**Mini Project**

**Compiler Design**

**Prepared By: Vineet Kumar Salar**

**Guideline to complete this mini project**

VirtualBox VM setup

If you have not used a VM before, the idea is that you will be running an entire operating system inside a program (VirtualBox) that looks to the virtual guest operating system like real hardware. We provide a downloadable image of the guest OS with everything installed, so that you only need to do minimal setup to get started.

Note that since a VM contains a whole second operating system, the system requirements are high. The VM will take a bit over 512MB of ram (so you will probably want to have 2GB of total ram to keep Windows or Mac OS X running smoothly), plus a bit over 2GB of hard disk space.

Getting VirtualBox

* First, download Oracle's VirtualBox, either from the VirtualBox website at <https://www.virtualbox.org/wiki/Downloads>, or from Oracle's download page at <http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html>. Choose the version for the operating system you are running on your computer. (N.B. If you are running macOS, you will need to download version 5 of VirtualBox. Not version 6.)
* Once the download completes, run the installer to install VirtualBox. You will need administrator access on your computer to do so.

Getting our VM image

* Download our linux virtual machine image [here](https://drive.google.com/file/d/193-Tmx-DSR3dPhTstEWwWGdhji_HGqE6/view?usp=sharing). Note that this download is approximately 750MB.
* Unzip the file into a convenient directory. The unzipped files are about 2GB, so make sure you have enough disk space available. If you are a Windows XP user and have trouble opening the zip file, try WinZip (shareware) or 7-Zip (free, open source) instead of the built-in zip support in Windows.
* Once you have unzipped the VM, double-click on the file "Compilers.vbox". This should open the VM in VirtualBox.

**Using the VM**

* **The provided account is "compilers" and the password is "cool"**.
* To start the VM, click the green "Start" button. This should make the VM boot.
* To shut down the VM, click on the round "Bodhi" button at the bottom left and click System. Then choose Power Off.
* We have installed what you need to do the assignments, plus a few other programs, such as vim and emacs. If you want to install other packages, you can use the Aptitude graphical package manager (under the Bodhi menu->Applications->Preferences) or the apt-get command line tool. If you are not familiar with these, there are many tutorials online that you can find through a quick Google search.
* To get a terminal, click on the terminal icon at the bottom of the screen. This should get you to the point where you can start the assignments.

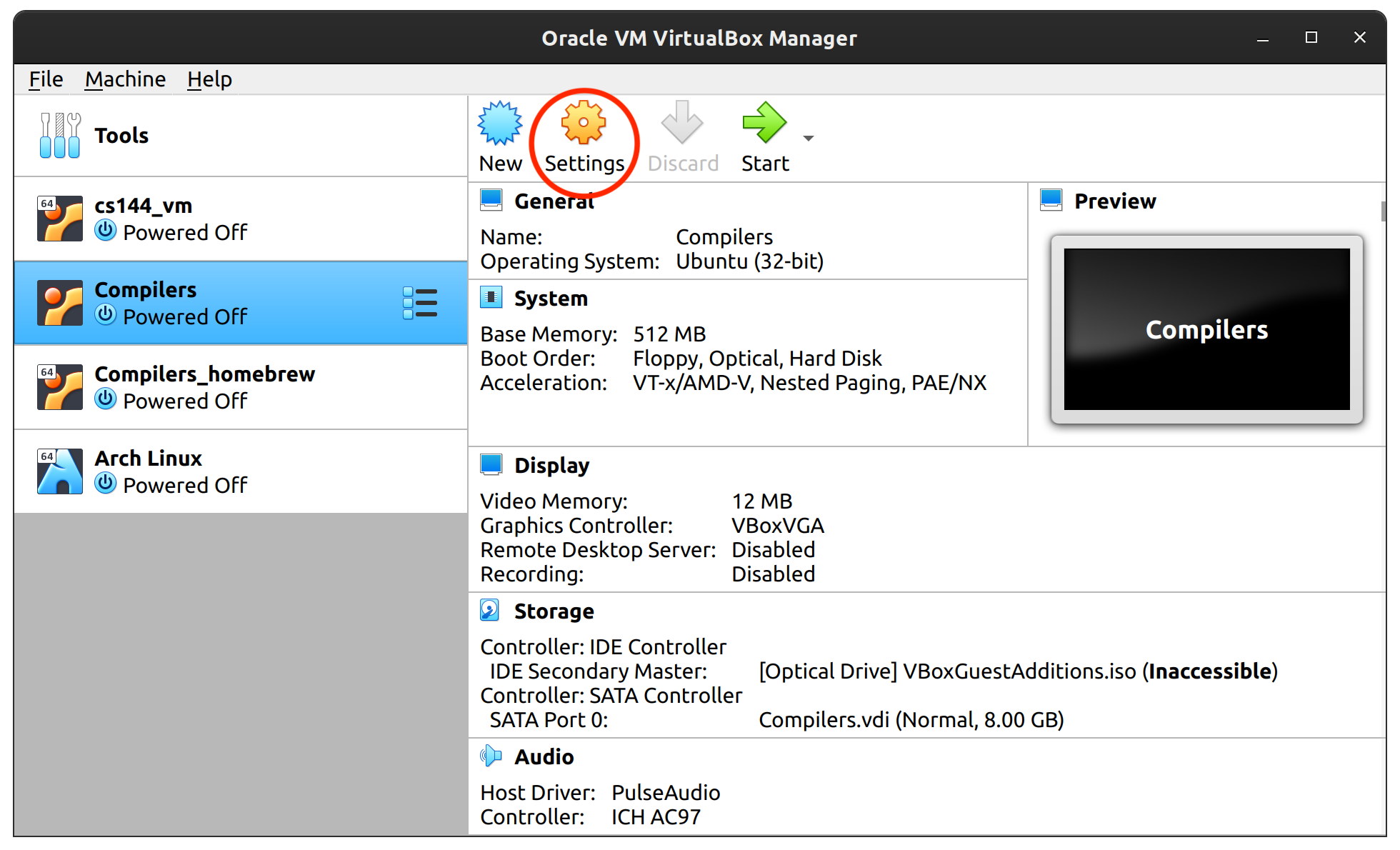
This VM is based on Bodhi Linux, which is itself based on the popular linux distribution Ubuntu. We used Bodhi Linux in order to keep the download size more manageable, since a full Ubuntu install is quite large. However, since it is based on Ubuntu, most Ubuntu software packages can be installed on Bodhi Linux as well.

