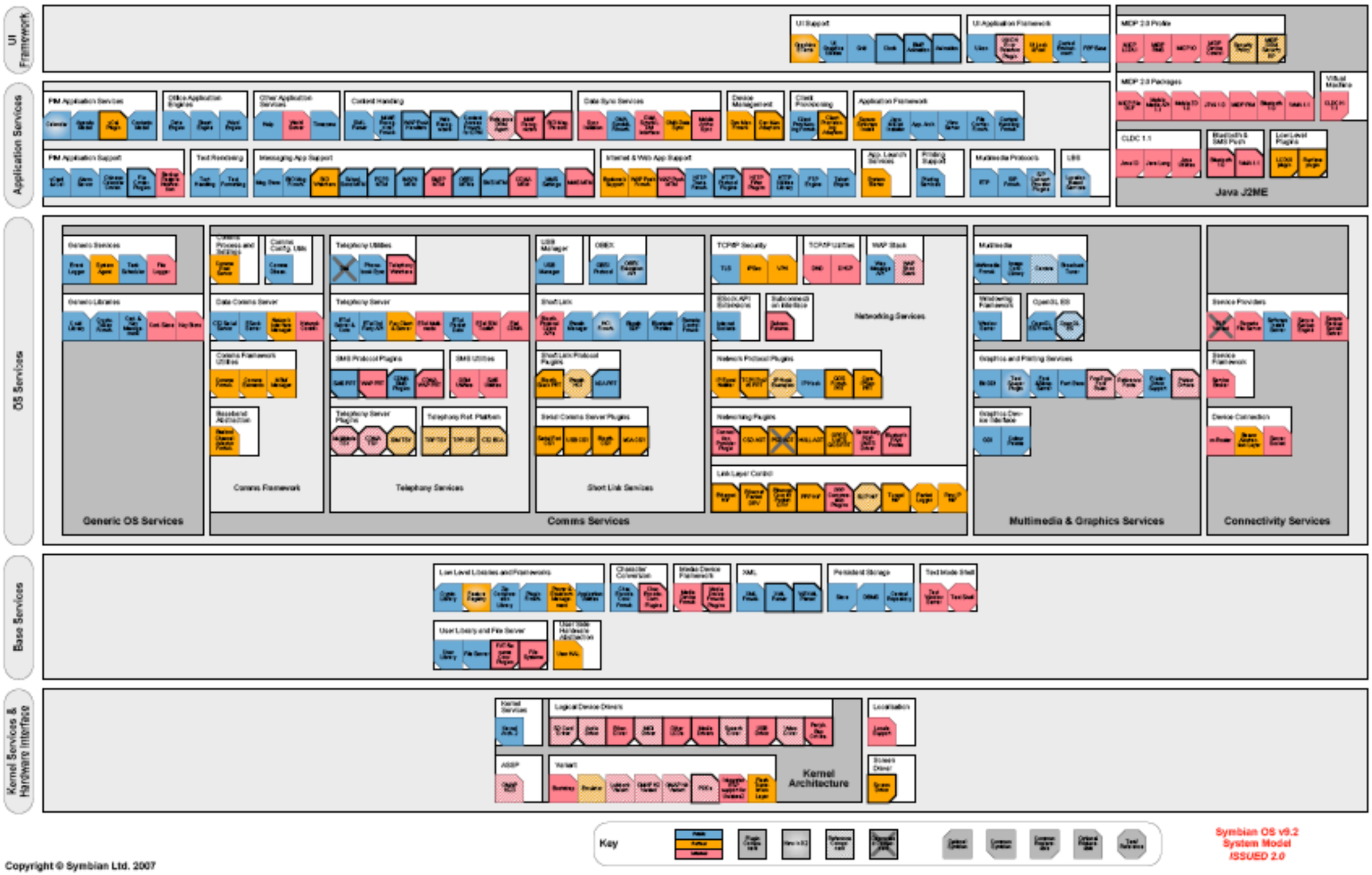
Symbian OS Architecture

In a little more detail



Symbian OS Architecture

Okay a little too much detail!



