Walkthrough of the lab computing environment: Introduction to CentOS, Linux Command Line, Eclipse and Submission

Site:	Moodle@York
Course:	LE/EECS1710 A - Programming for Digital Media (Fall 2018-2019)
Book:	Lab 0 - Preliminaries
Printed by:	Matthew J Kyan
Date:	Saturday, 8 September 2018, 5:41 AM

Table of contents

- 1. Preamble
- 2. CentOS
- 3. The Linux Terminal
- 4. The Eclipse IDE
- 4.1. Basic interface
- 4.2. Creating a new project
- 4.3. "Hello World"

1. Preamble

This is a self guided lab that walks you through the EECS PRISM lab environment, including a brief look at configuring look and feel of CentOS, some introduction to basic Linux commands (in the terminal), The Eclipse IDE, and creating a basic "Hello World" java application.

2. CentOS

To begin, find a workstation in the Prism lab and log in. Your user name and password are the same as for your Passport York account.

The lab computers you are using run a Linux operating system called CentOS. The default desktop environment is called GNOME. The basic funtionality of the desktop environment is similar to that provided by Microsoft Windows and Mac OS X.



The default user interface provided by GNOME is minimalist which might make it difficult for new users to figure out how to do things. What follows is a list of instructions for customizing the user interface so that it displays more information to the user. You can easily undo these changes as you become more comfortable with the desktop environment.

Click the Activities button in the upper left corner of the screen. In the search bar type tweak and the Tweak Tool icon should appear. Click on the Tweak Tool icon.



The Tweak Tool application should appear. It allows the user to adjust certain aspects of the desktop environment. In the left column you will see the categories of tweaks that can be adjusted by the user.

Q Tweaks	Appearance	×
Appearance	Global Dark Theme	OFF
Desktop	Applications need to be restarted for change to take effect Theme	
Extensions	Window	Adwaita <i>(default)</i> 🔻
Fonts	GTK+	Adwaita <i>(default)</i> 🕶
Keyboard and Mouse	lcons	Adwaita 🔻
Power	Cursor	Adwaita <i>(default)</i> 🔻
Startup Applications	Shell theme 🔺	~
Top Bar		
Typing		
Windows		
Workspaces		

Click the Extensions category. Adjust the ON/OFF switches so that they have the settings shown below.

The Applications menu switch enables a menu that is similar to the Windows Start menu.

The Places status indicator switch enables a menu similar to the Windows This PC category in Windows Explorer.

The *Window list* switch enables a list of the currently running desktop applications at the bottom of the screen (similar to the Windows task bar).

** each of these will re-create a similar feel to your EECS VM (if setup on your own computer - see tutorial section of coursewebsite)

Q Tweaks	Extensions ×
Appearance Desktop	OFF Alternatetab Substitute ait-tab with a window based switcher that does not group Remove
Extensions	Applications menu Remove Add a category-based menu for applications. Remove
Fonts	
Keyboard and Mouse	OFF Launch new instance Always launch a new instance when clicking in the dash or the application view.
Power	ON Places status indicator Remove Add a menu for quickly navigating places in the system. Remove
Startup Applications	Window list
Top Bar	ON Window dst Remove Display a window list at the bottom of the screen. Remove
Typing	Install Shell Extension Get more extensions (None)
Windows	
Workspaces	

Click the *Top Bar* category. If you want to, check the *Show date* box so that the current date appears beside the clock at the top center of the desktop.

Q Tweaks	Top Bar ×
Appearance	Show Application Menu
Desktop	Clock
Extensions	Show date
Fonts	Show seconds
Keyboard and Mouse	Calendar Show week numbers
Power	
Startup Applications	
Top Bar	
Typing	
Windows	
Workspaces	

Click the Windows category. Adjust the ON/OFF switches for Titlebar Buttons so that they have the settings shown below.

These switches enable a minimize and maximize button to appear in the top right corner of an application window. By default, only a close button appears in the top corner of an application window.

Q Tweaks	Windows	_ = ×
Appearance Desktop Extensions	Attached Modal Dialogs Automatically Raise Windows Resize with Secondary-click	ON OFF OFF
Fonts	Window Action Key	Super
Keyboard and Mouse	Focus Mode	Click
Power	Titlebar Actions Double-click	Toggle Maximize 🔻
Startup Applications	Middle-click	None 🔻
Top Bar	Secondary-click	Menu 🔹
Typing	Titlebar Buttons	
Windows	Maximize	ON
Workspaces	Minimize	ON
	Window scaling	1 – +

Close the **Tweak Tool** application by clicking the close button in the upper right corner of the window, or use the *Tweak Tool* menu found near the top left of the screen.

3. The Linux Terminal

Quoting the Introduction to user commands manual entry:

"Linux is a flavour of Unix, and as a first approximation all user commands under Unix work precisely the same under Linux (and FreeBSD and lots of other Unix-like systems).

Under Linux there are GUIs (graphical user interfaces), where you can point and click and drag, and hopefully get work done without first reading lots of documentation. The traditional Unix environment is a CLI (command line interface), where you type commands to tell the computer what to do. That is faster and more powerful, but requires finding out what the commands are."

The Terminal program provides a command-line interface called a *shell* where you type commands to tell the computer what to do. Start the Terminal program using the *Applications* menu.

Favorites	۵	Firefox Web Browser	
Accessories	(5)	Thunderbird Mail	
Documentatio	• >	manacibila Plan	
Education	2	Terminal	
Graphics			
Internet			
Office			
Programming			
Sound & Video			
Sundry			
System Tools			
Universal Acce	SS		
Utilities			
Other			
Activities Over	view		

A window similar to that below should appear:



The prompt ea40 % indicates that the name of this computer is ea40. You will see a different name corresponding to the name of your computer. You will also see a number beside the name (probably the number 1); the number is the history number of the current command line.

In EECS1710, it is important that you become comfortable working with files from the command line because this is how you will be submitting your program files for grading (including during tests).

You are probably already familiar with the notion of a *computer file* that represents some piece of information on a storage device (e.g., MP3 music files, JPEG picture files, word processor documents).

You are probably also aware that you can use *folders* to organize your computer files (e.g., under Windows you have Desktop, Documents, and Favorites folders). You can even put folders inside of other folders.

