TINKER ACADEMY

Mobile App Development

Handout 1: Welcome & Getting Started

Note your Student ID. You will need to use it throughout the Course.

Connecting to the Network

1. Select “Cupertino Community Center” if your Student ID is divisible by 2. Else choose “Cupertino Community Center 3”.
2. Open a browser (preferably Chrome, Safari or Firefox)
3. Type in “www.google.com” (without the quotes). Type in *exactly* as indicated. If nothing shows up, check again, did you include the 2 “dots”? Did you spell www google and com correctly.
4. You should be taken to a Cupertino Parks and Recreation. Accept the agreement and click the required buttons to activate the network.

Installing the Virtual Machine (VM)

1. Borrow the USB drive from your Tinker Academy instructor
2. Create the folder “tinkeracademy” (without the quotes) under Documents using Finder or Windows Explorer. Type it in *exactly* as indicated.
3. Copy the folder “installers” from the USB drive to under “tinkeracademy” using Finder or Windows Explorer
4. Eject the USB Drive and hand it back to the Tinker Academy instructor
5. Locate the VirtualBox installer under “tinkeracademy” using Finder or Windows Explorer
6. Install the VirtualBox application
7. Congratulations, You completed a major milestone. Give yourself a pat on the back :)

Importing the Virtual Machine (VM)

1. Locate the Virtual Machine “tinkeracademy.ova” under “tinkeracademy”
2. Double click on “tinkeracademy.ova”. You should get the import screen in VirtualBox with an “Import” Button. Click on the “Import” button to Import the Virtual Machine.

Starting the Virtual Machine (VM)

1. Once the Import is complete and successful, you should see the VM “TinkerAcademy” in the side panel in VirtualBox.
2. If it says “Powered Off” click on the Start Button (Green Arrow) in the VirtualBox Toolbar. This will start the VM.
3. If it says “Running” click on the Show Button (Green Arrow) in the VirtualBox Toolbar. This should display the VM window.

4. Once the VM starts up you will be presented with a login screen. Type in “password” without the quotes. Type it in exactly as indicated and hit “Enter”.
5. Once the login is completed you should see a Desktop with a few icons. The Screen might go fuzzy for a few seconds before displaying the Desktop. *That is ok.*
6. Congratulations. You are now running Linux within your laptop.
7. Double click on the “Firefox” icon in the Sidebar. This should launch Firefox. Verify you have network access. Close “Firefox”

Launching the Virtual Machine in Full Screen

1. Use the VirtualBox menu View->Switch to Fullscreen to switch the VM to fullscreen mode
2. Use the same VirtualBox menu View->Switch to Fullscreen to switch the VM back out of fullscreen mode

Registering the Virtual Machine

1. Identify the “Home Folder” among the icons in the left side bar. *It’s not that difficult.* Look for the “Home Icon”. This will launch Nautilus, the file browser for Ubuntu Linux. Nautilus works pretty much the same way as Windows Explorer or Finder.
2. Locate “Bookmarks” in the Nautilus sidebar. Click “Setup” which will open up the “Setup” folder.
3. Locate the file “tinker academy.config”. We are now going to edit the file. Right click, select “Open with Sublime Text 2”. This will launch the text editor. You should see a single line which looks something like this

   StudentId=2014000

4. Replace 2014000 with your Student Id. *Yes, you can do it*. *Make sure you don’t add any extra spaces or other characters, just your student id*. For example if your Student Id is 2014004, you should have

   StudentId=2014004

5. Hover the mouse pointer near the top of the document and you should see the “Sublime Text 2” application menu. Select File->Save to save the file. Alternatively use Ctrl S to save the file.

Shutting Down the Virtual Machine

1. Click on the red close window button (to the top left on a Mac, top right in Windows).
2. You will prompted with a confirmation message asking if you want to “Power Off” the machine. Click the button to confirm power off.
3. In a few minutes the VM will shut down and you should see the VirtualBox side panel with the “Tinker academy” VM indicating “Powered Off”

Restarting the Virtual Machine

1. Start VirtualBox
2. Click on the VM “TinkerAcademy” in the VirtualBox side panel
3. Click on the Start Button (Green Arrow) in the VirtualBox Toolbar. This will start the VM.
4. Once the VM startup you will be presented with a login screen.