symbian Academy

Symbian OS Architecture

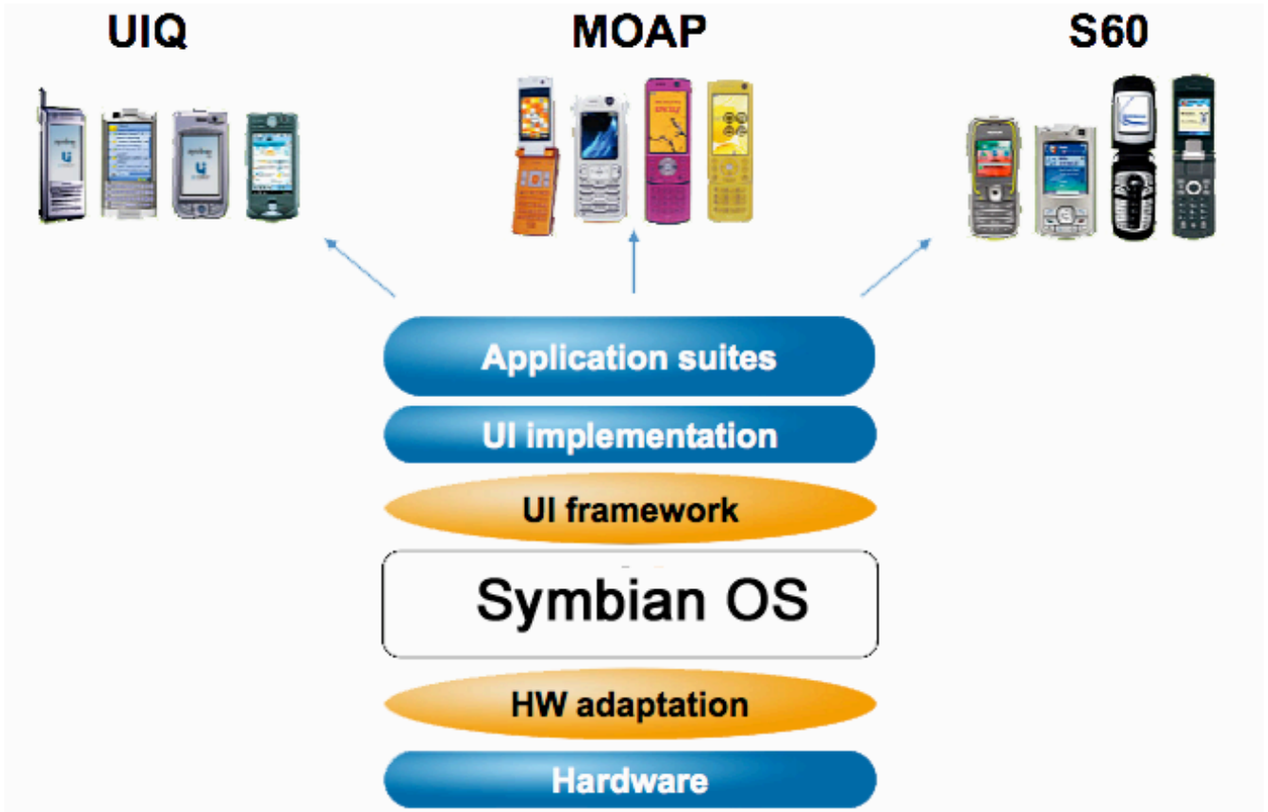
The Symbian System Model shown on the previous slide is available here

- http://developer.symbian.com/main/oslibrary/sys_models/index.jsp

Or refer to "The Symbian OS Architecture Sourcebook"

- http://developer.symbian.com/main/learning/press/

UI Platform



UI Platform

Various UI platforms run on top of Symbian OS:

- S60 from Nokia
- UIQ from UIQ Technology (part of Sony Ericsson)
- NTT DoCoMo's MOAP user interface for the FOMA™ 3G network in Japan

The platforms provide

- The GUI and extend the applications and middleware provided by Symbian

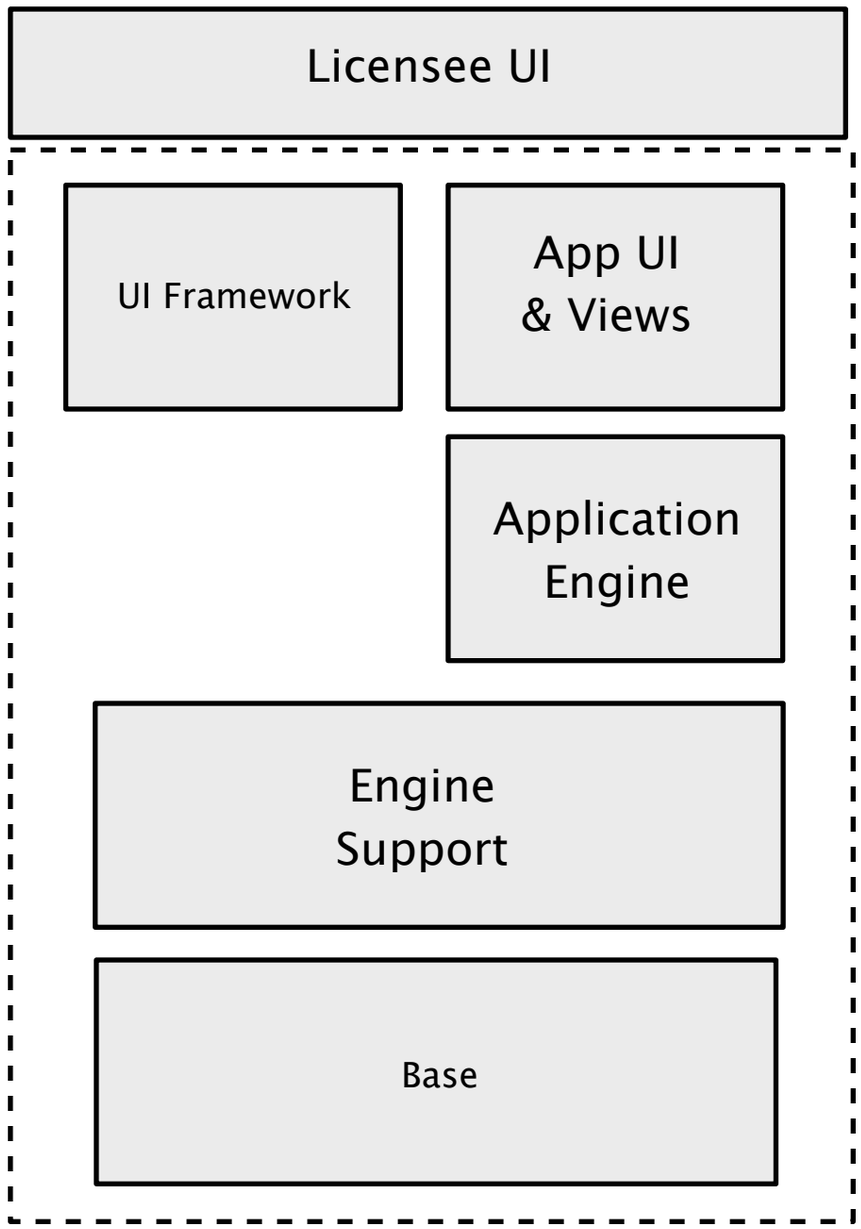
Each platform has a different look and feel

