301AA - Advanced Programming [AP-2017]

Lecturer: Andrea Corradini andrea@di.unipi.it
Assistant: Lillo Galletta galletta@di.unipi.it

Department of Computer Science, Pisa
Academic Year 2017/18

AP-2017-02: Motivations and Introduction
Software is Everywhere
# Software as Competitive Advantage

## Most Admired Companies Making IT a Competitive Advantage (Forbes)

- Accenture
- Amazon
- Apple
- Cleveland Clinic
- General Electric
- Goldman Sachs
- Google
- Hospital Corporation of America
- IBM
- Intermountain Healthcare
- JP Morgan Chase
- Kaiser Permanente
- Mayo Clinic
- Microsoft
- Nestle
- Procter & Gamble
- Progressive Insurance
- Schlumberger
- Target
- Toyota
- Wells Fargo
Programming in the 21 century

• Software as complex as ever
• Command line interface not enough
• Data comes from multiple sources: structured (DB) and unstructured
• Single computer not enough
• Software development is a group activity
• Deployment on Web or mobile devices
Complexity Prompts for Innovation

- Object-Oriented Programming allows ever larger applications to be built
- But limited support for reuse
- OS + libraries not enough
- Reusable components are needed
- Multi-tier applications development increases the choices on how to build applications
Key Ingredients for Complex Software

• **Advanced features** extending programming languages

• **Component models** to ensure reusability

• **Frameworks** to support efficient development of (component based) applications

• **Execution environments** providing runtime support for ever dynamic software systems
The Software Architect

• A new role is needed: **Software Architect**
• to create, define or choose an **application framework**
• to create the component design according to a **component model**
• to structure a complex application into pieces
• to understand the interactions and dependencies among components
• to select the **execution environment / platform** based on cost/performance criteria
• to organize and supervise the development process
What are Frameworks?

• **Software Framework**: A collection of *common code* providing *generic functionality* that can be *selectively overridden* or specialized by user code providing *specific functionality*

• **Application Framework**: A software framework used to implement the *standard structure* of an application for a *specific development environment*
Framework Features

- Frameworks, like *software libraries*, provide *reusable abstractions* of code wrapped in a well-defined API.
- But: **Inversion of control**
  - Unlike in libraries, the overall program's flow of control is not dictated by the caller, but by the framework.
- Helps solving recurring design problems.
- Drives solution
  - Provides a default behavior.
  - Dictates how to fill-in-the-blanks.
- Non-modifiable framework code
  - Extensibility: usually by selective overriding.
Object-oriented programming frameworks consist of a *set of abstract classes*

- An application can be built simply inheriting from pre-existing classes in the framework.
- Instantiation of a framework consists of composing and subclassing the existing classes.
Examples of Frameworks

• General software frameworks
  – .NET – Windows platform. Provides language interoperability
  – Android SDK – Supports development of apps in Java (but does not use a JVM!)
  – Spring – Cross-platform, for Java applications
  – Cocoa – Apple’s native OO API for macOS. Includes C standard library and the Objective-C runtime.
  – Eclipse – Cross-platform, easily extensible IDE with plugins
Examples of Frameworks

• Frameworks for Application with GUI
  – **Gnome** – Written in C; mainly for Linux
  – **Qt** - Cross-platform; written in C++
Examples of Frameworks

- Web Application Frameworks [based on Model-View-Controller design pattern]
  - **ASP.NET** by Microsoft for web sites, web applications and web services
  - **GWT** - Google Web Toolkit (GWT)
  - **Rails** - Written in Ruby - Provides default structures for databases, web services and web pages.
Examples of Frameworks

• Concurrency
  – **Hadoop Map/Reduce** - software framework for applications which process big amounts of data in-parallel on large clusters (thousands of nodes) in a fault-tolerant manner.
    • **Map**: Takes input data and converts it into a set of tuples (key/value pairs).
    • **Reduce**: Takes the output from Map and combines the data tuples into a smaller set of tuples.
Framework Design

• Intellectual Challenging Task
• Requires a deep understanding of the problem domain
• Requires mastering of software (design) patterns, OO methods and polymorphism in particular
Design Patterns

• *General conceptual solutions to recurrent design problems*

• *More abstract than frameworks*
  – Frameworks can be embodied in code, but only *examples* of patterns can be embodied in code
  – Design patterns explain the intent, trade-offs, and consequences of a design

• *Smaller architectural elements than frameworks*
  – A typical framework contains several design patterns but the reverse is never true.