Right Click in VM on Mac

1. Open System Preferences, Trackpad
2. Enable “Secondary Click”, Toggle the small arrow to the right and select “Click with two fingers”.

Updating the Course

1. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Update”. Choose “Run”. Notify an Instructor if you see a window popup with the message “update course failed”. *You messed up. No, just kidding :). We’ll fix it for you.*
   If you are doing this after class hours:
   Hop onto Skype, and request help in the class chat group.
2. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-GME-1 and locate the “Quiz0.odt” under “quiz0” (which is under “quiz”). Select the file.
3. Double click Quiz0.odt to open it in LibreOffice 3
4. Answer the 5 questions in the Quiz. Once you are done, navigate to the top to see the menu and select File->Save. Alternatively use Ctrl S

Submitting Homework

1. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Submit”. Choose “Run”. Notify an Instructor if you see a window popup with the message “submit course failed”.

What is Lua?

Lua is a powerful computer programming language. It was invented to be simple, but extremely powerful. Lua is an interpreted language which also means its easy to use.

Lua is a programming language similar to C but its designed to be easier to use. It uses simple but powerful concepts. It is interpreted. In a few keystrokes you can do much more than you do in a language like Java. This lets you you “focus” on the program and less on the syntax.

The Origins and History of Lua

Lua was invented by professors at the University of Rio de Janeiro, Brazil in 1993.

Lets Step back .. What is a Program?

A program is a set of instructions.
These instructions can be for anyone or anything. Consider this

"Wingardium Leviosa"

"Wingardium Leviosa" is an instruction from a student of Hogwarts School of Witchcraft and Wizardry to his or her wand to make objects fly or levitate.

Here the instructions are meant for the wand. If the incantation is correct, the wand "magically" "follows" the instructions and causes the objects pointed to to fly or levitate.

Computers work similarly but instead of spell incantations, they understand a coded set of instructions. Specifically the CPU of a computer can "follows" these instructions (The CPU is the brain of the computer).

Here is the first few instructions of a program that the CPU can understand (Specifically the Intel x86 CPUs). The complete program will displays "Hello World".

```
cffa edfe 0700 0001 0300 0080 0200 0000
1000 0000 b005 0000 8500 2000 0000 0000
1900 0000 4800 0000 5f5f 5041 4745 5a45
```

These instructions are encoded using special numbers called hexadecimal numbers. But that is not the most important thing to know right now.

The most important thing to know is that we (humans) find it tedious to write programs using these instructions. So we Muggles (being oh, so clever) came up with an Idea!. What if we create another set of instructions that *we* can create more easily. Then (and here is the clever part), we will take these instructions and translate them to the hexadecimal instructions that the CPU can understand.

So we came up with something like this that makes a little bit more sense. The example below is the complete C Program that will display "Hello World". You can now clearly see the text "Hello World" in the program.

```
include <stdio.h>
int main()
{
   printf("Hello World");
   return 1;
}
```

Its not important that you understand the C program above. What is important is that the C program needs to get translated into the hexadecimal instructions before the CPU can follow them.

When a CPU “follows” an instruction, it means “executes” the instructions.

What is an Interpreted Language?

Lua is an interpreted language. C is a compiled language. What does this mean?

Compiled Language

Compile the program into machine code
Execute the program

Interpreted Language

Execute the program using the python interpreter

Getting Started in Lua

We are going to write our first Lua Program. Along the way, we will learn enough to get a Lua program up and running. We will use Sublime Text 2 to create our Lua programs.

