Object-Oriented Programming: Smalltalk

- **Language Overview:**
  - It was based on the Flex Language and SIMULA67.
  - As an object-oriented language, all of its computing capacities are done through objects and methods.
  - Smalltalk is not just a language, it is also a complete software development environment.

- **Evaluation:**
  - It’s windowing system has become the dominant user interface system.
  - Although object-oriented programming ideas got popular from SIMULA67, they were more developed within Smalltalk.
  - It’s readability and writability are not to complex, and its Graphical User Interface (GUI) turned to be one of its most used features.
Combining Imperative and Object-Oriented Language: C++ Language

- C++ builds languages facilities, borrowed from Simula 67, on top of C to support, object-oriented programming, much of what Smalltalk pioneered.
- It evolved from C by improving its imperative features and adding constructs to support object-oriented programming.

Design Process

Moving from C to C++

- In 1980, Bjarne Stroustrup, from Bell Laboratories, made the first step in creating C++ from C.
- Function parameter type checking, conversion, and Classes were added.
- Derived classes, friend classes, public/private access controls, constructors and destructors.
- Later in 1981, inline functions, default parameters, and overloading of the assignment operator were added, and the resulting language was called C++

Design Process

Goals of C with Classes

- Primary goal – provide a language where programs could be organize with classes and inheritance.
- Second goal – little or no performance penalty relative to C.
- Third goal – that it could be use for every application for which C could be used, therefore virtually none of the features of C would be removed, not even those considered to be unsafe.

C++ is born

- 1984 – Inclusion of virtual methods, method name and operator overloading, and reference types. This version was called C++.
- 1985 - CFront , first implementation which translated C++ programs into C programs. (Release 1.0)
- 1985 – 1989 – continued to evolve based on user feedback.
- Release 2.0 – support for multiple inheritance and abstract classes.
- Release 3.0 (1990) – addition of templates and exception handling.
- Standardized in 1998 (ISO 98)

Design Process

.NET + C++ = MC++

- In 2002, .NET was released which included a new version of C++, named Managed C++ (MC++)
- Provides access to the functionality of the .NET Framework
- Inclusion of properties, delegates, interfaces, and reference types for garbage-collected objects. (multiple inheritance not supported)

Language Overview

- Constructs to define types: Classes and Structs.
- Multiple Inheritance is supported.
- Methods = member functions (Unlike Java)
- Supports both procedural and object-oriented programming.
- Operators and methods can be overloaded.
- Dynamic binding provided by virtual methods.
- Templates – methods and classes can be parameterized.
- Exception handling.

Evaluation

Pros

- Became and remains a very popular language due to the availability of good and inexpensive compilers.
- Almost completely backward compatible with C.
- Allows to link C++ code with C code, therefore easier to learn for many C programmers.

Cons

- Very large and complex language (PL/I)
- Inherited most of the insecurities of C.
An Imperative-Based Object Oriented Language: Java

Design Process

In 1990, Sun Microsystems decided that neither C or C++ would be satisfactory for developing software for consumer electronic devices. The design of Java was guided by the fundamental goal of providing greater simplicity and reliability than believed were provided by C++. Java designers started with C++, removed numerous constructs, changed some, and added a few others. The resulting language provides much of the power and flexibility of C++, but in smaller, simpler, safer and more reliable language.

Evaluation

- Java trims out excess and/or unsafe features of C++, making it more reliable
- Any language could be translated to an intermediate form and run on any platform that had a virtual machine for that intermediate form, yet the trade-off is cost in interpretation. The initial Java interpreter was at least ten times slower than any C++ compiler.
- To make Java competitive to C++ Java programs are now translated to machine code before being executed, using Just In Time (JIT) compilers.
- In the first years, Java was primary used for programming World Wide Web pages using applets.
- Java applets are small programs that run on the Web client when it finds an applet call in the HTML of a page displaying. When called, the intermediate code form of the applet is downloaded from the server to the client and the output is displayed in the browser. Java applet have recently decayed due to the appearance of client-side scripting languages, such as JavaScript.
- The use of Java increased faster than any other language. This is because Java offered a powerful, reliable language that was easier and safer than C++. It is now used for a wide variety of applications.

