IST488/688 Social Web Technologies, Fall 2011
Lab 1.
Instructor: Keisuke Inoue

Description:

In this lab, you will work on implementing concepts for a social web site using object-oriented programming. All the information you will need is covered in Cooper, Chapter 2. The following instructions will help you to accomplish the task, but will not tell you all the small steps. The lab is designed to encourage your learning and enhance your understanding. You may want to look up the lecture slides, textbook, or any other online resources. If you have a question, feel free to ask the instructor or the assistant.

Note: this lab assignment is designed as an individual work. To maximize your learning experience, you are encouraged to work on it independently. You may ask questions to instructors and/or friends (but keep it quiet) and look up resources, but you are not allowed to copy-and-paste any resources other than you created.

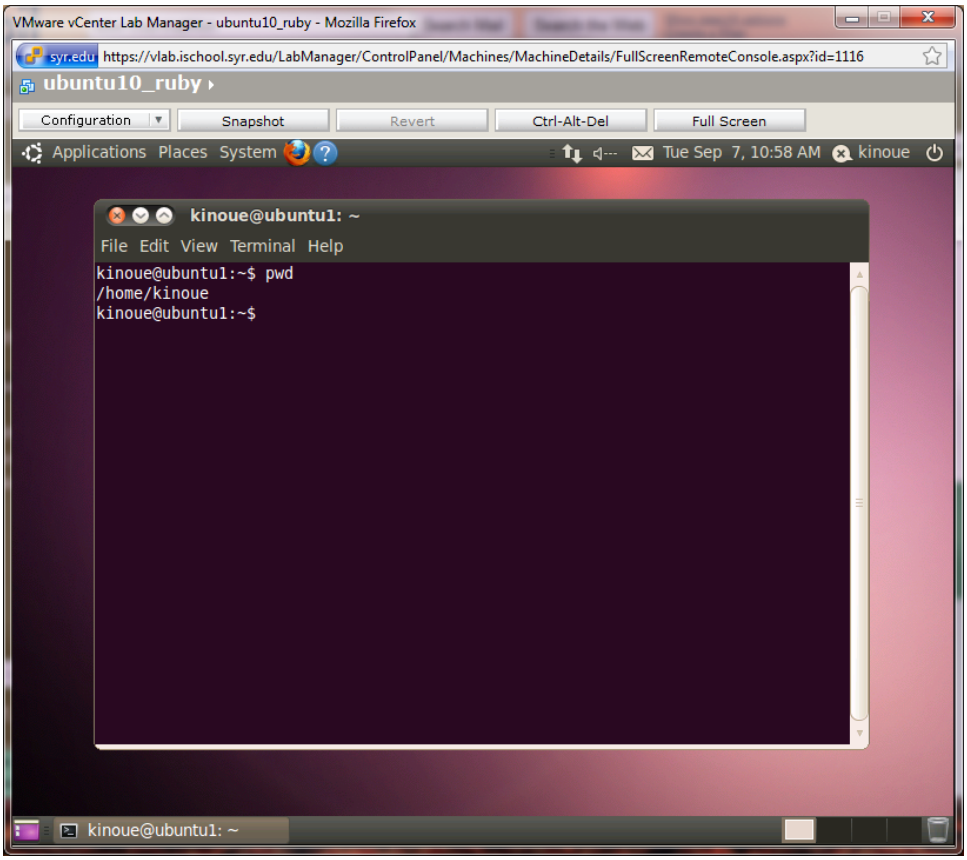
Scenario:

You are starting a new online dating site. (Maybe because you heard that online dating business has grown to a billion dollar market (http://www.mindbranch.com/listing/product/R237-214.html). Using your expertise in object-oriented programming, you are starting to design and implement concepts for your new online dating service. You can think about some specific target, too, for example: dating site for SU students, artists, musicians, librarians, sport fans, runners, etc. Use your imagination to tap on a new market!

Lab Instructions:

1. Setup and basic Linux operations

Before we jump in to the programming, let us prepare the environment and practice basic Linux operations.



1.1. Login to your Linux system, using VMware vSphere Client. (For the detailed instruction, see the other documentation “vSphere Login Instructions”.) Throughout this course, you will use the following user account on Linux:

User name: itell
Password: SU2orange!