- Designed to support the characteristics of the handsets which use it

UI and Engine Separation

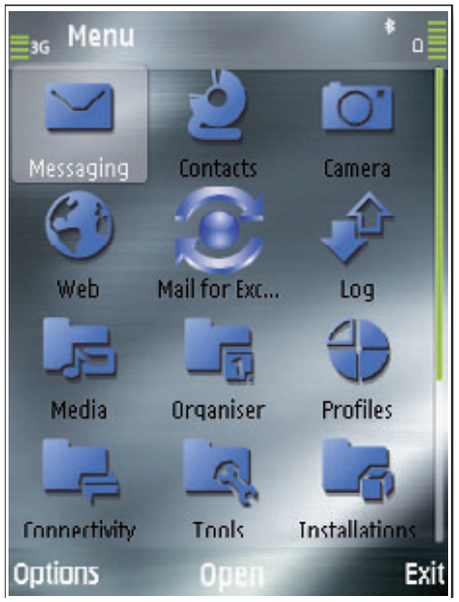
Broadly speaking the Licensee builds on the UI Framework

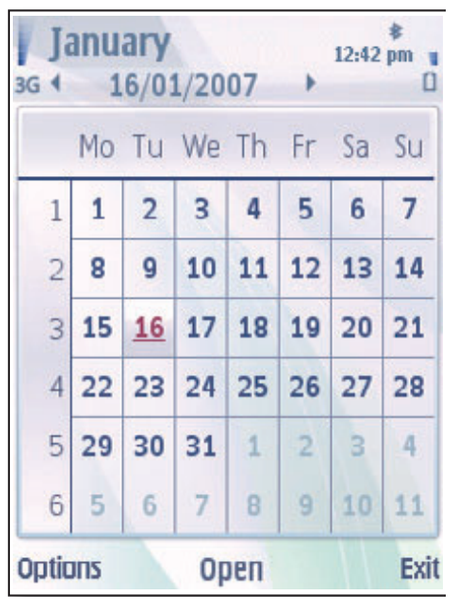
- Provided by Symbian OS
- We will touch on some of UI programming in the examples
- But the focus of this course is on core Symbian idioms and concepts



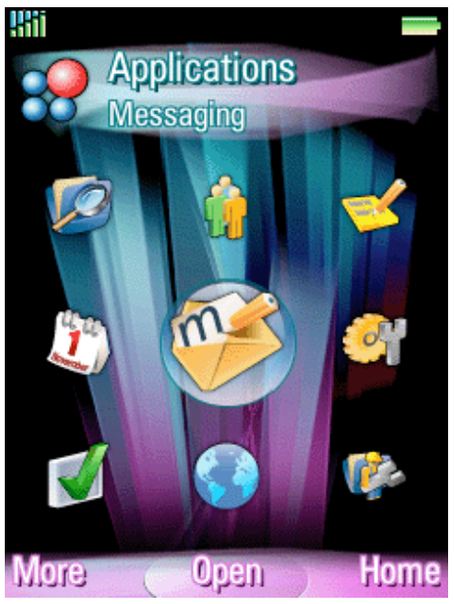
Examples of UI Platforms' Look and Feel

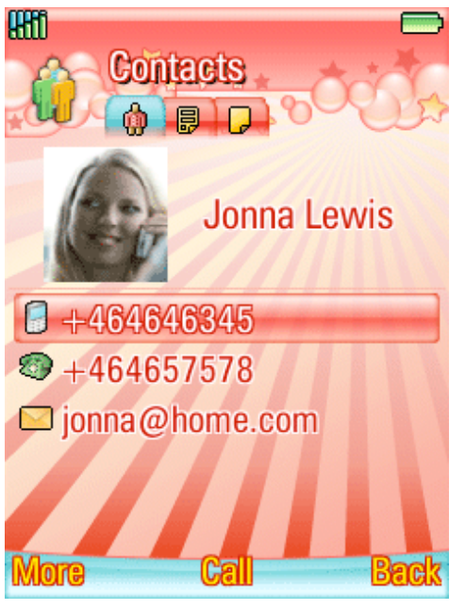
Nokia S60





Sony Ericsson UIQ





symbian Academy

Curriculum

Prerequisites

Prerequisites for this course are:

- C++ programming
- Object Oriented Design basics

Recommend at least some background knowledge in:

- Operating Systems theory
- Basic communication protocols

Curriculum

This course is based on the Symbian Academy Fundamentals of Symbian OS curriculum

- The professional qualification for Symbian OS developers is the based on this curriculum
- The Accredited Symbian Developer
- All Symbian in-house staff are expected to have this qualification
- The recommended book for this course is the "The Accredited Symbian Developer Primer: Fundamentals of Symbian OS" published by Symbian Press, 2006.

Training Exercises

Available training types:

Test your knowledge

Answer different types of questions, e.g. find errors in code or explain concepts.