You might also know that every file and folder has a unique name called the *absolute path* or *absolute pathname*. The absolute path is simply the full list of folders that you need to traverse to reach a particular file or folder. For example, in Windows the the absolute path C:\Users\Bob\Music refers to Bob's Music folder, and the backslashes \ are used to separate folder names.

In Linux, it is more common to use the term *directory* instead of folder.

Your *home directory* is where all of your EECS account files are stored. Of course, you can (and should) create directories inside of your home directory to organize your files. Every person with a EECS account has their own home directory.

No matter which lab computer you use, your home directory always has the same absolute pathname.

By default, the shell will start in your home directory.

To see the full pathname of your home directory, invoke the **pwd** command by typing it in the terminal and pressing the *Enter* key. You should get a slightly different result than that shown below.



pwd prints the full path of the current working directory.

Notice that the terminal you are using has been pre-configured to display the current working directory in the title bar.

Linux uses a single hierarchical directory structure to organize all of its files. This is different than Windows where each drive (C:, D:, etc.) has its own separate directory structure.

The top-most directory in Linux is called the **root** directory, and is identified by the slash character /.

Use the change directory command **cd** to switch to the root directory.

						ea40:/		-	×
File	Edit	View	Search	Terminal	Help				
ea40 /eec ea40 ea40	301 s/hon 302 303	% pwd ne/bur % cd % ∎	ton /						

Use the command ls ("ell-ess" not "one-ess") to list the files contained in the current working directory.

You might not get the exact same output as shown in the figure because the EECS computer systems are regularly updated. Also, if you do this exercise from your home VM setup, you will actually be browsing only files local to your machine (again results will differ).

The names listed in blue are all directory names. Many of these directories are common to all standard Linux distributions, and are described by the Filesystem Hierarchy Standard.

		cu	40.)		-	
File Edit View	Search Terminal	Help				
a40 301 % pwd eecs/home/burt a40 302 % cd / a40 303 % ts CMC dev in cs eecs oot cse etc a40 304 %	ton / home media s lib mnt lib64 obj	opt run proc sbi root srv	sys tmp usr	var xconf xsys		

Notice that the root directory contains other directories. Those directories in turn contain other directories, and so on. One of the directories in the root directory is named **eecs**.

Use the change directory command to switch to the eecs directory.

Notice that you did not need to specify the full path /eecs because you were already in the root directory / (you can always use the full path if you want).

						ea40:/e	ecs			-	•	×
File	Edit Vi	ew S	earch T	erminal H	Help							
ea40 /eecs ea40 1 bin boot ea40 ea40	301 % \$/home/ 302 % CMC CS 303 % CMC CS 304 % 305 %	pwd burtou cd / ls dev eecs etc cd ee Cd ee	home lib lib64 cs	media mnt obj	opt proc root	run sbin srv	sys tmp usr	var xconf xsys				

The eecs directory contains all of the files and directories belonging to people in the Department of Electrical Engineering and Computer Science.

Use the command ls to list the files contained in the current working directory.

Again, you may see some differences from the figure.

	ea	40:/eecs		-	• ×
File Edit View Search Ter	rminal Help				
ad0 301 k pwd (eecs/home/burton aa40 303 k cd / aa40 303 k ls 1 CMC dev home bin cs eecs lib boot cse etc lib64 bad0 304 k cd eecs aa40 305 k ls course dept fac home bad40 306 % ■	media opt n mnt proc s obj root s local ltsave	un sys bin tmp rv usr researc	var xconf xsys h		

Many of the Linux command-line programs allow the user to specify optional flags that control how the program should behave. The flags are often (but not always) specified using a hyphen -.

Use the long format flag -l ("hypen-ell") with ls to display extra information about the files in the current working directory.

The long format listing produces a lot of output, most of which won't make sense to you at this time. We may revisit these details later in the course.

ea40:/eecs	-	۰	×
File Edit View Search Terminal Help			
ea40 301 % pwd			
ea40 302 % cd /			
ea40 303 % ls			
1 CMC dev home media opt run sys var			- 1
bin cs eecs lib mnt proc sbin tmp xconf			
boot cse etc lib64 obj root srv usr xsys			
ea40 304 % cd eecs			
ea40 305 % Ls			
course dept rac nome tocat itsave research			
total 12			
lrwxrwxrwx 1 root root 25 May 1 08:42 course -> /eecs/dept/cou	rse/o	unne	nt
drwxr-xr-x 10 root root 4096 Aug 30 14:50 dept			
drwxrwsr-x 13 root faculty 4096 May 15 15:18 fac			_
drwxr-xr-x 3 root root 0 Sep 5 14:32 home			
drwxrwxr-x 22 root tech 4096 Feb 8 2017 local			_
drwxr-xr-x 2 root root 0 Sep 1 10:21 ltsave			
drwxr-xr-x 2 root root 0 Sep 1 10:21 research			
ea40 307 %			
			- 1
			- 1
			_

Even expert users have a hard time remembering all of the options for all of the commands. All of the commonly used command-line programs have a manual page that you can access using the man program.

Access the manual for Is using the command man Is

	ea40:/eecs	- • ×
File Edit View Search Termi	inal Help	
ea40 301 % pwd		
/eecs/home/burton		
ea40 302 % cd /		
ea40 303 % ls		
1 CMC dev home me	adia opt run sys var	
bin cs eecs lib mr	nt proc sbin tmp xconf	
boot cse etc lib64 ob	oj root srv usr xsys	
ea40 304 % cd eecs		
ea40 305 % ls		
course dept fac home l	local ltsave research	
ea40 306 % ls -l		
total 12	05 11 1 00 10	
lrwxrwxrwx 1 root root	25 May 1 08:42 course -> /eecs/	dept/course/current
drwxr-xr-x 10 root root	4096 Aug 30 14:50 dept	
drwxrwsr-x 13 root faculty	/ 4096 May 15 15:18 Tac	
drwxr-xr-x 3 root root	4006 Est 0 2017 Jacob	
drwxrwxr-x 22 root tech	4090 Feb 8 2017 Local	
drwr yr y 2 root root	0 Sep 1 10:21 ((Save	
and0 207 % map 1c	o sep i io.zi research	
	ea40:/eecs	- • ×
File Edit View Search Termi	inal Help	
LS(1)	User Commands	LS(1)
NAME		
ls - list directory	/ contents	
SYNOPSIS ls [OPTION] [FIL	.E1	
DESCRIPTION List information Sort entries alphab fied.	about the FILEs (the current direc petically if none of -cftuvSUX nor -	tory by default). -sort is speci-
Mandatory argument too.	ts to long options are mandatory	for short options
-a,all		
do not ignor	re entries starting with .	
-A,almost-all do not list	re entries starting with . implied . and	

You can move forward in the manual by pressing the *Space* key. Move backwards by pressing the b key. Quit using the q key.

