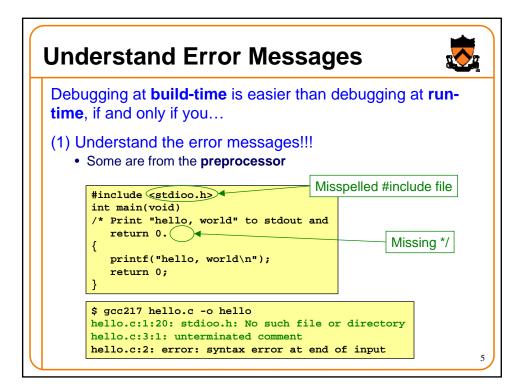
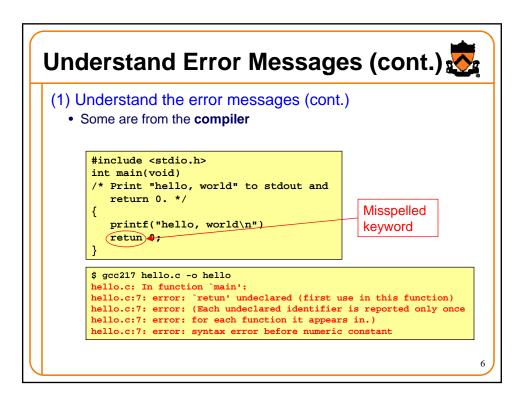
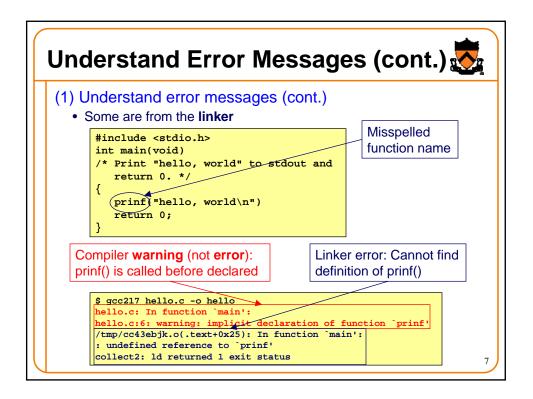
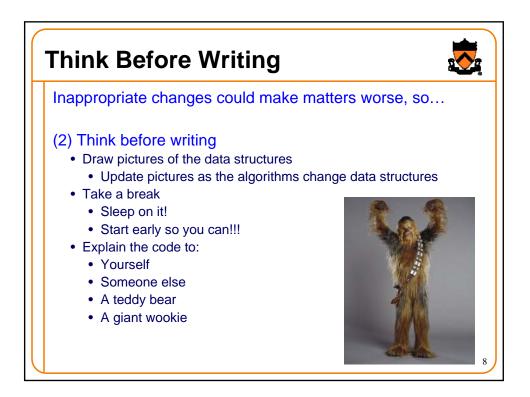


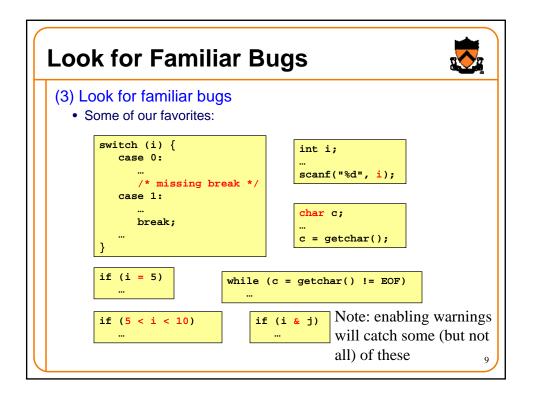
Debugging Heuristic	When Applicable
(1) Understand error messages	Build-time
(2) Think before writing	
(3) Look for familiar bugs	
(4) Divide and conquer	Run-time
(5) Add more internal tests	
(6) Display output	
(7) Use a debugger	
(8) Focus on recent changes	