Extend pre-written source code

A small framework has been marked with several "edit"-positions, which explain the individual tasks.

More freedom, more explanations

Instead of many small pre-defined tasks, tutorials contain more detailed explanations that help you finish the module.

The real challenge

At the end of the course, you should do a Symbian OS project, which will help you to repeat many of the basics as well as dive into more advanced topics.

Curriculum Core Subjects

Tool Chain

Symbian OS Types & Declarations

Leaves and the Cleanup Stack

Two-Phase Construction and Object Destruction

Descriptors

Dynamic Arrays

Active Objects

System Structure

Client Server

File Server, Store & Streams

Sockets

Platform Security

Binary Compatibility

Tool Chain

Introduces the Symbian OS tool chain and development environment

- Build Tools
- Hardware Builds
- Installing an Application to Phone Hardware
- The Symbian OS Emulator

Symbian OS Types & Declarations

Examines the fundamental Symbian OS types, naming conventions and coding style, and the usage paradigms of each basic type of Symbian OS class

- T Classes
- C Classes
- R Classes
- M Classes
- Static Classes
- Factors to Consider when Creating a Symbian OS Class
- Why Is the Symbian OS Naming Convention Important?

Symbian OS Types & Declarations

Exercise

The goal of the first exercise is to get to know the basic data types of Symbian OS as well as the console mode of the Symbian OS emulator.


```
==================Console - Types==
Welcome to example 1!
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5
Guess a random num [0..9]
Your guess: 1
Guess again...
Your guess: 7
Guess again...
Your guess: 5
Guess again...
Your guess: 3
Correct!
DYNAMIC TEXT

[press space to exit]
_
```

Leaves and the Cleanup Stack

Examines the Cleanup Stack and the difference between standard C++ and Symbian OS in handling leaks and exceptions

- Leaves: Lightweight Exceptions for Symbian OS
- How to Work with Leaves
- Comparing Leaves and Panics
- What Is a TRAP?
- The Cleanup Stack
- Detecting Memory Leaks

Leaves and the Cleanup Stack

Exercise

This module provides a hands-on experience with leaves and various ways on how the Cleanup Stack can help with handling risky memory situations.

Repetition Form

Exercise

```
==================Console - Leaves========
Inside CLeaveClass::CLeaveClass()
Inside CLeaveClass::PrintText()
Inside CLeaveClass::~CLeaveClass()
--------------------
Inside RCleanupTest::RCleanupTest()
Inside RCleanupTest::Close()
MainL() failed, leave code = -2

[Press space to exit]

No memory leaks detected! [Press space]
```

Two-Phase Construction and Object Destruction

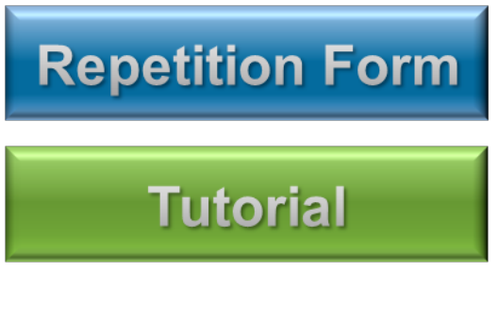
Examines 2-phase construction in Symbian OS, safely creating objects and avoiding memory leaks

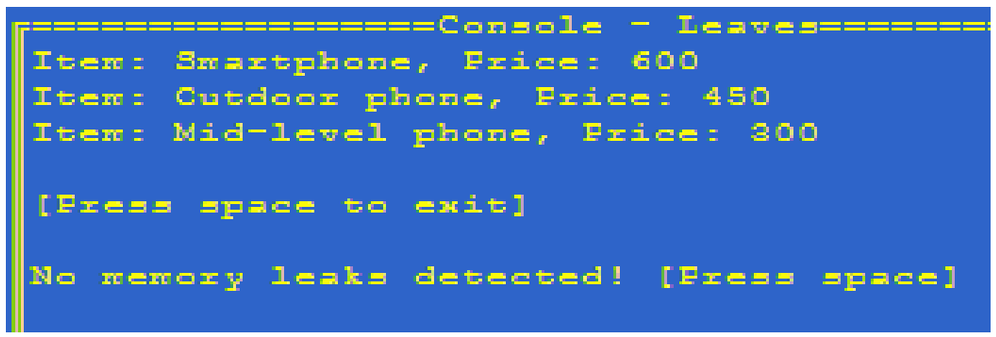
- Two-Phase Construction
- Object Destruction

Two-Phase Construction and Object Destruction

Exercise

In this module, you will see how two-phase construction works in Symbian OS and what can happen if you do not implement it correctly.





Descriptors

Examines the motivation for using descriptors, how to use buffer and pointer descriptors and when to use package descriptor classes

- Features of Symbian OS Descriptors
- The Symbian OS Descriptor Classes
- The Inheritance Hierarchy of the Descriptor Classes
- Using the Descriptor APIs
- Descriptors as Function Parameters
- Correct Use of the Dynamic Descriptor Classes
- Common Inefficiencies in Descriptor Usage
- Literal Descriptors
- Descriptor Conversion

Descriptors

Exercise

The two parts of this module cover working with descriptors. Through various examples, you learn their individual differences as well as a few of the available manipulation functions.

