

CS2030S PE1 Cheatsheet	CS2030S					
Vim Essential	Usage > a - goes from COMMAND to INSERT > Esc - goes from INSERT to COMMAND > . - repeat last command > :wq / :x - save and quit > :w - quit > CTRL W V - split window > CTRL W arrow - switch between windows > vim -O *files - open multiple files horizontally > :e filename - open different file > :terminal - open terminal > CTRL N - autocompleate					
Vim Editing	Usage > u - undo > U - undo entire line > CTRL r - redo > C - replace to end of line > ciw - replace word > cc - replace entire line > :%s/old/new/gc - replace all old with new > p - paste clipboard > 4yy - copy 4 lines > yiw / yaw - copy word > Y - copy to end of line > 4dd - delete 4 lines > diw / daw - delete word > D - delete to end of line > gg=G - fix indenting					
Vim Global	Usage > :h keyword - open help for keyword > :sav file - save file as > :clo - close current pane > :ter - open terminal window					
Vim Cursor	Usage > h - left > j - down > k - up > l - right > H - top of screen > M - middle of screen > L - bottom of screen > w - forwards to start of word > e - forwards to end of word > b - backwards to start of word > ge - backwards to end of word > 0 - start of line > \$ - end of line > gg - first line > G - last line > 4G - go to line 4 > f4 - next occurrence of 4 > F4 - previous occurrence of 4 > ; - repeat previous occurrence movement > { - previous paragraph > } - next paragraph > CTRL e - screen down one line > CTRL y - screen up one line > CTRL f - screen down one page > CTRL b - screen up one page > CTRL d - screen down ½ page > CTRL u - screen up ½ page > /pattern - search for pattern > ?pattern - search backwards for pattern > n - repeat search > N - repeat search in opposite direction					
Lab Workflow	Access the PE Hosts > Access the PE hosts either through lab sessions or have to tunnel through stu.comp.nus.edu.sg > Accept link from Luminus main module page > Use any PE from pel11 to pel20 (can't use 116 or 117) > ssh -t hafeez@stu.comp.nus.edu.sg ssh hafeez@pel13.comp.nus.edu.sg > The second username will be different for exam setting > ~cs2030s/get-labX to copy lab X to directory > ./test.sh LabX to test code > ~cs2030s/submit-labX to submit lab X, can be done multiple times > cp -i ../lab1<username>/*.java . to copy files from different lab > rm -i Lab1.java > java LabX < inputs/LabX.Y.in > rm -i .Question.md.swp to delete swp file > touch filename to create file > checkstyle in terminal to checkstyle > java -jar ~cs2030s/bin/checkstyle.jar -c ~cs2030s/bin/cs2030_checks.xml *.java > ls -a then ~cs2030s/.vimrc  Topics ?> SOC Unix Account					

	<p>Types</p> <pre>?&gt; Checked_Exception ?&gt; Unchecked_Exception</pre> <p>Usage</p> <pre>&gt; Passes an exception if a method fails, can be handled elsewhere &gt; Checks the nearest try block and evaluate the catch there &gt; Always runs the finally block</pre> <p>Disadvantages</p> <pre>- Shouldn't use for decision-making and control-flow</pre> <p>Code &lt;JV</p> <pre>import java.lang.ExceptionName JV&gt;</pre> <p>Examples &lt;JV</p> <pre>public Constructor() throws E1,E2 {     throw new ExceptionName(); }</pre>				
<b>Exception</b>	<pre>try {} catch(e) {} finally {} JV&gt;</pre>				
<b>Valid Generics</b>	<p>Examples</p> <pre>&gt;&lt;JV Pair p = new Pair(2,3) JV&gt; &gt;&lt;JV Pair&lt;Integer,Integer&gt; p = new Pair&lt;&gt;(2,3) JV&gt; &gt;&lt;JV Pair p = new Pair&lt;Integer,Integer&gt;(2,3) JV&gt; &gt;&lt;JV Pair p = new Pair&lt;&gt;(2,3) JV&gt; &gt;&lt;JV List&lt;?&gt; L = new List&lt;String&gt;() JV&gt; &gt;&lt;JV Pair&lt;?,?&gt; p = new Pair&lt;Integer,Integer&gt;() JV&gt;</pre>				
<b>Invalid Generics</b>	<p>Examples</p> <pre>&gt;&lt;JV Pair&lt;&gt; p = new Pair&lt;&gt;(2,3) JV&gt; &gt;&lt;JV Pair&lt;Integer,Integer&gt; p = new Pair&lt;?,?&gt;() JV&gt; &gt;&lt;JV List&lt;?&gt; L = new List&lt;?&gt;() JV&gt;</pre>				
<b>Producer Extends</b>	<p>Definition</p> <pre>&gt; Take information out of the object</pre> <p>Usage</p> <pre>&gt; /Parameter_Type &gt; &lt;? extends T&gt;</pre> <p>Code</p> <pre>&gt;&lt;JV static &lt;? extends T&gt; T foo(T){} JV&gt;</pre>				
<b>Consumer Super</b>	<p>Definition</p> <pre>&gt; Insert information into the object</pre> <p>Usage</p> <pre>&gt; /Return_Type &gt; &lt;? super T&gt;</pre> <p>Code</p> <pre>&gt;&lt;JV static &lt;T&gt; T foo(Array&lt;? super T&gt;){} JV&gt;</pre>				
<b>PE Workflow</b>	<p>Process</p> <pre>::Set up environment &gt; Luminus multimedia Panopto &gt; ssh -t hafeez@stu.comp.nus.edu.sg ssh hafeez@pe103.comp.nus.edu.sg &gt; password U6BxL4kV &gt; ~cs2030s/start-pe-0900 &gt; ~cs2030s/submit-pypel &gt; java -jar ~cs2030s/bin/checkstyle.jar -c ~cs2030s/bin/cs2030_checks.xml *.java</pre>				
<b>Java Equals (Generics)</b>	<p>Code &lt;JV</p> <pre>if (this == other) return true; if (other instanceof B&lt;?&gt;) {     B&lt;?&gt; b = (B&lt;?&gt;) other;     if (b.x == other.x) return true;     if (b.x == null    other.x == null) return false;     return b.x.equals(other.x); }</pre>				
<b>Java List</b>	<p>Usage</p> <pre>&gt; Stores elements in an expanding fashion &gt;&lt;JV import java.util.* JV&gt;</pre> <p>Types</p> <pre>?&gt; Java_ArrayList ?&gt; Java_LinkedList</pre> <p>Methods</p> <pre>&gt;&lt;JV List(len) JV&gt; &gt;&lt;JV add() JV&gt; &gt;&lt;JV get() JV&gt; &gt;&lt;JV addAll(list) JV&gt; &gt;&lt;JV size() JV&gt; &gt;&lt;JV isEmpty() JV&gt; &gt;&lt;JV contains(obj) JV&gt; &gt;&lt;JV remove(obj) JV&gt; &gt;&lt;JV clear() JV&gt; &gt;&lt;JV set(id,obj) JV&gt;</pre>				
<b>Java Super</b>	<p>Usage</p> <pre>&gt; Can super() to parent construct &gt; Can super.method() to use parent method &gt; Can super.field to use parent class field</pre>				