If you have a problem that you cannot solve, please post in the course's Discussion Forum.  This course is unmoderated, so your best resource in this course will be the documentation, the video lectures, and fellow participants in the Discussion Forum.

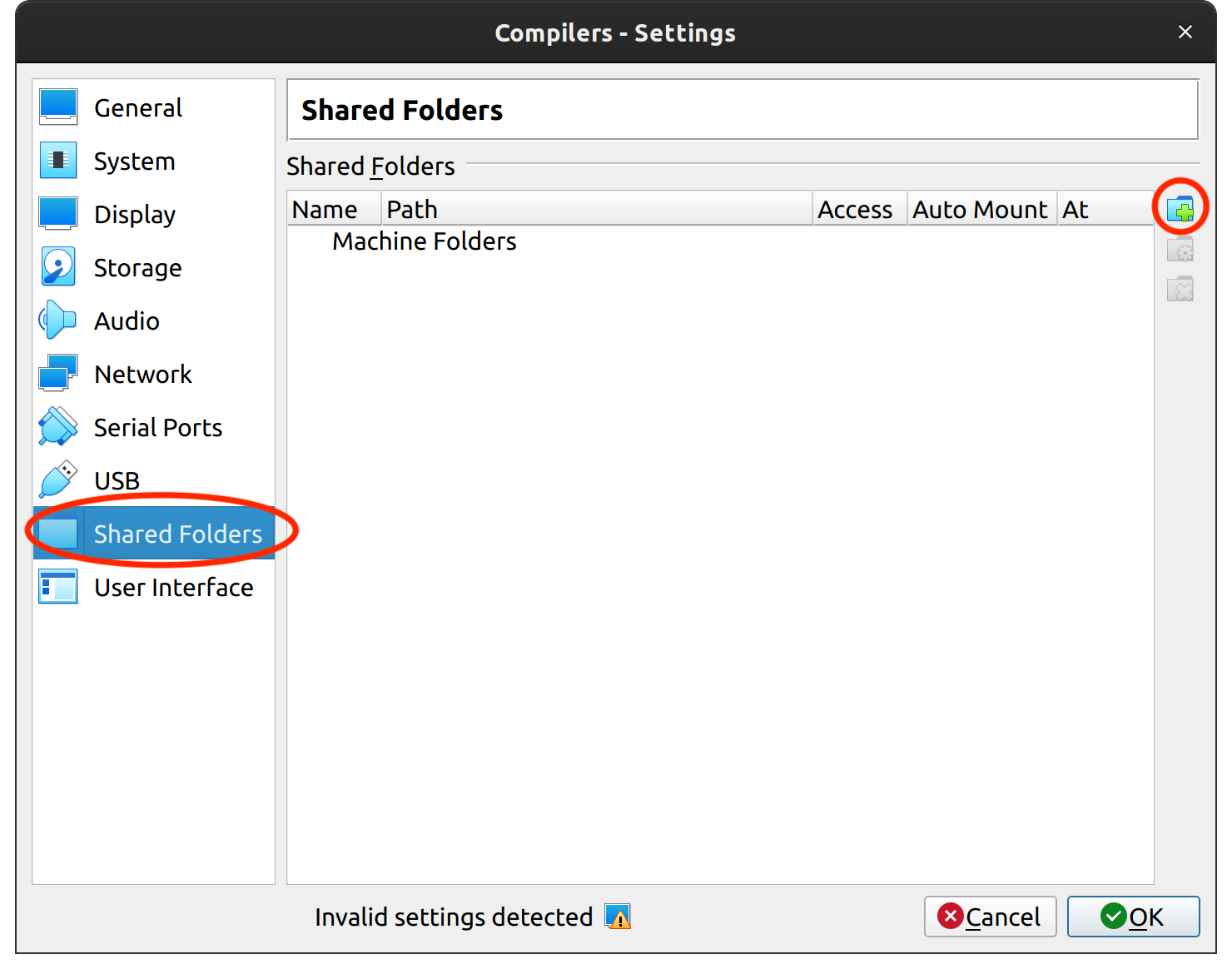
**Downloading Grading Scripts and Making a Shared Folder**