Is has a large number of optional flags. Don't worry about trying to understand them all right now; just get used to navigating the man page, but keep the following questions in mind.

What flag would you use to:

- list files in reverse order?
- list one file per line?

Try out your answers; did they do what you expect?

Recall that we are currently in the directory /eecs.

To move up one directory to the parent directory you use the command **cd**.. (which in this case will take us to the root directory).

The shell interprets .. as being the parent directory to the current working directory.



The **Is** command by itself will list the contents of the current working directory. If you want to list the contents of a different directory, you can specify the name of the directory.

Is bin will list the contents of the bin directory.

The directory **/bin** contains many of the essential command-line programs.



There are quite a few files in **/bin**. The files listed in green are all programs runnable by the user.

	ea40:/	-	×
File Edit View Search Terminal Help			
mysql waitpid	zegrep		
mysql zap	zeisstopnm		
nzip	zenheiset		
	zenity		
namei	zfgrep		
nano	zforce		
nasm	zarep		
	zip		
nautilus	zipcloak		
nautilus-autorun-software	Zipcmp		
nautilus-connect-server	zipdetails		
nautilus-sendto	zipgrep		
	zipinfo		
ncat	zipmerge		
ncurses5-config	zipnote		
ncursesw5-config	zipsplit		
ndiff	ziptorrent		
ndisasm	zless		
ndptool	zmore		
	ZDBW		
needs-restarting	zonetab2pot.py		
nenscript	zsh		
neotoppm	zsoelim		
ea40 310 %			

Use the change directory command to switch to the bin directory.

	ea40:/usr/bin	-	
File Edit View Search Terminal Help			
mysql zap	zeisstopnm		
nzip	zenheiset		
nail	zenity		
namei	zfgrep		
nano	zforce		
nasm	zgrep		
native2ascii	zip		
nautilus	zipcloak		
nautilus-autorun-software	zipcmp		
nautilus-connect-server	zipdetails		
nautilus-sendto	zipgrep		
nc	zipinfo		
ncat	zipmerge		
ncurses5-config	zipnote		
ncursesw5-config	zipsplit		
ndiff	ziptorrent		
ndisasm	zless		
ndptool	zmore		
neato	Znew		
needs-restarting	zonetab2pot.py		
nenscript	zsh		
neotoppm	zsoelim		
ea40 310 % <u>c</u> d bin			
ea40 311 %			

The **ls** command lists information about the files you specify. As you have already seen, if you specify no files, then **ls** lists the files in the current working directory.

Suppose that you want to list all of the files that begin with the letter l. You can do this by using the command **ls l***

Most Linux shells will interpret the asterisk * to mean "any sequence of characters"; thus, **Is I*** means "list all files with a name starting with I followed by any sequence of characters".

Notice that the ls command is found in /bin.

	ea40:/usr/bin		-	×
File Edit View Search	th Terminal Help			
neotopps	zsoelim			
ea48 318 % cd bin				
ea43 311 % ls l*				
12ping	lexgrog	lp.cups		
12test	lftp	lpoptions		
lancelot	lftpget	lppasswd		
last	libart2-config			
	libbluray test	lpg.cups		
lastcomm	libgcrypt-config	lpr		
lastlog	libglade-convert	lpr.cups		
	libanutls-config	lpra		
latrace	libguestfs-test-tool	lprm.cups		
latrace-ctl	libIDL-config-2	lprsetup.sh		
lchfn	libigee1284 test	lp solve		
lchsh	libnetcfg			
lconvert	libonal5-config	lostat.cups		
lconvert-at5		lpunlock		
	libreoffice	lrelease-gt4		
ld.bfd	libtar	lrelease.gt5		
ldd	libtool	ls		
blop.bl	libtoolize	lsattr		
Idns-chaos	libusb-config	lsblk		
Idns-compare-zones	libwacow-list-local-devices	lsb release		
Idosd	libert-fortmap	Iscaroup		

You can return to your home directory by using the **cd** on its own. If you don't specify which directory to change to, **cd** assumes you want to go to your home directory.

You could also use $cd \sim$ (where \sim is the "tilde" character usually located on the top left key of the keyboard).

	ea40:/eecs/hor	ne/burton	-	>
File Edit View Sea	rch Terminal Help			
ldns-notify	linux-boot-prober	lsed		
Lons-nsec 3-hash	lispatopga	lsns		
ldns-read-zone	list titles	lsscsi		
ldns-resolver		lssubsys		
Ldns-revoke	lo	lstope-ne-graphics		
ldns-rrsig	Indir	lsusb		
ldns-signzone	Ineato	lsusb.pv		
ldns-test-edns	Loadkeys	ltrace		
ldns-testns	loaduniwap	lua		
ldns-update	locale	luac		
ldns-verify-zone	localectl			
lons-version	localedef	luatex		
ldhs-walk	local-getcert	luit		
lons-zcat	locate	lupdate-ot4		
ldns-zsplit	logger	lupdate-gt5		
leaftoppa	login	lwp-download		
lefty	loginctl	lwp-dump		
less	Logname	lwp-mirror		
lessecho	logresolve	lwp-request		
lesskey	logview	Lynx		
lesspipe.sh	Look			
		lzop		
ea48 312 % cd				
ea48 313 %				

** we will learn more commands for the terminal as we progress through the course.

Summary

Command	Meaning
pwd	Print the full (absolute) pathname of the current working directory.
cd dirname	Change to the named directory.
cd	Change to the parent directory of the current working directory.
cd	Change to the user's home directory.

ls	List the contents of the current working directory.
ls dirname	List the contents of the named directory.
ls -l	List using long format the contents of the current working directory.
ls -l dirname	List using long format the contents of the named directory.
ls -ld <i>dirname</i>	List using long format the name (but not the contents) of the named directory.

