

CS1331-Checkstyle

A Guide to setup checkstyle for the CS1331 course at Georgia Tech. It will go over how to download and use checkstyle along with how to make aliases on Linux, Windows, or Mac.

At this point you should have Java installed and working on your computer. If not, go through the Java installation guide found [here](#)

Checkstyle is a style guide used in CS1331 to make sure that you are writing code that is readable and follows standard coding conventions. It scans through your code and lets you know of any style errors that you've made so that you can fix them. All homework that you submit should be run through checkstyle first because we take points off for checkstyle errors.

It is run through the terminal(*nix systems like Mac and Linux) or through the command prompt/powershell/git bash (Windows).

You will see the word 'directory' be used in this document frequently. A directory is just the technical term for a folder on your computer. If asked to make a directory, you just need to create a new folder.

Download Checkstyle

- Checkstyle-6.2.2.jar
- Checkstyle xml file

Where should you place these files?

We suggest that you create a folder called `cs1331` somewhere easily accessible (for e.g. `C:/cs1331` on windows or `/home/username/cs1331` on *nix).

Create a folder named `bin` in your `cs1331` folder and save the checkstyle files there.

When your next homework is assigned, you should copy the checkstyle files from the `bin` folder and paste it into your homework folder along with the `.java` files. (unless you set up an alias - more on that later)

Run Checkstyle

Checkstyle needs to be run on java source code. So, if you haven't created or written your `.java` files yet, you have to do so first.

Place the checkstyle files in the same directory as the code (`.java` files) and from the command-line, execute the following command:

```
java -jar checkstyle-6.2.2.jar -c cs1331-checkstyle.xml
```

The `java -jar` part means that I'm executing a `.jar` file. Then I give it the checkstyle file. Then I give it the 1331 specific instructions through the `.xml` file. Then I tell it what files I want it to check. `*.java` means all the java files in the current directory. Make sure that you have already navigated to the directory where your code is before you run checkstyle.

Here is what it looks like if you have some checkstyle errors (Don't worry about what these errors mean right now)

```
yash lambdas $ cp ~/gatech/cs1331/bin/checkstyle-6.2.2.jar .
yash lambdas $ cp ~/gatech/cs1331/bin/cs1331-checkstyle.xml .
yash lambdas $ ls
UsingLambdas1.class UsingLambdasReverseComparator.class UsingLambdas.class UsingLambdas.java checkstyle-6.2.2.jar cs1331-checkstyle.xml
yash lambdas $ java -jar checkstyle-6.2.2.jar -c cs1331-checkstyle.xml *.java
/home/yash/gatech/cs1331/rec/lambdas/UsingLambdas.java:16: Line is longer than 80 characters (found 90).
/home/yash/gatech/cs1331/rec/lambdas/UsingLambdas.java:63:28: '-' is not followed by whitespace.
Audit done. Errors (potential points off):
2
yash lambdas $
```

After going through my code and fixing all the errors, I run checkstyle again to make sure there are no more errors

```
yash lambdas $ java -jar checkstyle-6.2.2.jar -c cs1331-checkstyle.xml *.java
Audit done. Errors (potential points off):
0
yash lambdas $
```

We take one point off your homework for each checkstyle error, so make sure you run checkstyle before submitting!

Aliases

That is a very long command and a long process. You're probably going to have to open this guide and copy/paste it every time you want to use it. That's a little less than ideal isn't it? And who wants to remember to copy the checkstyle file into each homework directory? You're a computer science student - you're always looking for ways to do less work!

Well, we can shorten this whole process down. In fact, you can run checkstyle on any of your code in any directory by simply typing in the following command:

```
checkstyle *.java
```

Isn't that so much easier?

Here's how to do it.

Linux / Mac (*nix systems)

1. Open the terminal
2. Type `cd` and hit enter. This navigates to your home directory.
3. Type in `nano .bashrc` for Linux or `nano .bash_profile` for Mac
4. Go to the end of this file and add this line to it

```
alias checkstyle='java -jar /home/yash/cs1331/bin/checkstyle-6.2.2.jar -c /home/yash/cs1331/bin/cs1331-checkstyle.xml'
```

remember to replace the directory here with wherever your cs1331 directory is.

5. Save and close the file (Ctrl-X closes the file in nano. Then hit Y when it asks whether you want to save it, and then hit enter to confirm). Restart the terminal, and you're done!