In order to complete the assignments for this course, you need to download the grading scripts for each assignment. Since this version of Bodhi doesn't support TLS, you need to set up a shared folder between your host operating system and the virtual machine. That way you can download the scripts to your host system and access them on your VM.

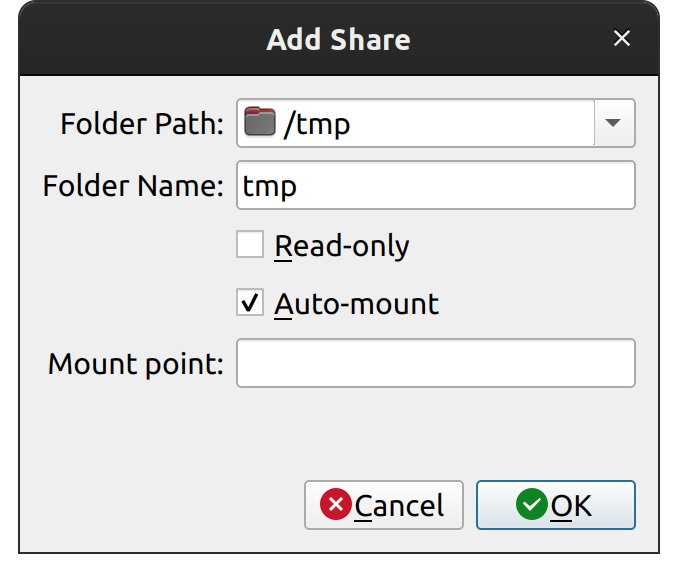
* Go the VirtualBox home screen and click on the settings icon