Clipse IDL		
hat is Eclipse? Eclipse an IDE (Integrated		
evelopment wironment), meaning		
at it is a program made		
provide you with all the		
apped up in a nice user		
terface.		
y features include:		
utomatic errordetection		
d highlighting: so that		
our code quickly:		
- sublic class HalleHarld		
public class Helloworld		
_ public static void i	main(String[] args) {	
String line1 =)	"Hello, the red buttoned x on the left means there's so	mething.wrong.herEl
String line2 =	"Yup, definitely missed a semicolon on the last line.";	Press 'F2' for foc
}		
3		
itocomplete/Content sist: similar to edictive text, as you type		
atocomplete/Content sist: similar to edictive text, as you type function you are thinking using, it will suggest ssible functions and ve you an idea of their		
utocomplete/Content sist: similar to edictive text, as you type function you are thinking using, it will suggest vessible functions and ve you an idea of their ntax		
<pre>intocomplete/Content sist: similar to edictive text, as you type unction you are thinking using, it will suggest ssible functions and ve you an idea of their ntax public class HelloWorld</pre>	{	
<pre>intocomplete/Content sist: similar to edictive text, as you type unction you are thinking using, it will suggest ssible functions and ve you an idea of their ntax public class HelloWorld public static void restatic void restatic void restatic</pre>	{ main(String[] args) {	
<pre>intocomplete/Content sist: similar to edictive text, as you type unction you are thinking using, it will suggest ssible functions and ve you an idea of their ntax public class HelloWorld public static void r double power = r </pre>	{ main(String[] args) { Math.p	
<pre>ntocomplete/Content sist: similar to edictive text, as you type function you are thinking using, it will suggest ssible functions and we you an idea of their ntax</pre>	{ main(String[] args) { Math.p.	pow
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking using, it will suggest sssible functions and we you an idea of their ntax</pre> public class HelloWorld public static void n double power = n }	{ main(String[] args) { Math.p [¶] PI:double - Math	<pre>pow public static double pow(double a,</pre>
<pre>ntocomplete/Content sist: similar to edictive text, as you type function you are thinking using, it will suggest ssible functions and we you an idea of their ntax public class HelloWorld public static void r double power = r } </pre>	{ main(String[] args) { Math.p.	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest sssible functions and ve you an idea of their ntax public class HelloWorld public static void r double power = r } </pre>	{ main(String[] args) { Math.p ^{§F} PI:double - Math ^{§F} pow(double a, double b):double - Math	<pre>pow public static double pow(double a,</pre>
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest vessible functions and ve you an idea of their ntax public class HelloWorld public static void t double power = t } </pre>	{ main(String[] args) { Math.p. ^{§F} PI: double - Math ^{§F} pow(double a, double b): double - Math	<pre>pow public static double pow(double a,</pre>
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest ussible functions and ve you an idea of their ntax</pre>	{ main(String[] args) { Math.p	<pre>pow public static double pow(double a,</pre>
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest sssible functions and ve you an idea of their ntax public class HelloWorld public static void t double power = t } </pre>	{ main(String[] args) { Math.p. % PI:double - Math % pow(double a, double b):double - Math	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power of the second argument. Special cases: • If the second argument is positive or negative zero, then the result is 1.0. • If the second argument is 1.0, then the result is the same as the first argument. • If the second argument is NaN, then the result is NaN. • If the first argument is NaN and the second argument is nonzero, then the result is NaN
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest sssible functions and ve you an idea of their ntax public class HelloWorld public static void r double power = r } } </pre>	{ main(String[] args) { Math.p. % PI:double - Math % pow(double a, double b): double - Math Press 'Ctrl+ Space' to show Template Proposals	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power of the second argument. Special cases: If the second argument is positive or negative zero, then the result is 1.0. If the second argument is 1.0, then the result is the same as the first argument. If the second argument is NaN, then the result is NaN. If the first argument is NaN, then the result is NaN. If the first argument is NaN and the second argument is nonzero. then the result is NaN. Press Tab' from proposal table or click for focus
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest ussible functions and ve you an idea of their ntax</pre>	{ main(String[] args) { Math.p % PI:double - Math % pow(double a, double b):double - Math % pow(double a, double b):double - Math % Press 'Ctrl+Space' to show Template Proposals	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power of the second argument. Special cases: If the second argument is positive or negative zero, then the result is 1.0. If the second argument is 1.0, then the result is the same as the first argument. If the second argument is NaN, then the result is NaN. If the first argument is NaN and the second argument is nonzero, then the result is NaN. Press 'Tab' from proposal table or click for focus
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest ssible functions and we you an idea of their ntax public class HelloWorld public static void r double power = r } } </pre>	{ main(String[] args) { Math.p. % PI: double - Math % pow(double a, double b): double - Math Press 'Ctrl+Space' to show Template Proposals	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power of the second argument. Special cases: If the second argument is positive or negative zero, then the result is 1.0. If the second argument is 1.0, then the result is the same as the first argument. If the second argument is NaN, then the result is NaN. If the first argument is NaN and the second argument ic nonzero. then the result ic NaN Press 'Tab' from proposal table or click for focus
<pre>utocomplete/Content sist: similar to edictive text, as you type function you are thinking 'using, it will suggest 'sssible functions and we you an idea of their ntax public class HelloWorld public static void r double power = 1 } } } ill in compiler: so that u can run code at the</pre>	{ main(String[] args) { Math.p V PI:double - Math V PI:double a, double b):double - Math Press 'Ctrl+Space' to show Template Proposals	pow public static double pow(double a, double b) Returns the value of the first argument raised to the power of the second argument. Special cases: • If the second argument is positive or negative zero, then the result is 1.0. • If the second argument is 1.0, then the result is the same as the first argument. • If the second argument is NaN, then the result is NaN. • If the first argument is NaN and the second argument is nonzero, then the result is NaN. • Press 'Tab' from proposal table or click for focus



