Introduction to Ruby

What is Ruby?
Ruby in CIS 170
“Hello, World”

Computer Programs

- From Tuesday’s lecture:
  - a computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.
  - a computation is a process that can be carried out by a person or a machine
  - If we want to carry out a computation on a machine, we need to write a program
  - a programming language is a notation for describing the steps of a computation
  - The earliest programming languages were developed in the 1950s
    - FORTRAN, COBOL, ALGOL, LISP, ....
  - Languages widely used today include Java, Perl, C++
  - The language we will use in CIS 170 is named Ruby

Ruby

- Ruby was defined in 1994 by Yukihiro Matsumoto
  - studied computer science and programming languages at Tsukuba University
  - had used Perl, Python, Java, and other languages, but none was “just right”
- His goals:
  - design a practical language
  - base it on important computer science concepts
  - implement these ideas in an “interactive” system (more on this later)
- The main concept: programs operate on sets of objects
  - numbers, strings, files, menus, windows, web pages, ...
  - objects in programs: Simula-67
  - object-oriented programming: Smalltalk (ca. 1975)
  - an idea used in Java, C++, and many other languages

Ruby

- Matsumoto’s goal was to design a language that had many of the same abilities as Perl
  - useful for writing scripts (interacting with other programs, processing text)
  - should also be effective for general purpose programming
  - one should be able to build large programs from smaller, independent pieces
- “Ruby” is a successor to “Perl”
  - pearl is the gemstone for June
  - ruby is the gemstone for July
Ruby Home Page

- Ruby is open source software anyone can download it and install it on their system.
- The official home page for the Ruby project is:
  http://www.ruby-lang.org/en/ (the “en” is for “English”)
- There are links for
  - downloads
  - documentation
  - mailing lists
    - mainly technical discussions, e.g. proposals for new features

Aside: open source is a movement. You can find the “open source manifesto” and learn more at:

GNU's Not Unix
www.gnu.org

Ruby in CIS 170

- We will be using Ruby for lab projects this term.
- Documentation on Ruby is available via the class web site.
- For most projects you will not need this extra information.
  - use it as an additional resource if you are curious or if you get stuck.

Where to Run Ruby

- For CIS 170 labs you have a choice:
  - run Ruby on your own computer
  - use one of the UO labs
- To install it on your own machine
  - Linux: download from www.ruby-lang.org, follow the installation instructions
  - Windows: download a package from http://rubyinstaller.rubyforge.org
  - Mac OS/X: Ruby is pre-installed on 10.4 and 10.5
- To run it at a UO computer center instructional lab
  - Klamath B13
  - McKenzie 101
  - see http://cc.uoregon.edu/campuslabs.html for locations and hours

Applications

- Normally when we think of programs we think of applications

Applications folder on my laptop
An application for composing and reading e-mail
Applications (cont’d)

- When we launch an application the system starts a program
  - inside the application are statements in a programming language
  - the statements tell the system how to create windows, where to place buttons and menus, etc

- It is possible to create a new application using Ruby
  - figure out what windows, menus, buttons, etc are needed
  - these “widgets” define the graphical user interface of the program
  - attach a set of statements in Ruby to each widget
  - when a user clicks on a widget the Ruby code is activated
  - But writing a complete application is very hard work
    - requires lots of experience, extensive knowledge of programming libraries

Before Graphical Interfaces

- In the early days of computing there were no graphical displays
  - no mice
  - no display terminals

- Programs were typed in on keypunch machines (which produced punched cards) or typewriters or teletype machines

Interactive Computing

- Most programs in this era used “batch processing”
  - an operator would enter a set of programs and data
  - a scheduler (part of the OS) would run programs one at a time
  - often a user would submit a program, return the next day to get results

- If a computer had a typewriter, it was possible to run an interactive program
  - the user would type a command
  - the processor ran the program
  - when the program was done the system would print the results
  - an effective way of running smaller programs
Terminals

- In the 1970s video terminals replaced typewriters and teletypes
  - electronic keyboard
  - video display instead of paper
  - encouraged the development of interactive applications

Terminals (cont’d)

- Today when we want to run a program interactively we use a terminal emulator
- An emulator is a very simple application
  - The application window represents the screen from an old-fashioned terminal
  - when the emulator is running, keystrokes you type are sent to the OS
  - you can start a program, and everything the program prints is displayed in the window

Terminal Emulator on OS/X

- Terminal emulators are part of every operating system
  - in OS/X, run the application named Terminal from the Applications/Utilities folder

Terminal Emulator on Windows

- Windows users can run a program named cmd (“command line processor”)
- Two other applications -- fxri and FreeRide -- are very useful
  - both are installed on the systems in the labs
Interactive Ruby

- In CIS 170 we will be using Ruby as an interactive programming language.
- An interactive language is basically a fancy calculator.
  1. Start Ruby.
  2. Ruby prints a prompt to let you know it is ready.
  3. Type an expression.
  4. Ruby evaluates the expression and prints the result.
  5. Go back to step 2.

- The CS jargon for this cycle is `read-eval-print loop`.

Interactive Ruby (irb) on Mac OS/X

Interactive Ruby Examples

- For the remainder of the term I will be showing Ruby examples without a picture of a terminal emulator.
  - I will assume you know how to start Interactive Ruby on your system.
  - The pieces of Ruby shown in these slides should work equally well on OS/X, Windows, or Linux systems.
  - In these slides, prompts and other strings printed by Ruby will be shown in red, and characters I type will be shown in black.
  - Example: the "hello world" program (Lab 0):

```
p "hello, world"
```

Final Comments on Interactive Ruby

- Interactive Ruby has lots of "bells and whistles."
  - You can also hit delete and use other keys to edit the expression.
  - Hint: learn how to use "command history."
  - Hit the up-arrow key, go back and edit the previous command -- saves typing.
  - Explore these features and others on your own.

Get started on "Project 0" today!

Make sure you can start a terminal application, start Ruby, and evaluate an expression.