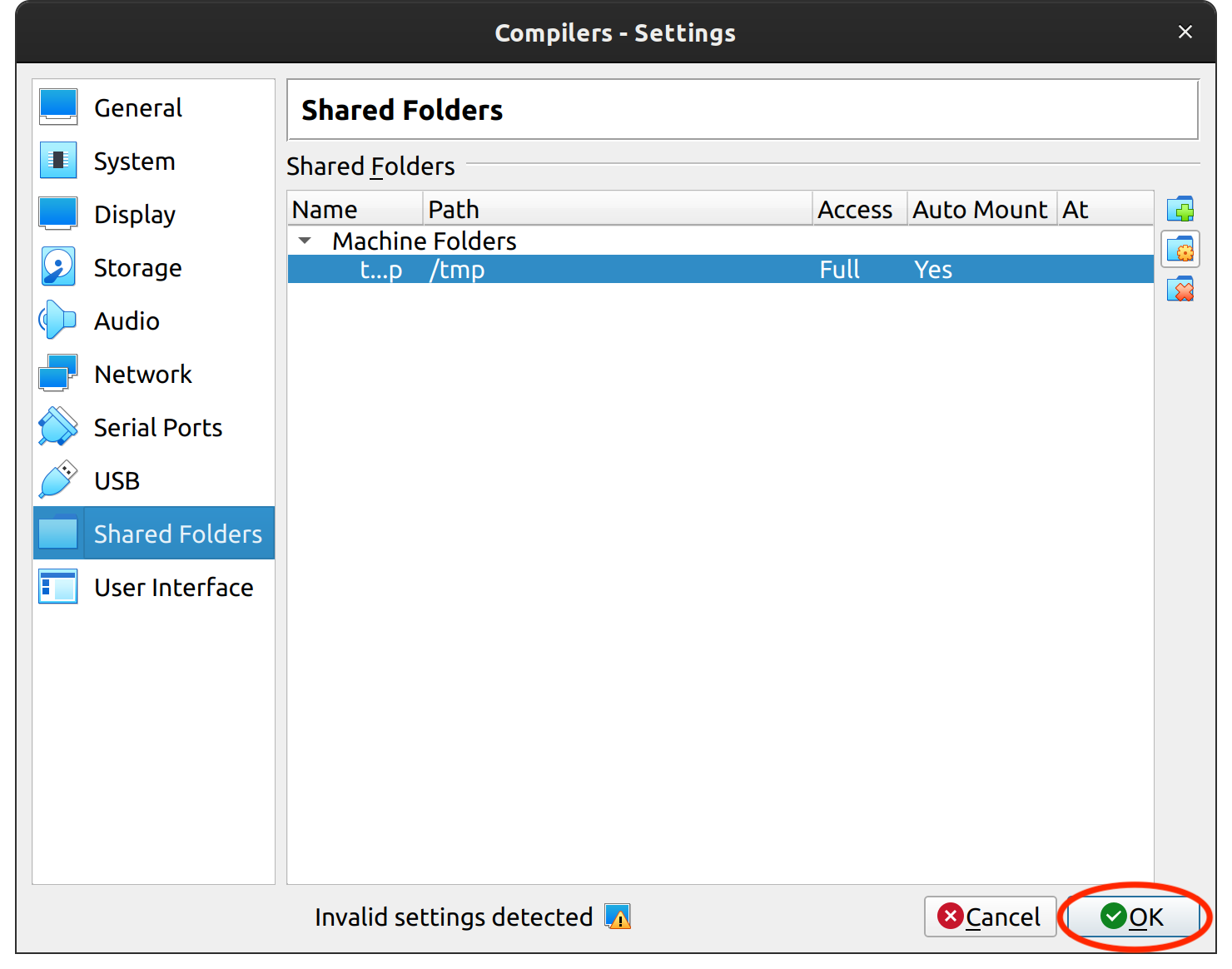
* Go to the shared folders menu and click the "add" button



* Select a path where you want your shared folder to be on the host system
* Tick auto-mount



* Click Ok



**Using the Shared Folder**

For this section, we're going to refer the name of your shared folder as <FolderName>. For instance, in the example above, <FolderName> refers to tmp.

* Auto-mounted folders are in /media/sf\_<FolderName>
* If you didn’t tick auto-mount, you can mount a shared folder with the following:

$ sudo mkdir -p /media/sf\_<FolderName>  
$ sudo mount -t vboxsf -o rw,gid=vboxsf <FolderName> /media/sf\_<FolderName>

* To make the shared folder editable, you will need to run the following command and reboot the virtual machine.

$ sudo usermod -a -G vboxsf compilers

* Now any files written to your shared folder will appear in both your virtual machine and your host machine. For ease of access, you can create a symbolic link from your shared folder into ~/cool with the following:

$ ln -s /media/sf\_<FolderName> /home/compilers/cool/<Foldername>

* To use the grading scripts, download the script for to your host machine's shared folder and run the following command, where <num> is the number of the programming assignment you want to download.

$ mv /media/sf\_<FolderName>/pa<num>-grading.pl /home/compilers/cool/assignment/PA<num>

* Now you can run the script by running the following in the assignment directory.

$ perl pa<num>-grading.pl

## Getting Started with the VM

If you wish to write a compiler for the course, you will need an  
environment in which to do so. Due to the complexity of getting all the  
needed tools installed correctly, we provide a pre-configured Linux  
system via VirtualBox VM.

VirtualBox allows you to install the VM on your own computer. Instructions on setting up VirtualBox are available in the VirtualBox VM Setup tab.

Once the VM is set up, you can compile the example programs. Once the assignments become available, you will be able to work on these inside the VM as well.

To play around with the example cool programs, make a directory and copy over one or more examples from the /usr/class/cs143/examples directory. These examples are the same as the ones posted on the website. The coolc command will run the reference compiler to generate the assembly output (.s) file, which you can run using the spim simulator. For example, to compile and run "hello\_world.cl", run in a terminal (where $ represents the prompt):

$ mkdir examples

$ cd examples

$ cp /usr/class/cs143/examples/hello\_world.cl .

$ coolc hello\_world.cl

$ spim hello\_world.s

SPIM Version 6.5 of January 4, 2003

Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).

All Rights Reserved.

See the file README for a full copyright notice.

Loaded: /usr/class/cs143/cool/lib/trap.handler

Hello, World.

COOL program successfully executed

Stats -- #instructions : 152

#reads : 27 #writes 22 #branches 28 #other 75