Repetition Form

Exercise

Exercise

```
==============Console - Descriptors 1=====
Compare() using str1 and str2 = -12
Compare() using str1 and KString1 = 0
Find KFind1 in str1 = 3
Match str1 and KMatch1 = 2
Append str1 to str2 = MY TEXTMy Text
Delete chars from str2 = MY Text

[Press space to exit]

No memory leaks detected! [Press space]
```

```
==============Console - Descriptors 2======
Current length: 6
Hello World!
Current length: 12
Current length: 6
I'm a  : TBuf
I'm an: RBuf
I'm an: HBuf
I'm no: TBuf
I'm no
String -> Number: 123
456
I'm only ASCII!

[Press space to exit]

No memory leaks detected! [Press space]
```

Dynamic Arrays

Examines the use of Symbian OS dynamic arrays in preference to standard C++ arrays, and the choice of dynamic array class depending on desired usage and characteristics of array elements

- Dynamic Arrays in Symbian OS
- **`RArray`**, **`RPointerArray`** or **`CArrayX`**?
- Array Granularities
- Array Sorting and Searching
- **`TFixedArray`**

Dynamic Arrays

Exercise

In this module, you will write a class that represents a student with a name and several other details. Dynamic arrays will be used to store several students as well as to sort or find elements.

Repetition Form

Exercise

Tutorial

```
==============Console - Dynamic Arrays=======
Original array
#0: Andrew (24), 474
#1: Judi (28), 742
#2: Hugh (22), 239

Sorted array
#0: Hugh (22), 239
#1: Andrew (24), 474
#2: Judi (28), 742

Ordered Insert
#0: Hugh (22), 239
#1: Andrew (24), 474
#2: Mopius (23), 568
#3: Judi (28), 742

Finding an element
Student 742: Judi

Sort by name
#0: Andrew (24), 474
#1: Hugh (22), 239
#2: Judi (28), 742
#3: Mopius (23), 568

Search student using name and number:
Found student: Andrew

[Press space to exit]
```

Active Objects

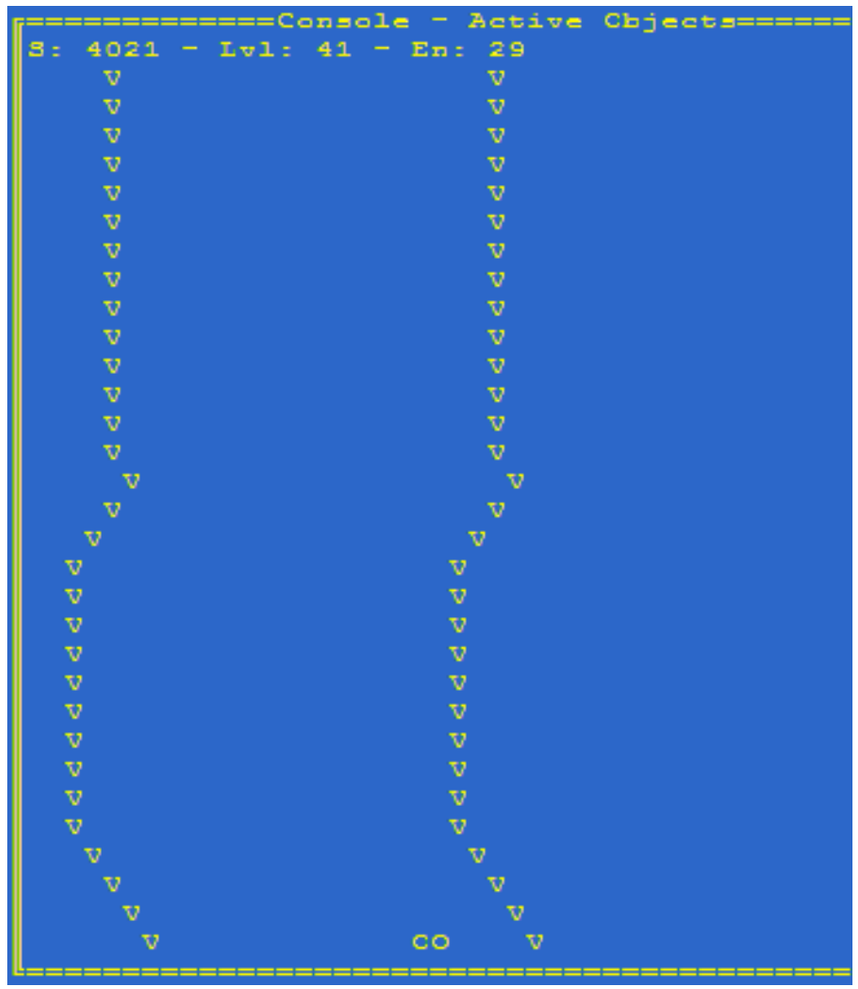
Examines why Active Objects are preferred over Threads and under what conditions, and the use and implementation of simple Active Objects and their relation to the Active Scheduler

- Event-Driven Multitasking on Symbian OS
- Class `CActive`
- The Active Scheduler
- Canceling an Outstanding Request
- Background Tasks
- Common Problems

Active Objects

Exercise

To make the subject of active objects more interesting, this exercise is based on a small console-based action game. You will have to write two active objects that provide the game engine with input.