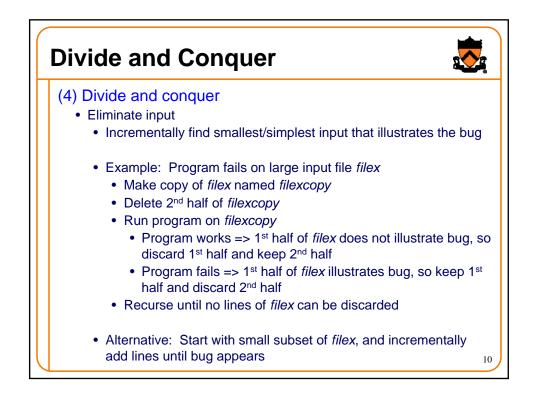


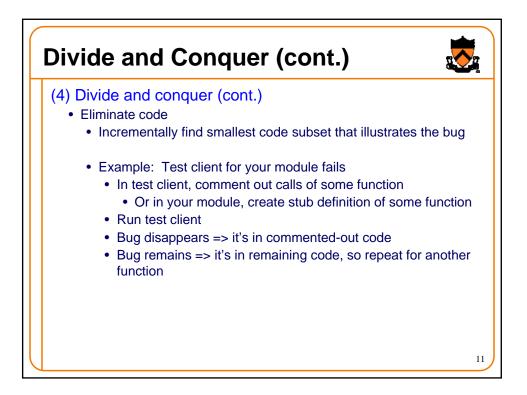


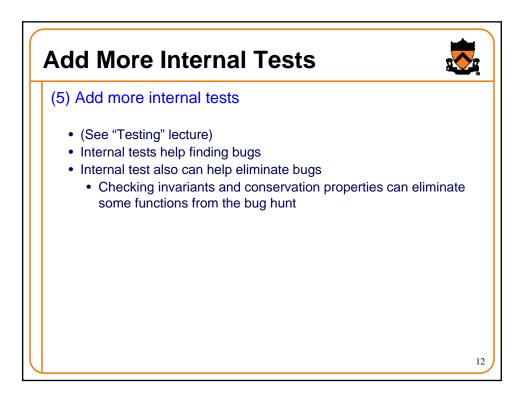


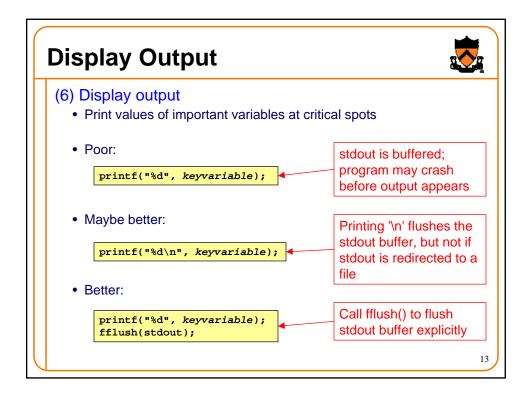


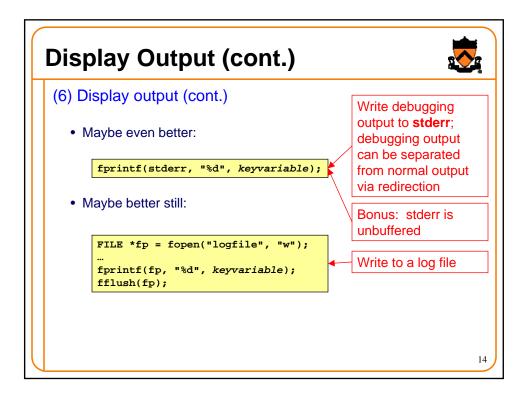


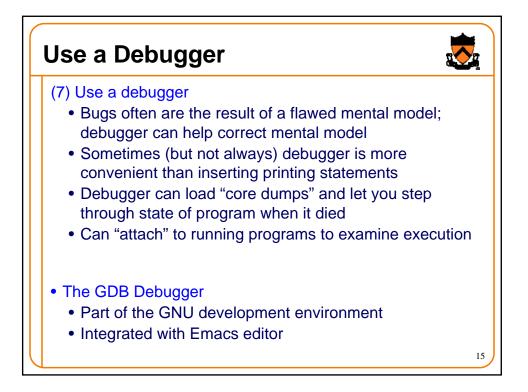


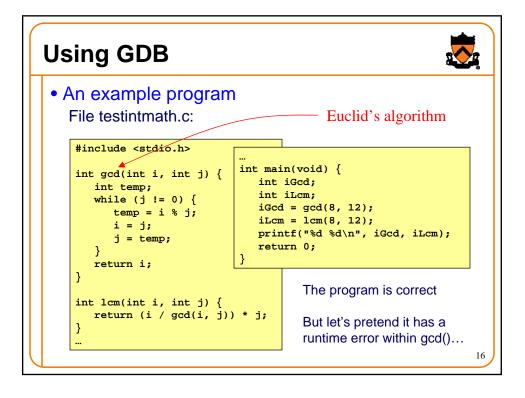


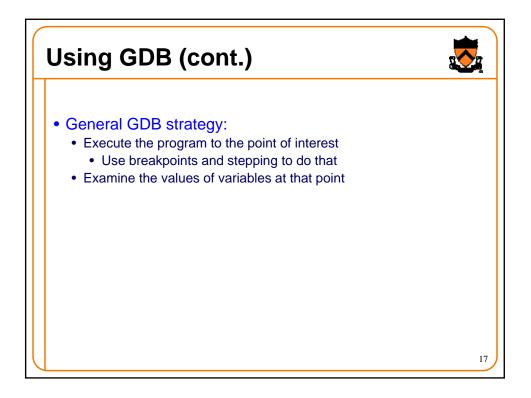


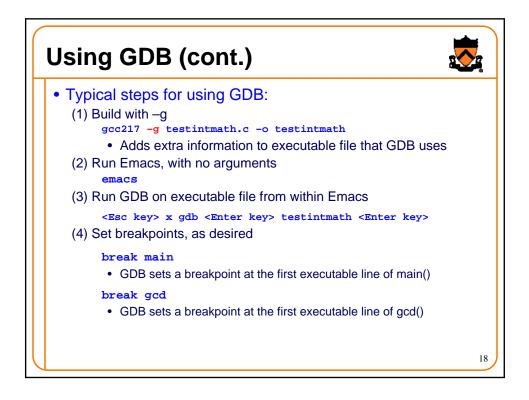


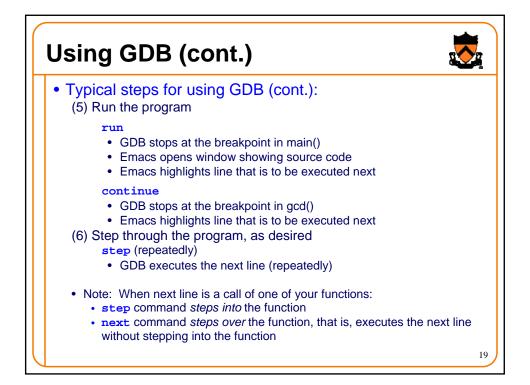


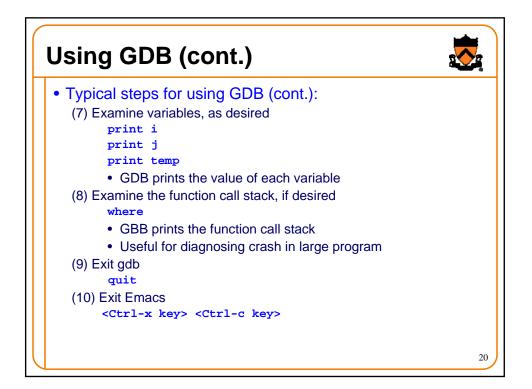