And a debug

mode: to help you find errors and where they occur in your program

** of course there are many other features too!

4.1. Basic		
interface		
Start		
up		
application		
"Eclipse"		
which		
is		
located		
under		
the		
"Applications		
->		
Programming		
->		
Ecupse"		
down		
menu		
Once		
t is		
run,		
you		
will		
be and the second se		
greeted		
with		
à		
screen		
vnicn		
viii look		
something		
ike		
his:		
	Edipse Laureber	Y
	E cupse Lauriner	^
Select a directory as workspace		
Eclipse uses the workspace directory	to store its preferences and development artifacts.	
		0

elect a dire	ectory as works	space			
Eclipse uses	the workspace di	rectory to store it	s preferences a	nd developme	nt artifacts.
Norkspace:	/home/user/wor	rkspace		•	Browse
I lee this a	s the default and	do not ask again			
		i do not ask again			
Recent Wo	*kspaces				
				Cancel	ОК
				Currect	

Keep
the
default
settings,

you	
will	
be	
setting	
up	
code	
projects	
within	
the	
'workspace'	
folder	
on	
your	
home	
directory.	
You	
can	
use	
the	
terminal	
later	
to	
browse	
this	
directory	
and	
see	
the	
files	
that	
are	
generated	
when	
you	
create	
new	
Java	
programs.	
After	
clicking	
'ok',	
the	
main	
interface	
will	
-44	
start.	
At	
At this	
At this point	
At this point you	
At this point you may	
At this point you may be	
At this point you may be presented	
At this point you may be presented with	
At this point you may be presented with the	
star. At this point you may be presented with the "Welcome	
At this point you may be presented with the "Welcome to	
At this point you may be presented with the "Welcome to Eclipse"	
At this point you may be presented with the "Welcome to Eclipse" tabbed	
At At this point you may be presented with the "Welcome to Eclipse" tabbed pane.	
At At this point you may be presented with the "Welcome to Eclipse" tabbed pane. You	
At At this point you may be presented with the "Welcome to Eclipse" tabbed pane. You can	

this			
for			
now			
(click			
on			
the			
'X').			
You			
can			
look			
at			
it			
later			
by			
looking			
under			
Help			
->			
Welcome			
You			
might			
like			
to			
try			
some			
of			
the			
basic			
tutorials			
later			
to			
become			
more			
familiar			
with			
the			
interface			
itself.			
For			
now,			
the			
app			
will			
look			
something			
like			

r

		workspace	– Java	– Eclipse				-		×
File Edit Source Refactor	Navigate Sear	h Project	Run	Window	Help					
📑 🕶 🔚 🕼 📄 🔍 📲	8 🞯 🕶 🤷 🕶	* • • •	· 🤬 •	1 🔊 🥲	× -	<u></u> ₽ • ÿ •	• * \$ \$ • •	⇒ •		
						Quid	k Access	Ê	2	8
ቹ Package Exp 🛿 🗖 🗖						🗐 Task Li	st 🖾			
Ē\$ ₽ ▼						☆	₽ <u>+</u> ₽ <u>+</u> ₽	×	6 8	
						Find	All +	Acti	vate	?
						🗄 Outline	: 🖾 😜	▽		
						An outline	is not availabl	e.		
	🔝 Problems 🛛	@ Javadoc	🕒 De	claration	📮 Conso	le		▽		
	0 items									
	Description						Resource	P	ath	
					1					

The	
window	
above	
shows	
a	
set	
of	
four	
"views"	
arranged	
into	
a	
"perspective".	
This	
particular	
configuration	
of	
views	
is	
called	
the	
"Java	
Perspective".	
We	
will	
look	
into	
each	
of	
the	
four	

"views"		
(clockwise,	2,	
starting		
from		
top		
left):		
a 41-a		
• the		
Evelope	32 	
view		
• the		
• uic editor		
view		
(not		
labelled		
the	·	
big		
empty		
square		
in		
the		
middle)		
• the		
"Outline'	e"	
view		
(we		
can		
dismiss		
the		
"task"		
view		
for		
now		
to		
simplify	ý	
this		
to		
4		
views)		
• a		
consistin	na	
of		
a		
set		
of		
several		
tabbed		
panes		
(labelled	d	
Problems	ns,	
Javadoc,	>,	
Declarati	tion,	
Console)		
Use		
Window		
->		
Perspective	ve	
->		
Open		

Perspective	
in	
order	
to	
select	
the	
"Java	
Browsing"	
and	
then	
the "	
"Debug"	
This	
niis will	
demonstrate	
to	
vou	
what	
a	
few	
different	
perspectives	
look	
like.	
Essentially	
each	
"perspective"	
configures	
the	
IDE	
for	
a	
different	
mode	
of	
operation	
(e.g.	
debug	
in	
15 tailorad	
to	
tracking	
what	
is	
happening	
as	
your	
code	
executes,	
the	
original	
iava	
perspective	
is	
setup	
for	
editing	
files	
and	

them,
etc.)
Switch
back
to
the
Java
Perspective.
Use
Window
te
do
this.
Alternatively,
there
are
also
short-
cut
buttons
to
switch
· · ·
view in
III the
linner
right-
hand
corner.
Holding
the
mouse
over
a
button
will
reveal
u tooltin
description.
1
HINT:
lf
you
ever
nnu vourself
in
the
wrong
perspective,
you
can
always
get
to
the
Java

Perspective	
using	
Window	
->	
Perspective.	
Now	
we're	
back	
in	
the	
Java	
Perspective.	
Let's	
have	
a 1	
look	
at	
niter .	
eanor	
view.	
11	
the	
hio	
empty	
pane	
,	
the	
middle	
of	
the	
window.	
The	
title	
bar	
is	
empty,	
but	
you	
can	
see	
the	
Dar descriptory	
are	
there	
(e.g.,	
to	
minimize	
and	
maximize	
the	
view).	
Once	
there	
is	
a	
file	
open	
within	