• *Less specialized than frameworks*
  – Frameworks always have a particular application domain
  – Design patterns can be used in nearly any kind of application
The 23 Design Patterns of the Gang of Four
Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides

*Design Patterns: Elements of Reusable Object-Oriented Software [1995]*
Course objectives and Syllabus
Course Objectives

• Understand programming language technology:
  – Execution Models
  – Run-time
• Analyze programming metaphors:
  – Objects
  – Components
  – Patterns
• Learn advanced programming techniques
• Present state-of-the-art frameworks incorporating these techniques
• Practice with all these concepts through small projects
Run-time Systems

• Virtual Execution Environment
  – Memory Management
  – Thread Management
  – Exception Handling
  – Security
  – Debugging Support
  – AOT and JIT Compilation
  – Dynamic Link/Load
  – Reflection
  – Verification

• A concrete example: the JVM
Selected Advanced Concepts in Programming Language

• Overloading and Type Classes in Haskell
• Lambda expressions and Streams in Java 8
• Closures vs Delegates in CLI
• Algebraic data types and Active patterns in F#
• Associative arrays in scripting languages
• Extensions in Swift
Advanced Programming Techniques

• Generic Programming
  – Java Generics
  – C++ templates
  – C# Generics
  – Scala generics
• Generative Programming
  – Metaprogramming
  – Reflection
  – Template
  – Generators
• Actor based programming
  – Scala and Akka
Component Models and Frameworks

- Component-oriented Programming
- JavaBeans and NetBeans
- Spring and Spring Beans
- COM
- CLR and .NET
- OSGi and Eclipse
- Hadoop Map/Reduce
### IEEE Spectrum Ranking 2017-2014

<table>
<thead>
<tr>
<th>Language Rank</th>
<th>Language</th>
<th>Types</th>
<th>Spectrum Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Python</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>100.0</td>
</tr>
<tr>
<td>2.</td>
<td>C</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /> <img src="server.png" alt="Server" /></td>
<td>100.0</td>
</tr>
<tr>
<td>3.</td>
<td>Java</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>99.4</td>
</tr>
<tr>
<td>4.</td>
<td>C++</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /> <img src="server.png" alt="Server" /></td>
<td>96.9</td>
</tr>
<tr>
<td>5.</td>
<td>C#</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>88.6</td>
</tr>
<tr>
<td>6.</td>
<td>R</td>
<td><img src="computer.png" alt="Computer" /></td>
<td>88.1</td>
</tr>
<tr>
<td>7.</td>
<td>JavaScript</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>85.3</td>
</tr>
<tr>
<td>8.</td>
<td>PHP</td>
<td><img src="computer.png" alt="Computer" /></td>
<td>81.1</td>
</tr>
<tr>
<td>9.</td>
<td>Go</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>75.7</td>
</tr>
<tr>
<td>10.</td>
<td>Swift</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>74.3</td>
</tr>
<tr>
<td>11.</td>
<td>Arduino</td>
<td><img src="phone.png" alt="Phone" /> <img src="server.png" alt="Server" /></td>
<td>72.4</td>
</tr>
<tr>
<td>12.</td>
<td>Ruby</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>72.0</td>
</tr>
<tr>
<td>13.</td>
<td>Assembly</td>
<td><img src="server.png" alt="Server" /> <img src="phone.png" alt="Phone" /></td>
<td>71.7</td>
</tr>
<tr>
<td>14.</td>
<td>Matlab</td>
<td><img src="computer.png" alt="Computer" /></td>
<td>68.6</td>
</tr>
<tr>
<td>15.</td>
<td>Scala</td>
<td><img src="computer.png" alt="Computer" /> <img src="phone.png" alt="Phone" /></td>
<td>68.0</td>
</tr>
</tbody>
</table>
Most Popular Coding Languages of 2016