System Structure

Examines the underlying structure of the Symbian OS, including threads, processes, DLLs and memory management

- DLLs in Symbian OS
- Writable Static Data
- Executables in ROM and RAM
- Threads and Processes
- Inter-Process Communication (IPC)
- Recognizers
- Panics and Assertions

Client–Server Framework

Examines the use cases of the Client / Server model within a handset, system components utilizing the model and a classic Server implementation

- The Client–Server Pattern
- Fundamentals of the Symbian OS Client–Server Framework
- Symbian OS Client–Server Classes
- Client–Server Data Transfer
- Impact of the Client–Server Framework

File Server and Streams

Examines the use of files, stores and streams for storing persistent and temporary data

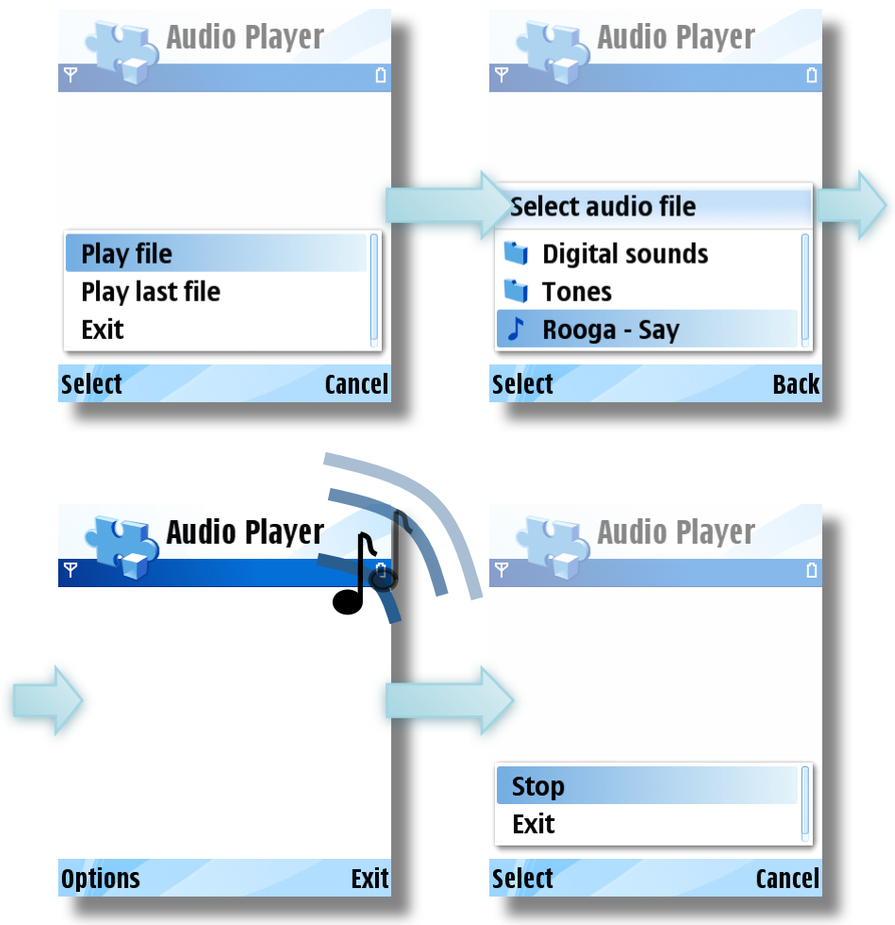
- The Symbian OS File System
- Streams and Stores

Client–Server Framework
File Server and Streams

Exercise

You will create a fully working audio player application for mp3/aac/...-files based on the S60 platform. To make this work, client server communication is demonstrated through the use of the multimedia framework as well as the file server.

or



Sockets

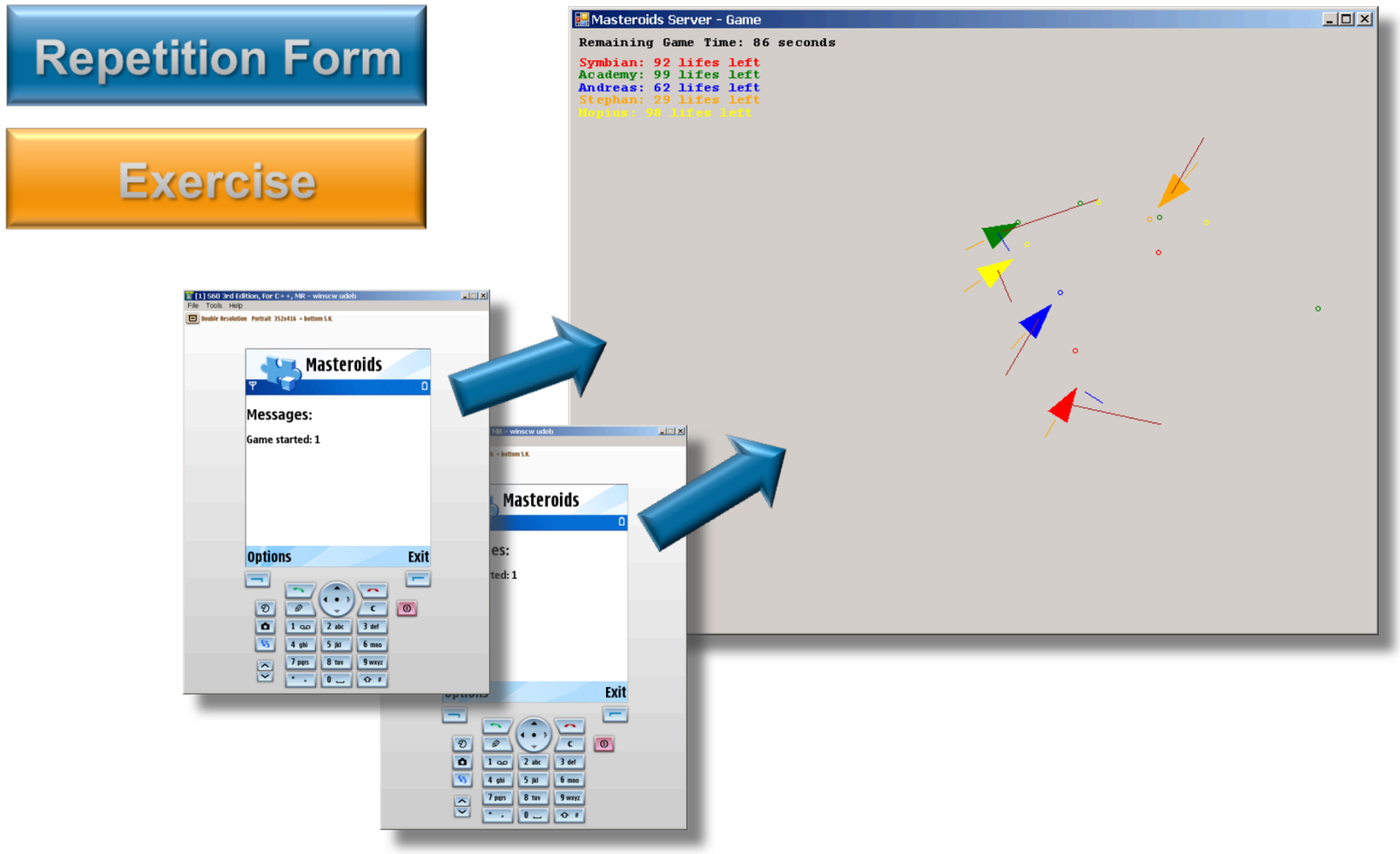
Examines the communication services and asynchronous communication events with the socket server architecture