Eclipse,
this
view
will
be
populated
with
content
(but
at
this
point
you
have
no
project
nor
any
files).
Now
let's
look
on
the
lefthand
pane
at
the
"Package
Explorer"
view.
1 his
15
-1
an
vour
nniects
will
he
listed
(as
VOU
create/import
them).
It
is
shown
below



4.2.			
Creating			
а			
new/			
nroiect			
project			
Create			
anew			
Project			
called			
" 1710-			
lab01 "			
by			
File			
>			
New			
>			
Project			
and			
selecting			
the			
Java			
Project			
wizard			
Your			
version			
of			
Eclipse			
have			
the			
option			
File			
> New			
>			
Java			
Project.			
If			
so, select			
that			
instead			
and			
then			
select			
tne Java			
Project			
wizard			

	workspace	– Java – Eclipse	- • ×
File Edit <u>S</u> ource Refac <u>t</u> or Navigate	e Se <u>a</u> rch Project	Run Window Help	
New	Shift+Alt+N 🕨	/ Java Project	• *\$
Open File		📑 P <u>r</u> oject	ss 🖻 😫 🐯 🏘 🕵
📮 Open Projects from File System		🖶 Package	line ⊠ 😜 マ 🗆 🗖
Close	Ctrl+W	🎯 Class	ine is not available.
Close All	Shift+Ctrl+W	🞯 Interface	
🔚 Save	Ctrl+S	🞯 Enum	
📓 Save As		Annotation	
🐚 Save All	Shift+Ctrl+S	🛱 Source Folder	
Revert		😕 Java Working Set	
Move		Folder	
🗹 Rename	F2	Lintitled Text File	
Refresh	F5	JUnit Test Case	₽ ₽ □
Convert Line Delimiters To	•	🗂 Task	
👜 Print	Ctrl+P	Evample	Resource Path
Switch Workspace		La c <u>x</u> ampte	
Restart	4	<u> ○</u> ther Ctrl+N	
n Import			
N Export			
Properties	Alt+Enter		
Exit			

When you select the wizard, the New Java Project dialog box appears as shown below. In this dialog, do the following: • In the

the Project name: field, enter 1710-

	lab01
	(see
	below).
•	Under
	JRE.
	leave
	the
	option
	Use
	an
	execution
	JRE
	JavaSE-1.8
	as
	selected
	(see
	below).
	JRE
	for
	"Java
	Runtime
	Environment"
	and
	consists
	of
	the
	Java
	Machine
	(JVM),
	Java
	platform
	core
	classes,
	and
	supporting
	Java
	libraries.
	The
	JRE
	is
	the
	runtime
	portion
	software.
•	Under
	Project Lowent
	Layou,
	the
	option
	Use
	project
	folder
	as
	root

	sources
	and
	class
	files
	(see
	below).
	NOTE:
	The
	other
	option
	Create
	separate
	folders
	for
	sources
	and
	class
	files
	is
	selected
	by
	default,
	but
	we
	do
	<u>not</u>
	want
	that
	option.
	Lagua
•	Leave
	alea
	as-
	15.
•	Select
	the
	Finish
	button.

New Java Pro		×			
Create a Java Project Create a Java project in the workspace or in an ext	P	r			
Project name: 1710-lab0 Use default location Location: /home/user/workspace/1710-lab0 JRE		Browse			
 Use an execution environment JRE: Use a project specific JRE: Use default JRE (currently 'jdk-1.8.0_144') 	JavaSE-1.8 jdk-1.8.0_144	← ← Configure JREs]] <u>-</u>		
Project layout Use project folder as root for sources and class files Create separate folders for sources and class files <u>Configure default</u>					
Working sets Add project to working sets Working sets:	•	New Select			
? < Back Next >	Cancel	Finish			

Next we are returned to the Java Perspective. In Package Explorer view, you will now see а

	workspace - Java - Eclipse	П
example).		
this		
to		
mismatched		
slightly		
is		
name		
project		
(the		
below		
shown		
is		
example		
An		
lab01 ".		
" 1710-		
named		
folder		



Now we will expand the project fully. Click on the little triangle to

the
loft
-£
the
package
name
to
expand
the
folder.
Do
you
see
a
sub-
directory
nomed
This
means
you
DID
NOT
follow
the
instructions
correctly
above.
Delete
the
nroject
(right
(light-
CIICK
on
the
project
and
select
DELETE
from
the
popup
menu).
Repeat
the
steps
above.
You
should
see
a
project
entry
labelled
"IRE
Sustam
LIDRARY
(see
below).
Click
on
the

little

triangle
to
the
left.
You
will
see
many
resources
that
are
available
for
your
project.
They
were
placed
there
automatically
by
Eclipse.
It
will
look
something
like
the
screen
shot
below
(depends
on
your
particular
system).

	workspace -	Java – Eclipse		- • ×
File Edit <u>S</u> ource Refac <u>t</u> or Navigate S	e <u>a</u> rch Project	Run Window Help		
📑 🕶 🔚 🐚 ! 🗳 ! 🕸 🎯 🕶 ! 🎴	• * • • • •	9a 🔹 😕 😂 🛷 🔹 🗎	🖗 🔹 🏹 🔹 🏷 🗢	
		(Quick Access	12 茶 🕲
😫 Package Explorer 🕱 📃 🗖		-	🗖 🗄 Outline 🛙	- 8
E 😫 🗊 🗸				5 9
🔻 🚧 1710-lab0			An outline is not	available.
▼ 🛋 JRE System Library [JavaSE-1.8]				
🕨 🔤 resources.jar - /eecs/local/pkg/				
🕨 📅 rt.jar - /eecs/local/pkg/jdk-1.8.				
🕨 🔤 jsse.jar - /eecs/local/pkg/jdk-1.				
🕨 🔤 jce.jar - /eecs/local/pkg/jdk-1.8				
🕨 🔤 charsets.jar - /eecs/local/pkg/jc				
🕨 💀 jfr.jar - /eecs/local/pkg/jdk-1.8.				
🕨 👼 cldrdata.jar - /eecs/local/pkg/jd	🔝 Problems 🛛	@ Javadoc 😟 Declaratio	n 📮 Console 🛛 💱	▽ □ □
🕨 👼 dnsns.jar - /eecs/local/pkg/jdk-	0 items			
🕨 👼 jaccess.jar - /eecs/local/pkg/jdk	Description			Resource
🕨 🚋 jfxrt.jar - /eecs/local/pkg/jdk-1.				
🕨 👼 localedata.jar - /eecs/local/pkg.				
🕨 👼 nashorn.jar - /eecs/local/pkg/jd				
🕨 👼 sunec.jar - /eecs/local/pkg/jdk-				
1710-lab0				



are
some
extra
files
in
this
version).
This
example
runs
several
commands,
and
shows
the
output
for
each
(pwd,
cd
and
ls
-
within
various
directories).