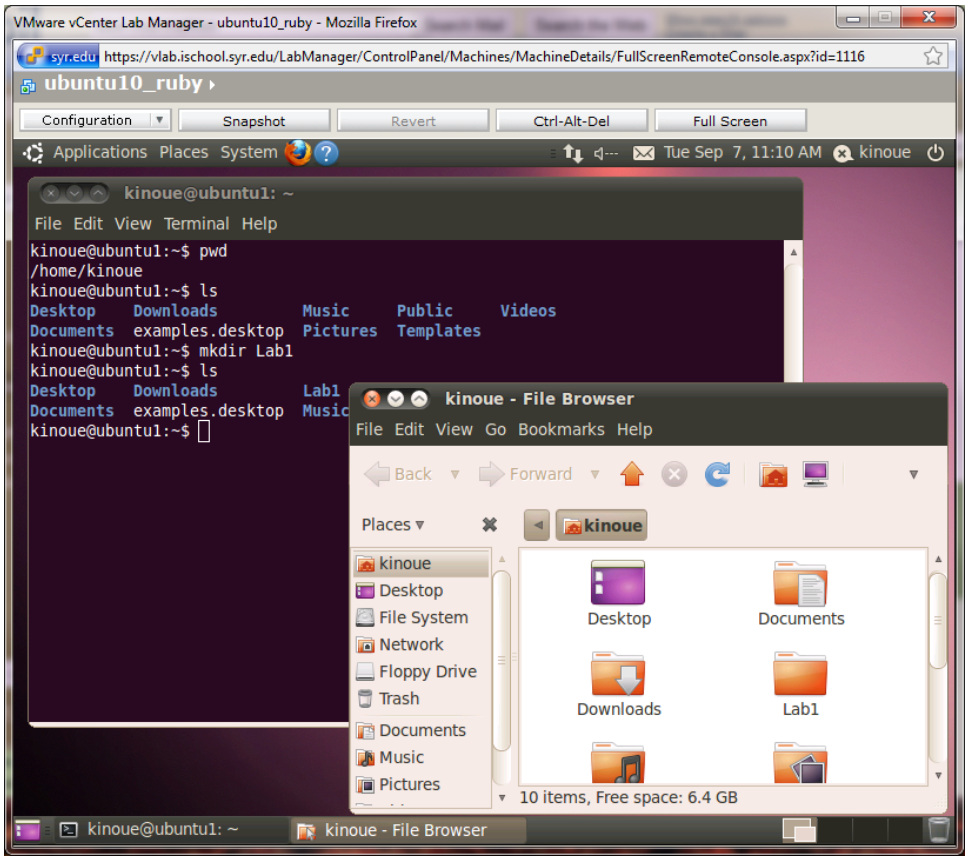
1.2. Once you are logged in, launch the Terminal application by selecting “Applications” → “IST488/688 Tools” → “Terminal”.

1.3. Now type in “pwd” and hit return. The terminal will tell you where you are in the file system. It should say “/home/[your user name]”. This is your home directory. (See the screenshot above.)

1.4. Type in “ls”, and you will see the list of files or directories in your home directory.

1.5. Now, create a new directory (or folder… it’s a same thing.), named “Lab1”, by typing “mkdir Lab1”.

1.6. Type in “ls” and you can see that the directory is created. You can also make sure the same thing by launching a File Browser by selecting “Places → Home Folder”.



1.7. Now on the terminal, move to the Lab1 directory, by typing “cd Lab1”. (You can also double-click the Lab1 icon on the File Browser window.). Type in “pwd”, and you should see “/home/[your account]/Lab1”.

1.8. In this class, we are not going use any special development environments, such as Visual Studio. The simplicity of Ruby and Rails allow you to use any text editor to develop a full-scale Web application.