Open the Program

1. Click the “Setup” folder in “Nautilus” under “Bookmarks”
2. Double click “Course Update”. Choose “Run”.
3. Click the “Courses” folder under “Bookmarks”. Navigate to the TA-GME-1 and locate the “StarterPack1.lua” under “starterpack1” (which is under “starterpack”).
4. Click the “Desktop” folder in “Nautilus” under “Computer”. Double click the “Sublime Text 2” icon to launch Sublime Text 2. Sublime Text 2 should display the folder “tinker academy” in the left pane.
5. Navigate to “Courses”, “TA-GME-1”, “starterpack”, “starterpack1”. Open “StarterPack1.lua”
6. Open Terminal and run the 3 commands below exactly as shown below including the dot (.).

```
cd

source .bash_profile

golua1
```

Run the Lua Program

1. Type in the following

   lua StarterPack1.lua

   The lua interpreter will run the StarterPack1.lua program.

   You should see the text “Hello World” output in the Terminal

2. Congratulations! You have successfully written your first Lua Program. *High five your nearest Tinkerer who has also completed. If you are the first, give yourself a pat on the back and ask to High five the Instructor!*

About the Hello World Program Tradition

It has been the accepted practice among computer programmers to introduce students to a new language using a very simple program called the “Hello World” program. The program just prints “Hello World” to the output.

Modify the Lua Program

1. Open the “StarterPack1.lua”.
2. Remove the lines --[[and --]]

```
--[[
function fact (n)
      if n == 0 then
            return 1
      else
            return n * fact (n - 1)
      end
end
--]]

--[[
print("Enter a number:")
a = io.read("*n")
```

```
print(fact(a))
--]]
```

2. Run the Lua program
   Lua will now prompt you for a number. Enter a number between 1 and 10 and press the Enter key

   You should see the factorial of the number printed to the screen.

   A factorial is a number multiplied by every number below it until 1. It is not important that you know what a factorial is for this class.

   It is important you understand the following
   - Every Lua program is stored in a file that ends in .lua
   - You invoke a Lua program using the Lua Interpreter as shown above

Understanding a Lua Program

Open the file StarterPack1.lua

- The first line is a print statement. It is a command to the Lua Interpreter to print the text “Hello World” to the screen.
- Lines 4-10 define a Lua function. Functions are small programs that carry out a specific task when they are called and return a value. We will be covering functions in more detail in the next few classes.
- Line 14 prints a text string similar to the first line.
- Line 16 is special. It first calls the function fact, takes the value returned by the function and prints **that value** to the screen

Variables in Lua

A Variable is a placeholder for a value. It is very easy to create a variable in lua.

myVariable = 2 will create a variable myVariable in lua and set it to the value 2

Expressions in Lua

An expression is a combination of other expressions, text strings, variables which when combined together and evaluated results in a value

2 * myVariable is an expression. If myVariable has the value 2, then the value of the expression is 4

Text expressions can be created using a special operator .. (2 dots)

print(“Tinker”..”Academy”) will print the text “TinkerAcademy” in the Terminal

Quiz 1: Welcome & Getting Started

Import the Quiz

1. Run the “Course Update” script under “Setup”
2. Open Quiz1.odt under “Courses” TA-JAV-1 “quiz” “quiz1”
3. Attempt each question. Type in the answers in the “Answer:” box.

Submitting the Quiz
Open Nautilus. Run “Course Submit” under “Setup” to submit the quiz. Notify an Instructor if you see a window popup with the message “submit course failed”. Else, logon to skype and ping the class chat for help.

Homework 1: Welcome & Getting Started

Overview

In this homework you will print some more text to the Console.

Open the Program

1. Run the “Course Update” script under “Setup”
2. Open Homework1.lua under “Courses” TA-GME-1 “homework” “homework1”
3. Open Terminal
4. Type in the following commands

```
cd

source .bash_profile
```

“goluahomework1”

5. Run the program in Terminal using the following command
   “lua Homework1.lua”

Submitting Homework

Make sure you are on WiFi. Open Nautilus. Run “Course Submit” under “Setup” to submit the homework.Notify an Instructor if you see a window popup with the message “submit course failed”. Else, logon to skype and ping the class chat for help.