File Edit View Search Terminal Help [user@AP-ESO-ADV-BD ~]\$ pwd /home/user [user@AP-ESO-ADV-BD ~]\$ [user@AP-ESO-ADV-BD ~]\$ ls db Documents local server workspace ws_4413 Desktop Downloads mcode startup.m ws_1021 www
<pre>[user@AP-ESO-ADV-BD ~]\$ pwd /home/user [user@AP-ESO-ADV-BD ~]\$ [user@AP-ESO-ADV-BD ~]\$ ls db Documents local server workspace ws_4413 Desktop Downloads mcode startup.m ws_1021 www [user@AP.ESO-ADV.PD ~]\$</pre>
/home/user [user@AP-ESO-ADV-BD ~]\$ [user@AP-ESO-ADV-BD ~]\$ ls db Documents local server workspace ws_4413 Desktop Downloads mCode startup.m ws_1021 www
[user@AP-ESO-ADV-BD ~]\$ [user@AP-ESO-ADV-BD ~]\$ ls db Documents local server workspace ws_4413 Desktop Downloads mCode startup.m ws_1021 www
[user@AP-ESO-ADV-BD ~]\$ ls db Documents local server workspace ws_4413 Desktop Downloads mCode startup.m ws_1021 www [usor@AP.ESO-ADV.PD ~]\$
db Documents local server workspace ws_4413 Desktop Downloads mCode startup.m ws_1021 www Iusor@AD_ESO_ADV_PD_cl\$
Desktop Downloads mCode startup.m ws_1021 www
[ucoreAD_ESO_ADV_BD1¢
[dset@Ar-ESO-ADV-DD -]\$
[user@AP-ES0-ADV-BD ~]\$ cd workspace
[user@AP-ES0-ADV-BD ~/workspace]\$
[user@AP-ESO-ADV-BD ~/workspace]\$ ls
1710-lab0 RemoteSystemsTempFiles week1
[user@AP-ES0-ADV-BD ~/workspace]\$
[user@AP-ES0-ADV-BD ~/workspace]\$ cd 1710-lab0/
[user@AP-ES0-ADV-BD 1710-lab0]\$
[user@AP-ES0-ADV-BD 1710-lab0]\$ ls
[user@AP-ESO-ADV-BD 1710-lab0]\$ ls -la
total 8
drwxrwxr-x 3 user user 57 Sep 12 03:39 .
drwxr-xr-x. 6 user user 83 Sep 12 03:39
-rw-rw-r 1 user user 289 Sep 12 03:39 .classpath
-rw-rw-r 1 user user 368 Sep 12 03:39 .project
drwxrwxr-x 2 user user 40 Sep <u>1</u> 2 03:39 .settings
[user@AP-ES0-ADV-BD 1710-lab0]\$ []

HINT:

you can clear your terminal window by typing

clear

4.3. "Hello World"						
Within Eclipse, right- click the project name ("1710- 1ab01") in the Package Explorer and select New						
Class:						_
File Edit <u>S</u> ource Refac <u>to</u> 📬 🕶 🔚 🕼 🚍 🔌	r Navigate Se <u>a</u> rch Projec 📽 ▾ ! 🎴 ▾ 🎋 ▼ 💽	ct Run Window ▼ 🌯 ♥ 🕴 🕭 🔑 .	Help	↔ ↔	▼ ♥ *	2
l Package Explorer X	New	•		. ~	-	
E 1710 John	Go Into		Project		ت ble	₽
→ M JRE System Library	Open in New Window		🛱 Package		Dic.	
	Open Type Hierarchy	F4	🗳 Class			
	Show In	Shift+Alt+W 🕨	Interface			
	🗎 Сору	Ctrl+C	G Enum			
	E Copy Qualified Name		Source Folder			
	Paste	Ctrl+V	/ Java Working Set			
-	A Delete		🗳 Folder		-	
	Remove from Context		😭 File		ource	
	Source	Shift+Alt+S	🖹 Untitled Text File			
	Refactor	Shift+Alt+T 🕨	📑 JUnit Test Case			
	🚵 Import		Example			
	🎦 Export		Cther	C+eLLN		
1710-lab0	A B C H		<u>u</u> ue	CULTIN		

In the resulting dialog box, enter Lab01 in the

Name			
field			
and			
select			
the			
checkbox			
where			
Eclipse			
asks			
if			
you'd			
like			
to			
include			
a			

main method:

	New Java Class				×
Java Class		C			
Source folder:	1710-lab0		Brow	/se	
Package:		(default)	Brow	/se	
Enclosing type:			Brow	/se	
Name: Modifiers:	Lab01 public package private abstract final static	protected			
Superclass:	java.lang.Object		Brow	/se	
Interfaces:			Ad	d	
			Rem	ove	
Which method stub	s would you like to create?				
Do vou want to add	 public static void main(String[] args) Constructors from superclass Inherited abstract methods comments? (Configure templates and default) 	value her	e)		
,	Generate comments				
?	Ca	ncel	Fini	sh	