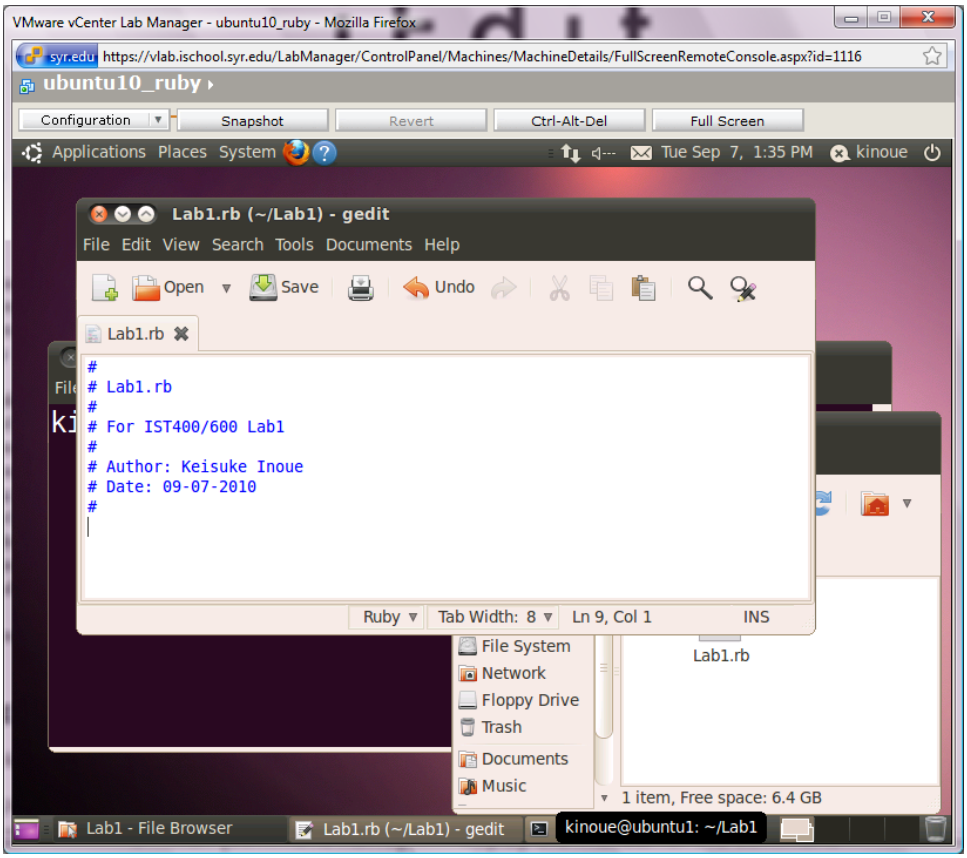
Launch your favorite text editor by selecting one of the following options:

- “Applications” → “IST488/688 Tools” → “Komodo Edit”

- “Applications” → “IST488/688 Tools” → “jEdit”
- “Applications” → “IST488/688 Tools” → “gEdit”
- “Applications” → “IST488/688 Tools” → “GNU Emacs 23”

If you have no preference, my recommendation is to use Komodo Edit.

1.9. On your editor, save a new file, named “Lab1.rb”, in the Lab1 directory. In the top part of the file, insert lines, starting “#”, followed by basic information: the file name, purpose, author, and date. The lines starting with “#” are called comments; they are ignored by the interpreter, but improve the readability of the code.



Phew! That’s it for the setup. Now let’s move on to the programming…

Just to review, here are some of the basic commands you need to remember while using Linux:

- ls: print out the files and directories in the current directory
- pwd: print out the current directory
- cd [directory name]: change the current directory to the specified one
- mkdir [directory name]: create the directory
- rmdir [directory name]: remove the directory
- man [command name]: display the manual for the command

Some useful short cuts:

- !! : repeat the previous command
- .. : the parent directory of the current directory, i.e., “cd ..” means moving up one directory.
- ~ : your home directory, i.e., “cd ~/Lab1” means moving to the directory you created.

2. Design

Now, let's conceptualize the users of your new online dating site. What should the attributes of the users be? Here are some basic ones: name, birth_year, birth_month, birth_date, current_city, and favorite_food. Think at least two more attributes on your own. The attributes should be either a string or a number. Let's keep them simple for now.