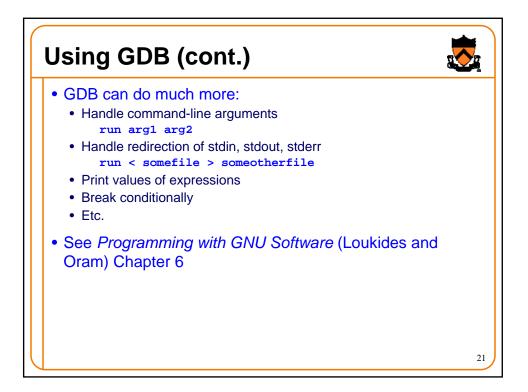


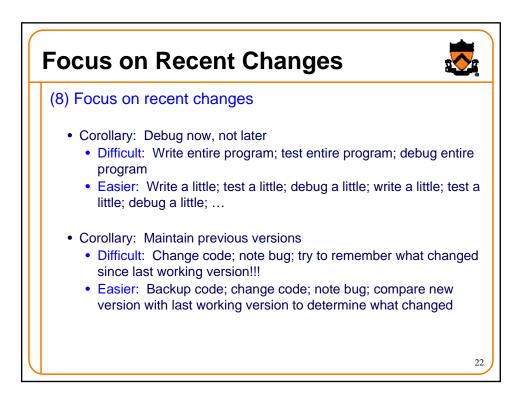


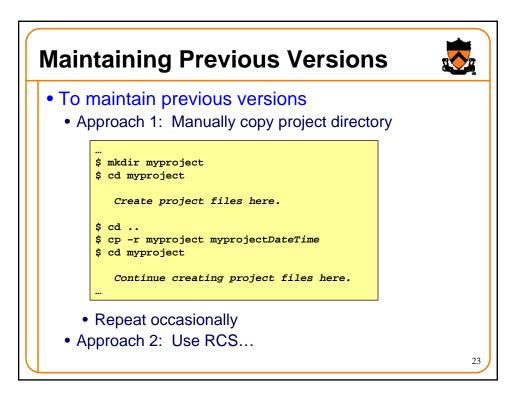


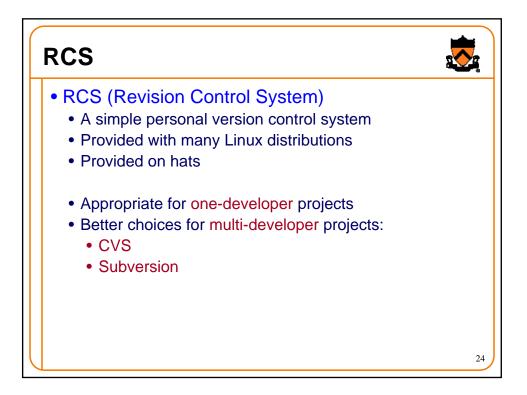


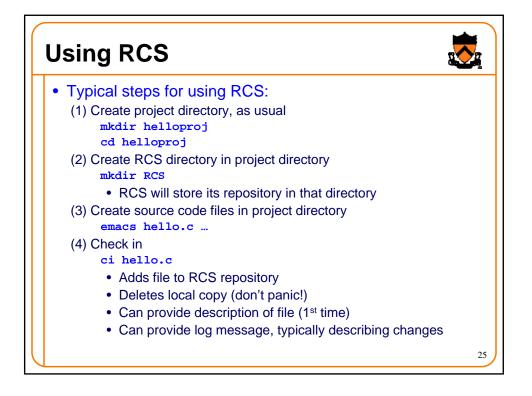


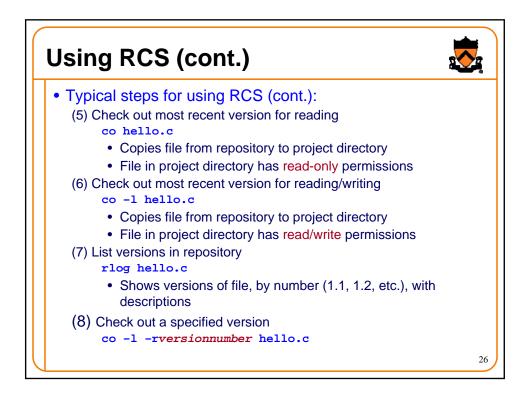


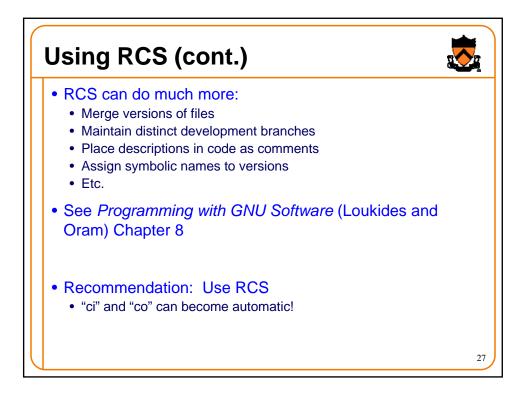




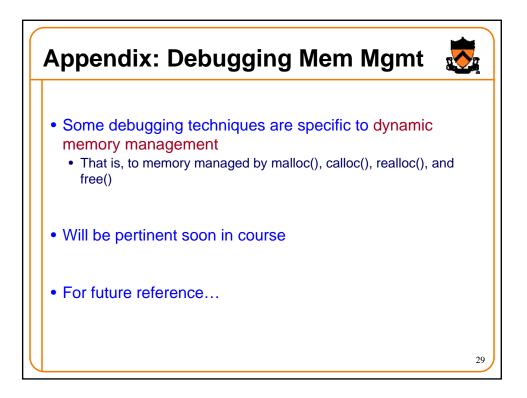


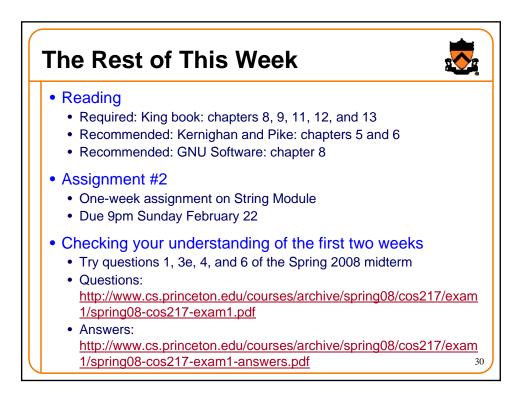


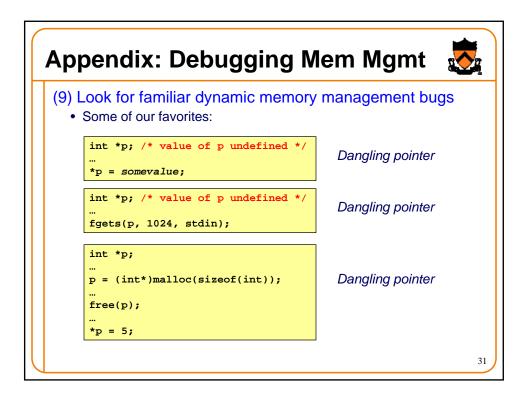