(If it didn't work, try adding the same line in ~/.bash_profile instead of ~/.bashrc or vice-versa)

Screenshots of the process:

- Open terminal. You can see that my checkstyle alias is not set yet.

```
Welcome, yash
yash - $ checkstyle
bash: checkstyle: command not found
yash - $ nano ~/.bashrc
```

- Open ~/.bashrc. Yours will probably look different. Add the checkstyle alias into the file.

```
Activities yash@yasharch:~ Tue Jan 19, 12:15 PM
GNU nano 2.8.3 File: .bashrc Modified

#
# ~/.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return

# Welcome message
echo "Welcome, $USER"

# aliases
alias ls='ls --color=auto'
alias checkstyle='java -jar /home/yash/gatech/cs1331/bin/checkstyle-6.2.2.jar -c /home/yash/gatech/cs1331/bin/cs1331-checkstyle.xml'
alias logisim='java -jar ~/Downloads/Logisim.jar'
alias c='commandctl connect wifi 645a0427140b 475477696669 managed 1eee8021x'
alias cs='commandctl connect wifi 645a0427140b 47546774686572 managed_psk'
alias cs1331='cd ~/gatech/cs1331/'
alias cs2110='cd ~/gatech/cs2110/'
alias cs4='cd ../../../../'

#cs1331-hw5 aliases
alias compile='javac -encoding utf8 -d target/classes/ -cp src/main/java/ src/main/java/boardview/ChessFX.java'
alias runit='java -cp target/classes boardview.ChessFX'

PS1='[\u@h \W]\$ '

# Enable tab completion
source ~/.git-completion.bash

# colors
green='\[\033[0;32m\]'
blue='\[\033[0;34m\]'
purple='\[\033[0;35m\]'
white='\[\033[0;37m\]'

Get Help Write Out Where Is Cut Text Mark Set Cur Pos Prev Page First Line WhereIs Next
Exit Read File Replace Uncut Text To Spell Go To Line Next Page Last Line To Bracket
```

- Save and close the file. Restart terminal. Now checkstyle alias is set. *Note: you still have to tell checkstyle what file to check or it will complain like it did in the screenshot*

```
Welcome, yash
yash - $ checkstyle
Must specify files to process
usage: java com.puppycrawl.tools.checkstyle.Main [options] -c <config.xml>
    file...
    -a          Use both default and Javadoc XML
    -c <arg>    The check configuration file to use.
    -f <arg>    Sets the output format. (plain|xml). Defaults to plain
    -j          Use Javadoc XML
    -o <arg>    Sets the output file. Defaults to stdout
    -p <arg>    Loads the properties file
    -v          Print product version and exit
yash - $
```

Windows

There are many ways to access a terminal in windows. The ways to make aliases with the default command-prompt and with powershell are very complicated and will not be shown here. (You could Google them and figure out how to do them if you want).

Aliasing in Windows can be done easily if you use git bash as your terminal. Here's how:

1. Open git bash
2. Type the command `cd` and hit enter. This navigates to your home directory.
3. Type in `touch .bashrc` (touch will create the file)
4. Type in `vim .bashrc` (this will let you edit the file)
5. In vim, typing 'a' will let you begin editing the file
6. Type in: `alias checkstyle='java -jar /c/cs1331/bin/checkstyle-6.2.2.jar -c /c/cs1331/bin/cs1331-checkstyle.xml'`
7. Type `Esc` to stop editing then type `:wq` and hit Enter to save and quit.
8. Restart git bash and you're done!

Remember to replace the path in step 6 with the path to where your cs1331 folder is.

If it didn't let you create the file in step 3, then try the command `touch .bashrc`. - Windows will remove the dot at the end by itself.

If it didn't work, try again with `.bash_profile` instead of `.bashrc`

Screenshots of the process:

- Open git bash, navigate to home and create the file (in my case, I had to use `.bash_profile`)



```
MINGW64/c/Users/Yash
vash@YashPC MINGW64 /g
$ cd
vash@YashPC MINGW64 ~
$ touch .bash_profile
vash@YashPC MINGW64 ~
$ vim .bash_profile
```

```

MINGW64/c/Users/Yash
alias checkstyle='java -jar /g/gatech/cs1331/bin/checkstyle-6.2.2.jar'
~/.bash_profile[+] [unix] (12:56 01/12/2015)
:wq

```

- Open the file in vim and add the alias.
(screenshot doesn't show the xml part of the command, but it must be included)
- Save and close the file. Restart git bash and test whether it worked

```

Yash@YashPC MINGW64 /g/gatech/cs1331/hw/arrays/histogram/solution (master)
$ checkstyle *.java
G:\gatech\cs1331\hw\arrays\histogram\solution\CSHistogram.java:9:13: Name 'MAX_SCORE' must match pattern '^[a-z][a-zA-Z0-9]*$'.
G:\gatech\cs1331\hw\arrays\histogram\solution\CSHistogram.java:22: 'for' construct must use '{}'.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:18: 'while' construct must use '{}'.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:18: Line is longer than 80 Characters (found 104).
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:19:47: '-' is not preceded with whitespace.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:19:48: '-' is not followed by whitespace.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:20: 'if' construct must use '{}'.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:20: Line is longer than 80 characters (found 93).
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:21: 'for' construct must use '{}'.
G:\gatech\cs1331\hw\arrays\histogram\solution\GradeHistogram.java:21:72: Empty statement.
Audit done. Errors (potential points off):
10
Yash@YashPC MINGW64 /g/gatech/cs1331/hw/arrays/histogram/solution (master)
$ |

```

Now you know how to run checkstyle and how to make an alias so that it is easy to use!