Language Overview

Differences Between C/C++ and Java

- Has both classes and primitive types, which are not objects based on classes, as does C++.
- Objects are accessed through reference variables, but the primitive types are value types.
- Java arrays are instances of a predefined class, whereas in C++ are not.
- C++ supports multiple inheritance directly in its class definitions. Java supports only single inheritance, though you can still gain multiple inheritance by the use of its interface construct.
- Java includes a relatively simple form of concurrency control through its synchronize modifier, which can appear in methods and blocks. In Java it is relatively easy to create threads.
- Addition of the Garbage Collector

Scripting Languages for the Web: JavaScript

- Joint venture between Netscape and Sun Microsystems
  - Originally called LiveScript
  - Follows ECMA-262 and ISO-16262 standards
  - Microsoft has a version called JScript
- Related to Java only by similar syntax
  - Dynamically typed (variable types are not determined at compile time)
  - Character strings and arrays have dynamic length
  - Does not support inheritance or dynamic method call binding
- Widely used as a web programming tool
  - Create dynamic HTML documents
  - Check the validity of form inputs
Slide 9

(C was small but did not support object oriented programming, which was necessary... and C++ had supported Object Oriented but was large and complicated since it also supported procedure-oriented programming)
ALEXIS, 2/1/2010

Slide 10

in C++ u can create wrapper classes to add features like index range checking in arrays and stuff, but this is implicit in Java
ALEXIS, 2/1/2010

concurrent programs
ALEXIS, 2/1/2010

Garbage collector which frees the programmer from being concerned with putting storage back to the heap when it is no longer needed, so it doesn't suffer from memory leakage.

memory leakage = the storage is allocated but never deallocated
ALEXIS, 2/1/2010
Scripting Languages for the Web: PHP

- Developed by Rasmus Lerdorf, Apache Group
- PHP: Hypertext Preprocessor
  - Originally called Personal Home Page
  - Basic object-oriented programming functionality
  - HTML-embedded server-side scripting designed to web applications
  - Allow simple access for HTML form
  - Excellent language for web access databases.
  - Support for database management systems.
- Related to Java and JavaScript
  - Dynamic nature of its strings and arrays, and its use of dynamic typing.
  - Similar to JavaScript in its syntactic appearance.
  - PHP’s arrays are a combination of JavaScript’s arrays and Perl’s associative arrays.

Scripting Languages for the Web: Python

- Developed by Guido van Rossum.
- Python characteristics
  - Supports multiple programming paradigms
  - Interpreted and interactive
  - Emphasizes code readability
  - High-level general-purpose PL that can be applied to many different classes of problems
  - Python code looks like executable pseudo code
  - Strong presence on the web

Scripting Languages for the Web: Ruby

- Motivation for Ruby development
  - Yukihiro Matsumoto, wasn’t comfortable with Python and Perl

- Ruby Characteristics
  - Pure object oriented language; but it supports other paradigms
  - Interpreted language
  - Dynamics and strong typing
  - Syntactically similar to Ada and Eiffel

A New Language for the New Millennium: C#

- Language based on C++ and Java.
- Also includes ideas from Delphi since its lead designer created Delphi.
- Part of the .NET Framework alongside with Visual Basic, Managed C++, J# and Jscript.
- All these languages are compiled into the same Intermediate Language (IL) and then compiled using a Just-In-Time compiler (JIT).
- Unlike Java it has support for pointers, structs, enum types, operator overloading and goto.
C# Evaluation

- Safer `enum` types due to implicit conversion to integers.
- Structs are changed to be a lightweight construct, able to implement interfaces.
- Switch statement changed by using `goto` statements at the end to be able to move to the next instead of using a break at the end.
- Contain Delegates which are object-oriented and type-safe method references.

C# More Features

- Variable number of parameters by using the `params` reserved word.
- Uses same typing system for primitives and objects.
- Also includes the `Rectangular Array (multidimensional)` array structure and a `for each` statement.