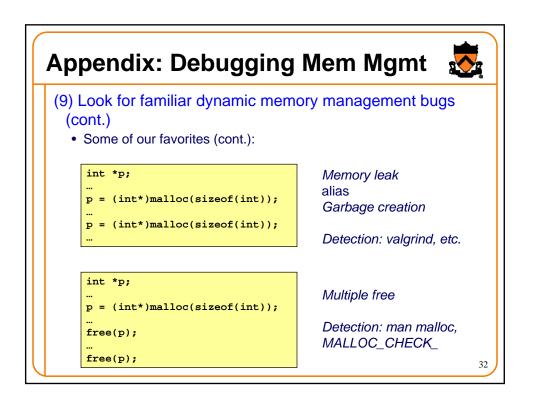


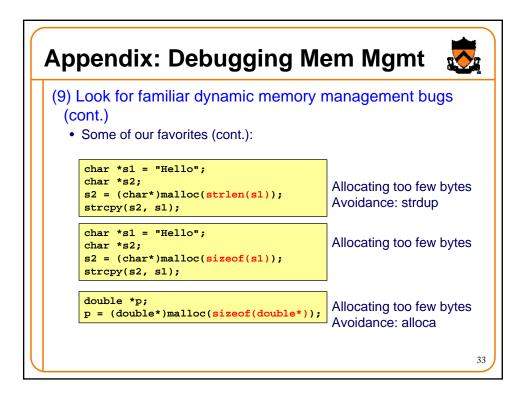
Debugging Heuristic	When Applicabl
(1) Understand error messag	ges Build-time
(2) Think before writing	
(3) Look for familiar bugs	
(4) Divide and conquer	Run-time
(5) Add more internal tests	
(6) Display output	
(7) Use a debugger *	
(8) Focus on recent changes	) ** )

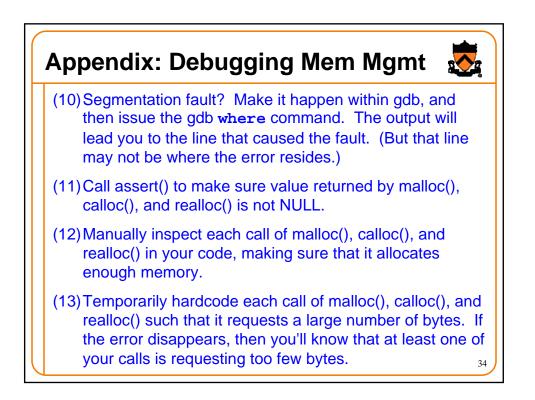












## Appendix: Debugging Mem Mgmt



- (14) Temporarily comment-out each call of free() in your code. If the error disappears, then you'll know that you're freeing memory too soon, or freeing memory that already has been freed, or freeing memory that should not be freed, etc.
- (15) Use the Meminfo tool. Programs built with gcc217m are much more sensitive to dynamic memory management errors than are programs built with gcc217. So the error might manifest itself earlier, and thereby might be easier to diagnose.

35