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 (4, 5 or 6)	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			

H09: Due Wednesday, 01.21 in Lecture

Exceptions (HFJ Ch11)
Assigned: Mon 01.12 Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,
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/W15: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 appears in Gauchospace),
- writing **either 4, 5, or 6** to indicate your **discussion section (lab)** 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:	
umail address:	@umail.ucsb.edu

If you collaborated with AT MOST one other person on this homework, write his/her name below. She/he should also have your name on his/her paper.

Reading Assignment:

- HFJ:Chapter_11, starting on p. **315** Risky Behavior
- If there are reading notes on the wiki, consult those too—sometimes they contain helpful hints.
- Note: Please note that each of the following is different—be sure you are clear on the difference, and when each should be used. (This has given students difficulty in previous quarters.)
  - Creating a new type of exception (involves creating a new class)
  - Signalling that an exception has happened (constructing an instance of an Exception class and throwing it)
  - Detecting that an exception has occurred (try/catch block)

---

(2) Exceptions in Java can be divided into two broad categories:

- One category is the kind that, if there is any chance it can happen in the code has to be "caught or declared to be thrown"
- The other category is the kind that can happen, but doesn't have to be "caught or declared to be thrown".

(a) (5 pts) Exceptions that do not have to be caught or declared to be thrown are extensions of (i.e. they are subclasses of) what class?

(b) (5 pts) What is the rationale for having some exceptions that do NOT have to be declared to be thrown or caught? (i.e. why did the designers of Java put that feature into the languge?)

(c) (5 pts) What is the rationale for having some exceptions that DO have to be declared to be thrown or caught? (i.e. why did the designers of Java put that feature into the languge?)

(3) (10 pts) In the code excerpt below, there is a test to see if the course is in the list of courses. If it is not, we want to signal that an exception has happened. Assuming there is an exception called NoSuchUCSBCourseException (i.e. that class has already been written and exists), fill in the blank with a line of Java code that indicates that this exception has happened, and the program needs to signal that.

```
public Course lookupCourse(String courseNum) throws NoSuchUCSBCourseException {
    Hashtable<String,Course> ucsbCourses = getCoursesHashTable();
    Course result = ucsbCourses.get(courseNum);
    if (result == null) {
             // your answer goes here!









    }
    return result;
}
```

(4) (10 pts) Assume that s is an object of type Student, and that there is a method `public void registerFor(String courseNum)` that might throw the NoSuchUCSBCourseException.

Write a segment of Java code that will:

- call s.registerFor(someCourse);
- print "Success" on System.out if the registration succeeded (i.e. that exception doesn't happen)
- will print "Sorry " + someCourse + "does not exist" if that exception occurs.

(5) (5 pts) Write the code that creates a new kind of Exception (for a card game) called "BadSuit" exception. It might be used, for example, if have a program that expects a character that is one of 'H','D','C','S' for Hearts, Diamonds, Clubs and Spades. BadSuit a subclass of "IllegalArgumentException".