

First name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (10, or 11)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

H21: Due Thursday 05.29 in Lecture. Total Points: 50

Review for Final (HFJ Ch 1-17)

MAY ONLY BE TURNED IN DURING Lecture ON Thursday 05.29, or offered in person, for in person grading, during instructor or TAs office hours.

See the course syllabus at <https://foo.cs.ucsb.edu/56wiki/index.php/S13:Syllabus> for more details.

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- coloring in the first letter of your first and last name (as it would appear in Gauchospace),
- writing **either 10,11** to indicate your **discussion section** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
uemail address:	@uemail.ucsb.edu

There is no new reading assignment for this homework. Instead, this homework is your final review for the final exam.

- True/False (12 pts) (two points each)

	Question	Circle One	
2	Since the Math class has only static methods, the first thing you must do to use it is create an instance of it with <code>Math m = new Math();</code>	T	F
	(optional) if false, explain:		
3	static methods don't have access to instance variables	T	F
	(optional) if false, explain:		
4	static methods don't have access to the "this" reference	T	F
	(optional) if false, explain:		
5	static variables can be used to count the instances of a class	T	F
	(optional) if false, explain:		
6	<code>System.out</code> is a static variable	T	F
	(optional) if false, explain:		
7	Because of Polymorphism, if an object <code>o</code> is really an instance of <code>Dog</code> , e.g. <code>Object o = new Dog();</code> then you call call <code>Dog</code> methods on it (e.g. <code>o.bark();</code>).	T	F
	(optional) if false, explain:		

- True/False (28 pts) (two points each)

	Question	Circle One	
8	Because of Polymorphism, if a class Animal has a speak() method that prints "Yo!", if Dog, Cat and Human all extend Animal, and Dog overrides speak() to print "Woof" and Cat overrides speak() to print "Meow!", but Human doesn't override speak(), then if you have an ArrayList<Animal>, and call speak() on each object in the list, Dog objects will say "Woof!" and Cat objects will say "Meow!", but Human objects will say "Yo!".	T	F
	(optional) if false, explain:		
9	a try block must be followed by a catch AND a finally block	T	F
	(optional) if false, explain:		
10	if you write a method that might cause a run-time exception, you MUST wrap that risky code in a try/catch block, or add "throws" to the header of the method	T	F
	(optional) if false, explain:		
11	if you write a method that declares it can throw a compiled-checked exception in the header, you must also wrap the exception throwing code in a try/catch block	T	F
	(optional) if false, explain:		
12	a method can only throw one kind of exception	T	F
	(optional) if false, explain:		
13	a finally block is always run whether an exception is thrown or not	T	F
	(optional) if false, explain:		
14	except in the special case of an inner class defined within a static method, an inner class instance is always tied to a specific outer class instance	T	F
	(optional) if false, explain:		
15	inner classes are sometimes used for ActionListeners because they have access to the outer classes instance variables, except for the private ones.	T	F
	(optional) if false, explain:		
16	GUIs in Swing are typically made by nesting multiple JFrame objects inside a JComponent	T	F
	(optional) if false, explain:		
17	If your main JPanel has a particular layout manager (e.g. FlowLayout), then all of the children panels nested inside must have the same layout.	T	F
	(optional) if false, explain:		
18	Marking an instance variable transient means that you want it to be saved when the object is serialized.	T	F
	(optional) if false, explain:		
19	the readObject() method of an ObjectInputStream automatically returns the correct type of the object, so you can directly assign the result of the readObject() method call to variables of the appropriate type, e.g. if instr is an ObjectInputStream, then the line of code Dog d = instr.readObject() is valid as long as all the objects in the ObjectInputStream were originally instances of class Dog.	T	F
	(optional) if false, explain:		
20	Although using a BufferedWriter object is more efficient when dealing with large volumes of data, a disadvantage is that you can't force the buffered writer to send all of its data before it is full (except when finally closing with the close() method.)	T	F
	(optional) if false, explain:		
21	The main purpose of using packages in Java is to allow the creation of JAR files for distributing code.	T	F
	(optional) if false, explain:		