- Python: 26.7%
- C++: 9.9%
- C#: 9.4%
- C: 7.37%
- JS: 6.9%
- Ruby: 5.9%
- Java: 22.6%
- PHP: 3.8%
- Go: 1.27%
- Scala: 1.04%
- Perl: 0.9%
- Obj-C: 0.8%
- TCL: 0.06%
- Clojure: 0.14%
- Lua: 0.19%
- R: 0.37%
- VB-NET: 0.37%
- Bash: 0.4%
- Haskell: 1.8%
<table>
<thead>
<tr>
<th>2015 Rank</th>
<th>2015</th>
<th>Change%</th>
<th>2014</th>
<th>Change%</th>
<th>2013</th>
<th>Change%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Python</td>
<td>26.67%</td>
<td>14.64%</td>
<td>31.24%</td>
<td>3.10%</td>
<td>5.21%</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
<td>22.58%</td>
<td>15.37%</td>
<td>19.57%</td>
<td>-11.85%</td>
<td>22.20%</td>
</tr>
<tr>
<td>3</td>
<td>C++</td>
<td>9.96%</td>
<td>1.76%</td>
<td>9.79%</td>
<td>-24.70%</td>
<td>13.00%</td>
</tr>
<tr>
<td>4</td>
<td>C#</td>
<td>9.39%</td>
<td>27.37%</td>
<td>7.37%</td>
<td>47.37%</td>
<td>5.00%</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>7.37%</td>
<td>21.37%</td>
<td>6.07%</td>
<td>48.14%</td>
<td>4.10%</td>
</tr>
<tr>
<td>6</td>
<td>JavaScript</td>
<td>6.88%</td>
<td>6.09%</td>
<td>6.48%</td>
<td>24.66%</td>
<td>5.20%</td>
</tr>
<tr>
<td>7</td>
<td>Ruby</td>
<td>5.88%</td>
<td>-17.27%</td>
<td>7.11%</td>
<td>-32.90%</td>
<td>10.60%</td>
</tr>
<tr>
<td>8</td>
<td>PHP</td>
<td>3.82%</td>
<td>5.45%</td>
<td>3.62%</td>
<td>9.84%</td>
<td>3.30%</td>
</tr>
<tr>
<td>9</td>
<td>Haskell</td>
<td>1.77%</td>
<td>17.24%</td>
<td>1.51%</td>
<td>25.83%</td>
<td>1.20%</td>
</tr>
<tr>
<td>10</td>
<td>Go</td>
<td>1.27%</td>
<td>-44.00%</td>
<td>2.26%</td>
<td>50.67%</td>
<td>1.50%</td>
</tr>
<tr>
<td>11</td>
<td>Scala</td>
<td>1.04%</td>
<td>-17.80%</td>
<td>1.27%</td>
<td>27.00%</td>
<td>1.00%</td>
</tr>
<tr>
<td>12</td>
<td>Perl</td>
<td>0.95%</td>
<td>-37.33%</td>
<td>1.52%</td>
<td>-6.17%</td>
<td>1.62%</td>
</tr>
<tr>
<td>13</td>
<td>Objective-C</td>
<td>0.82%</td>
<td>-17.62%</td>
<td>1.00%</td>
<td>265.76%</td>
<td>0.27%</td>
</tr>
<tr>
<td>14</td>
<td>Bash</td>
<td>0.46%</td>
<td>7.21%</td>
<td>0.43%</td>
<td>290.91%</td>
<td>0.11%</td>
</tr>
<tr>
<td>15</td>
<td>R</td>
<td>0.37%</td>
<td>165.71%</td>
<td>0.14%</td>
<td>-30.00%</td>
<td>0.20%</td>
</tr>
<tr>
<td>16</td>
<td>Visual Basic.NET</td>
<td>0.37%</td>
<td>825.50%</td>
<td>0.04%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Lua</td>
<td>0.19%</td>
<td>-44.51%</td>
<td>0.35%</td>
<td>337.50%</td>
<td>0.08%</td>
</tr>
<tr>
<td>18</td>
<td>Clojure</td>
<td>0.14%</td>
<td>-8.53%</td>
<td>0.15%</td>
<td>-48.28%</td>
<td>0.29%</td>
</tr>
<tr>
<td>19</td>
<td>Tcl</td>
<td>0.06%</td>
<td>-8.57%</td>
<td>0.07%</td>
<td>133.33%</td>
<td>0.03%</td>
</tr>
</tbody>
</table>
Top 10 Frameworks/Libraries

$1B+ VC-Backed Private Companies, by % of company usage

Git 70%
Jquery 63%
Hadoop 63%
Jenkins 43%
AJAX 37%
Prototype 37%
Backbone 37%
Selenium 27%
Spring 23%
Node.js 20%