- Introducing Sockets
- The Symbian OS Sockets Architecture
- Using Symbian OS Sockets

Sockets

Exercise

Instead of some artificial data transfer over sockets, this example requires you to complete the sockets-part of a fully functional multiplayer game client. The server is a desktop application and allows many (Symbian OS) clients to connect and play a game of good old Asteroids together – or rather against each other.



symbian Academy

Platform Security

Examines the three core concepts of Platform Security: The Trust Model, Capabilities and Data Caging. Introduces the designing, developing and distributing software on Symbian OS v9

- The Trust Model
- Capability Model
- Data Caging
- Secure Identifier, Vendor Identifier and Unique Identifier
- Application Design for a Secure Platform
- Releasing a Secure Application on Symbian OS v9
- The Native Software Installer

Compatibility

Examines binary and source compatibility along with the programming and design techniques which are required to maintain compatibility in code modules and APIs

- Levels of Compatibility
- Preventing Compatibility Breaks — What Cannot Be Changed?
- What Can Be Changed Without Breaking Compatibility?
- Best Practice — Designing to Ensure Future Compatibility

Project

Several project ideas in various difficulty levels are provided with the course materials. For most projects, the recommended team size is two. To give you enough time, the topics should be distributed to the groups as soon as possible.

To finish the projects, you will have to show that you understood the basic principles of Symbian OS. Usually, it is also required to do some research on one of the many APIs offered by Symbian OS, S60 or UIQ.

Resources

Recommended Reading

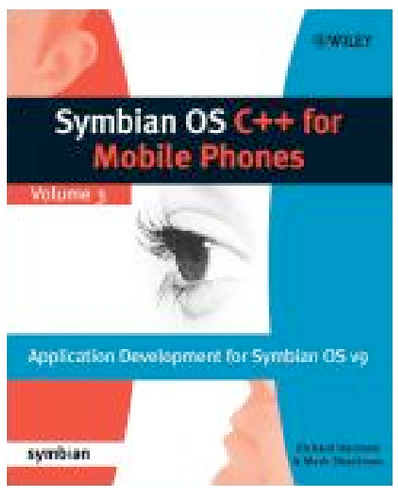
C++ reference books

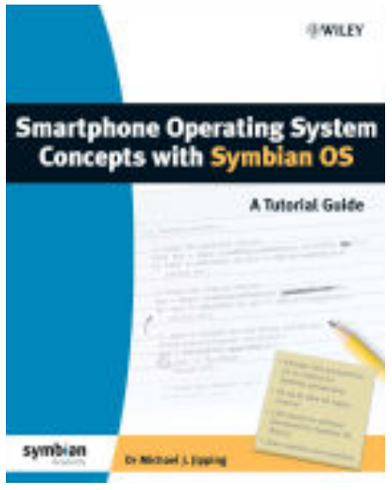
- Stroustrup, B. (2000) The C++ Programming Language (Special 3rd Edition), Addison-Wesley Professional
- Meyers, S. (2005) Effective C++: 55 specific ways to improve your programs and designs, 3rd Edition, Addison Wesley