Make sure you capitalize the first

in			
Lab01.			
In			
Java			
conventions			
the			
names			
of			
classes			
begin			
with			
a			
capital			
Thus			
vou			
should			
be			
careful			
to			
capitalize			
the			
letter			
of			
any			
new			
class			
you			
cicate.			
Click			
"Finish"			
to			
cicate			
the			
the new			
the new class.			
the new class.			
the new class.			
the new class. Since			
the new class. Since the			
the new class. Since the Lab01			
the new class. Since the Lab01 class has			
the new class. Since the Lab01 class has a			
the new class. Since the Lab01 class has a main			
the new class. Since the Lab01 class has a main method,			
the new class. Since the Lab01 class has a main method, it			
the new class. Since the Lab01 class has a main method, it is			
the new class. Since the Lab01 class has a main method, it is executable by			
the new class. Since the Lab01 class has a main method, it is executable by the			
the new class. Since the Lab01 class has a main method, it is executable by the Java			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual machine			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual machine (VM).			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual machine (VM). In other			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual machine (VM). In other words,			
the new class. Since the Lab01 class has a main method, it is executable by the Java virtual machine (VM). In other words, Lab01			

is a Java application, also called an app. Next you will see		
this:		
	workspace – Java – 1710-lab0/Lab01.java – Eclipse	_ O X
File Edit Source Refactor \square \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \square \blacksquare	Navigate Se <u>a</u> rch Project Run Window Help ♂ ▼ :	🤔 💪 🧳 ▾ Quick Access 😰 😤 🐯 🎋 🕵
Package Ex E E S IT10-lab0 # (default package) Lab01.java JRE System Library [J:	Lab01.java ☆ □ Lab01.java ☆ public class Lab01 { 4 public static void main(String[] args) { 5 // TODO Auto-generated method stub 7 } 9 } 10	<pre> ■ Cutline X ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■</pre>
E	🛚 Problems 🛱 @ Javadoc 😟 Declaration 📮 Console	; ▼ □ □
c) items	
	Description	Resource Path
	Writable Smart Insert 1:1	



this: 11 TODO Autogenerated method stub You can delete the entire line. The body of the method will be empty and there should only be blank lines between the { and the } that immediate following the main method. Now cut and paste the following source code lines into the body of the main method. Use Ctrl-C and Ctrl-V

for	
copy	
and	
paste	
between	
applications.	
Suctom out	~~;~+]~("************************************
System out	<pre>printin(''''''''''''''''''''''''''''''''''''</pre>
System.out	print("
"\.	
System out	print("World"):
System out	println():
System out	<pre>println("************************************</pre>
	, , , , , , , , , , , , , , , , , , ,
-	
Save	
the	
file:	
File	
- Savo	
Gave. Of	
01	
course,	
you	
use	
the	
standard	
shortcut	
for	
"file	
save"	
(Ctrl-	
(our S).	
V	
nou	
not	
notice	
it	
n, but	
Eclinse	
is	
automatically	
compiling	
the	
program	
at	
repeated	
intervals	
in	
the	
background	
A	
file	
called	
Lab01.class	
has	
been	
created	
(although	
the	
Package	
······	

Explorer			
will			
not			
show			
it,			
in			
order			
to			
reduce			
clutter).			
You			
can			
examine			
1t			
using			
the			
File			
or			
the			
Command			
Prompt.			
Tompa			
Almost			
certainly			
the			
code			
is			
not			
formatted			
according			
to			
coding			
conventions.			
With			
Echpse,			
çan			
automatically			
reformat			
source			
code			
using			
Source			
>			
Format.			
You			
will			
use			
this			
command			
frequently,			
SO			
the			
shorteut			
which			
is			
Ctrl-			
Shift-			
 F.			
This			
step			
is			

also		
described		
in		
the		
exercise		
"Formatting		
Your		
Code".		
The		
contents		
of		
the		
editor		
window		
should		
appear		
as		
shown		
below.		
By		
now,		
you		
should		
understand		
that		
the		
body		
of		
the		
main		
method		
is		
the		
portion		
between		
the		
inner		
"{"		
and		
" } "		
(curly		
braces).		





a		
green		
circle		
analoging		
enclosing		
a		
right-		
arrow:		
0.1		
An		
even		
faster		
shortcut		
ia		
15		
to		
use		
the		
keyboard		
shortcut		
which		
is		
Ctrl-		
F11.		
When		
the		
program		
·		
15		
run,		
a		
couple		
of		
4		
unings		
will		
happen.		
First,		
in		
the		
the		
bottom		
view,		
a		
fourth		
tabled		
pane		
called		
"Console"		
will		
be		
highlighted		
The		
output		
of		
your		
арр		
is		
SCIIL		
to		
the		
console.		
It		

- should look like
- this:

	workspace – Java – 1710-lab0/Lab01.java – Eclipse	- • ×
File Edit <u>S</u> ource Refac <u>t</u> or	Navigate Se <u>a</u> rch Project Run Window Help	
🖻 🕶 🔛 🔞 🗐 🖳 🔌 🛛 🖶	} ♂ ▾ ! Ք 🗾 🐨 🖩 🔳 🖬 🖉 ▼ 🐎 ▾ 🛇 ▾ 💁 ▾ ! 🖄	
월 • ☆ • ☆ • ☆ • ⇒ •	Qu	iick Access 🛛 😰 😫 🕸 🖏
😫 Package Ex 😫 🗖 🗖	🛿 Lab01.java 🕱	🗖 🗖 🗄 Outline 🖾 🗖 🗖
E 😫 💱 🔽	1 2 public class Lab01 {	Image: Image
🕶 😂 1710-lab0	3 4⊖ public static void main(String[] args) {	▽
🔻 🌐 (default package)	5 6 System.out.println("************************************	▼ O _P Lab01
🕨 🗾 Lab01.java	7 System.out.print("Hello"); 8 System.out.print(" ");	
🕨 🛋 JRE System Library [J;	<pre>9 System.out.print("World"); 9 System.out.printlp();</pre>	
	<pre>System.out.println(); System.out.println("************************************</pre>	
	13 }	
	14 15 }	
	16	
	I Problems @ Javadoc B Declaration ☐ Console B	
	🔳 💥 🎉 🖹	
	<pre>************************************</pre>	4/bin/java (Sep 12, 2017, 4:02:06 AM
	Hello World *************	
	Writable Smart Insert 13 : 6	



XPERIMENT.
or
istance,
ou
an
hange
ny
f
le
tatements
1
le
rogram
nd
xamine
rhat
appens.
ry
iis:
ſodify
le
burce
ode
nat Ar
My
ame
your and the second
hor .
rortd
lso,
otice
nat
here
re
oth
println
nd
print
tatements.
an
Jenuly that
dull Internet