Comparison of C++ and C#

Jim Fawcett

CSE687 – Object Oriented Design

Summer 2017

Both are Important

- C++ has a huge installed base.
  - Your next employer is very likely to be a C++ house.
- C# is popular for Website development and desktop Graphical User Interface applications.
  - Lots of job openings for C# developers.
- CSE681 – Software Modeling and Analysis
  - Focuses almost exclusively on C# and .Net.
- CSE687 – Object Oriented Design:
  - Focuses almost exclusively on C++ and the Standard Library.

Comparison of Object Models

- ***C++ Object Model***
  - All objects share a rich memory model:
    - Static, stack, and heap
  - Rich object life-time model:
    - Static objects live for the duration of the program.
    - Objects on stack live within a scope defined by { and }.
    - Objects on heap live at the designer's discretion.
  - Semantics based on deep copy model.
    - That's the good news.
    - That's the bad news.
  - For compilation, a source file must include information about all the types it uses.
    - That's definitely bad news.
    - But it has a work-around, e.g., design to interface not implementation. Use object factories.

- ***.Net Object Model***
  - More Spartan memory model:
    - Value types are stack-based only.
    - Reference types (all user defined types and library types) live on the heap.
  - Non-deterministic life-time model:
    - All reference types are garbage collected.
    - That's the good news.
    - That's the bad news.
  - Semantics based on a shallow reference model.
  - For compilation, a source file is type checked with metadata provided by the types it uses.
    - That is great news.
    - It is this property that makes .Net components so simple.

Language Comparison

- Standard C++
  - Is an ANSI and ISO standard.
  - Has a standard library.
  - Universally available:
    - Windows, UNIX, MAC
  - Well known:
    - Large developer base.
    - Lots of books and articles.
  - Programming models supported:
    - Objects
    - Procedural
    - Generic
  - Separation of Interface from Implementation:
    - Syntactically excellent
      - Implementation is separate from class declaration.
    - Semantically poor
      - See object model comparison.

- .Net C#
  - Is an ECMA standard, becoming an ISO standard.
  - Has defined an ECMA library.
  - Mono project porting to UNIX
  - New, but gaining a lot of popularity
    - Developer base growing quickly.
    - Lots of books and articles.
  - Programming models supported:
    - objects.
  - Separation of Interface from Implementation:
    - Syntactically poor
      - Implementation forced in class declaration.
    - Semantically excellent
      - See object model comparison.

Library Comparison

- Standard C++
  - Portable across most platforms with good standards conformance
  - I/O support is stream-based
    - console, files, and, strings
  - Flexible container facility using Standard Template Library (STL)
    - Now has hash-table containers
  - No support for paths and directories
  - Strings, regular expressions
  - Support for threads since C++11
  - No support for inter-process and distributed processing
  - No support for XML
  - Platform agnostic

- .Net Framework Class Library
  - Windows only but porting efforts underway
  - I/O support is function-based
    - console and files
  - Fixed set of containers that are not very type safe.
    - Has hash-table containers
  - Strong support for paths and directories
  - Strings and regular expressions
  - Thread support
  - Rich set of inter-process and distributed processing constructs
  - Support for XML processing
  - Deep support for Windows but very dependent on windows services like COM

Comparison of Library Functionality

Functionality	.Net Framework Libraries	Standard C++ Library
Extendable I/O	Weak	Strong
strings	Strong	Strong
Composable Containers	Moderately good	Strong
Paths and Directories	Strong	No
Threads	Strong	Good
Sockets	Moderately good	No
XML	Strong	No
Forms	Strong	No
Reflection	Strong	No

Additions to C++ Library Functionality

Functionality	Support Provided in Code from Website	Support Provided by you in Projects
Extendable I/O	NA - part of std library	-
strings	NA – part of std library	-
Composable Containers	NA – part of std library	-
Paths and Directories	FileInfo class, Path, and Directory classes	File Managers
Threads	Thread Pool	-
Sockets	Basic Demos, Socket class	Socket channels
XML	Reader, Writer, XML DOM	-
GUIs	Excellent WPF Framework	-
Reflection	No	No

End of Comparison