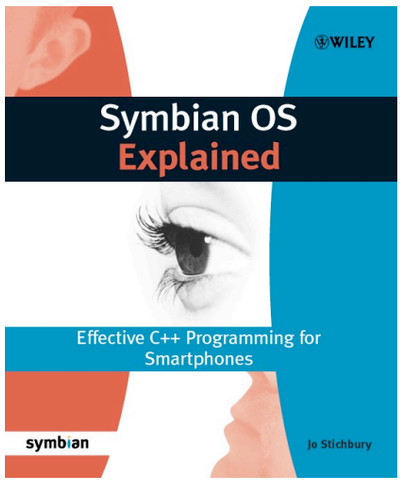
Symbian books

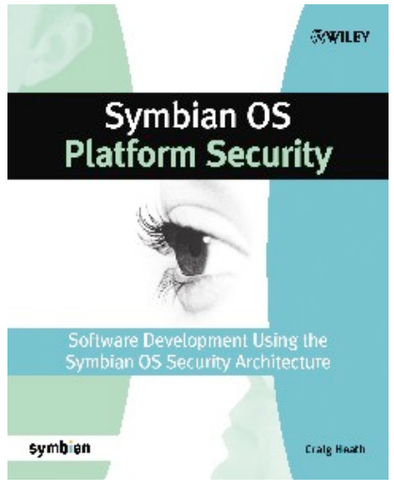
- Stichbury, J & Jacobs, M. (2006) The Accredited Symbian Developer Primer, John Wiley & Sons
- Jipping, M. (2007) Smartphone Operating System Concepts with Symbian OS, John Wiley & Sons
- Harrison, R. & Shackman M. (2007) Symbian OS C++ for Mobile Phones Volume 3, John Wiley & Sons
- Heath, C. (2006) Symbian OS Platform Security: Software development using the Symbian OS security architecture, John Wiley & Sons
- Morris, B. (2006) The Symbian OS Architecture Sourcebook: Design and evolution of a mobile phone OS, John Wiley & Sons

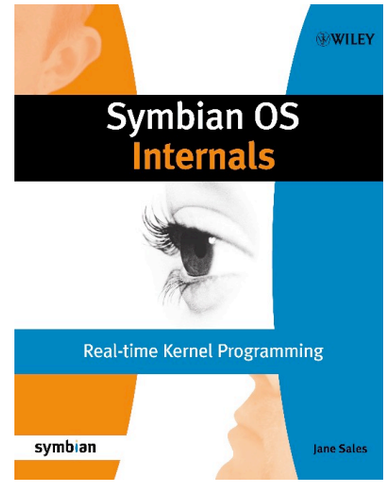
Symbian Press Titles

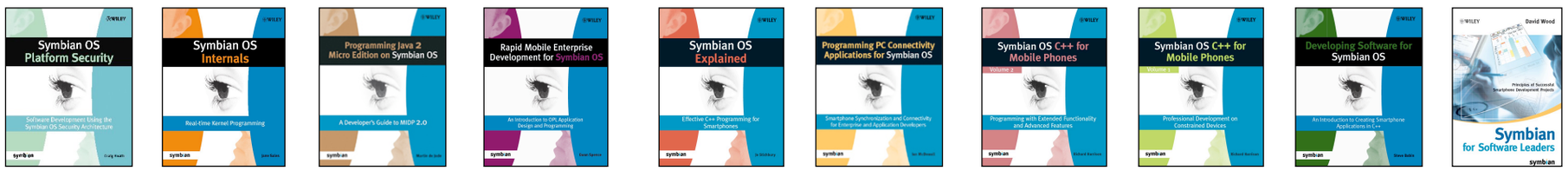












For information on other Symbian Press titles visit developer.symbian.com/main/learning/press

Regional Books

Symbian Press books translated into Chinese (PT Press)

- http://www.ptpress.com.cn/

Symbian Press books in India

- http://www.wileyindia.com/

Symbian Press books in Japanese

- http://developer.symbian.com/main/learning/press/translated/japanese.jsp

Useful Links

Symbian Developer Network

- http://developer.symbian.com

Symbian Academy

- http://developer.symbian.com/main/academy/

Developer Tools and SDKs

- http://developer.symbian.com/main/tools/

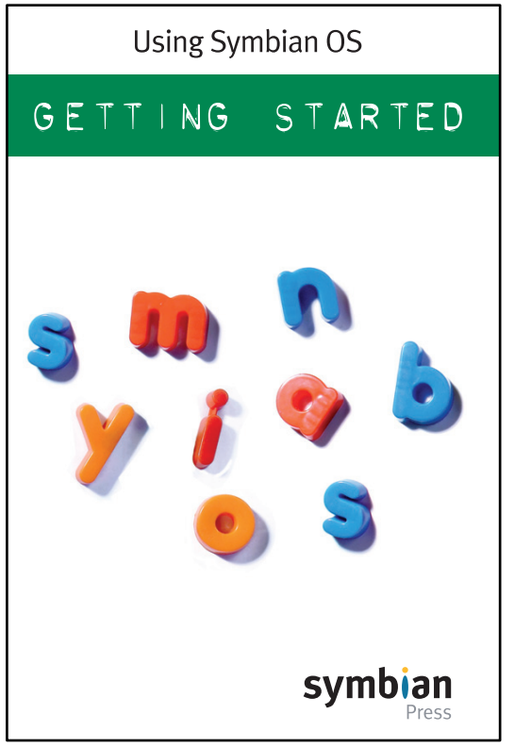
Forum Nokia for Universities

- http://forum.nokia.com/main/forum_nokia_for_universities/index.html

Essentials Booklets from Symbian Press

Free booklets (available for download as pdf)

- http://developer.symbian.com/main/learning/press/essential/booklets_using.jsp
- Also available in translation http://developer.symbian.com/main/learning/press/essential/booklets_translated.jsp



Good Luck and Have Fun!