3. Defining the User Class

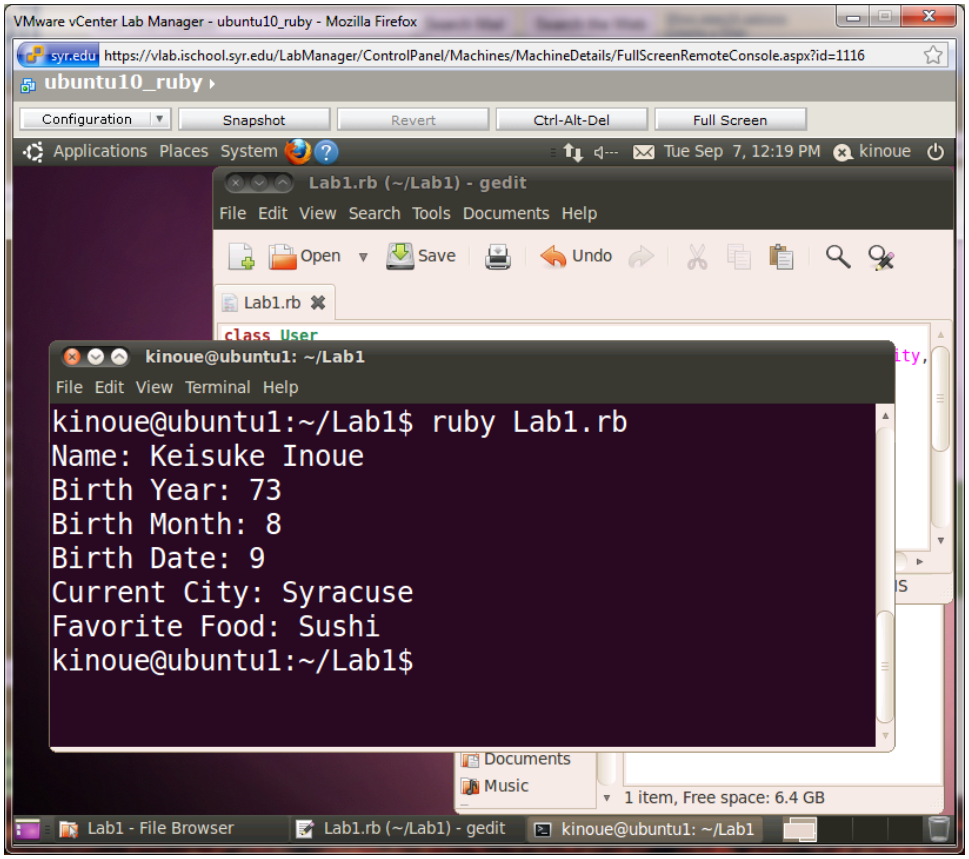
In your Lab1.rb, define a class, called User, which includes all the attributes that are listed above and the ones you came up with. Save the file, and run the script on your terminal, by typing "ruby Lab1.rb". Nothing should happen at this point. If you see error messages, your code has errors. Go back to the text editor and fix the problem before moving to the next step.

4. Creating a User object

Now that the class is defined, let's create an object. Below the class definition, create a variable, called u1, and assign a new instance of the User class. Set some values for all the attributes of u1. You can assign whatever the values. Once it's done, let's make sure that all the attributes are correctly set. Create statements to print out all the attributes, so that you will see something like below (of course, your output should vary.):

Name: Keisuke Inoue
Birth Year: 73
Birth Month: 8
Birth Date: 9
Current City: Syracuse
Favorite Food: Sushi

Once you inserted the statements, run the script on the terminal window, and make sure they are working.



5. Defining a method (1)

So far, the User class you defined does not have any original methods. Let's define a method, called intro, that introduces the user. The output of u1.intro should be something like:

> Hi, my name is Keisuke Inoue. I live in Syracuse, and I like sushi.

Not a very attractive self-introduction, but you got the idea. You can try creating better sentences, using your own attributes. Once the method is implemented, insert a statement that calls the method of u1 at the bottom of Lab1.rb. Run the script and make sure it's working.

6. Defining a method (2)

Go back to the User class definition. Define another method, called age, which tells you how old the user is in 2010. We will be able to define a more precise method later in the class, but for now, let's just estimate the User's age by subtracting the year of birth from 2011.

Once the method is implemented, insert a statement that calls the method of u1 at the bottom of Lab1.rb. Run the script and make sure it's working.

7. Defining a method (3)

Now that you have two method definitions under your belt, define an original method that tells something about the user and meet with the theme of the online dating site. Add/modify the attributes of the class if needed. For example: you may create new attributes for height and weight, and create a method that calculates the Body Mass Index (BMI) of the user, or you may create an attribute for annual income and come up with a method that estimates of the lifetime income of the user based on the age.

Provide a 1-2 line description of what the method does as a comment in your source code.

Once the method is implemented, insert a statement that calls the method of u1 at the bottom of Lab1.rb. Run the script and make sure it's working.

8. Submission

The due for this lab would be the midnight of Monday, September 8th, following our rule. HOWEVER, since we don't have time today to cover the submission process, **you will submit the lab during the class on Tuesday, September 9th.** The instructor will show you how. You will have longer time to work on for the later lab sessions, but for the first few labs, I would like to keep it short, so that you will receive quick feedback. I will consider accepting resubmissions, so try to submit something, by the due.

9. Evaluation

This Lab will be graded by the following criteria:

Defining a class (2pt)	The code includes a correct definition of the User class.
Creating an object (2pt)	The code includes a correct use of the User class through an object.
Creating an method 1 (2pts)	The code includes a correct definition of a User method, intro.
Creating an method 2 (2pts)	The code includes a correct definition of a User method, age.
Creating an method 3 (2pts)	The code includes a correct definition of a User method, designed by the student.