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 (9,10,11, 12,1 or 2)	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			

H02: Due Tuesday, 01.13 in Lab

More Flow Control (Ch 3)
Assigned: Mon 01.05 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/16wiki/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 9,10,11,12,1 or 2** 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: *Read Chapter 3, pp. 110-164. (If you don't have a copy of the textbook yet, there is one on reserve at the library.)*

Then, answer the following questions. Be sure to check both sides.

Reading: *Read Chapter 2, pp.41-97. (If you don't have a copy of the textbook yet, there is one on reserve at the library.) Then, answer the following questions. Be sure to check both sides.*

2. Section 3.1 discusses boolean expressions.
   a. (4 pts) What do we mean when we say "boolean expression"?

   b. (3 pts) What do we mean when we say "short-circuit evaluation"?

   c. (8 pts) The text discusses why we have to be careful with boolean values that convert to int values with the example `if (!time > limit)` Based on what you read in that section, explain why this code can NEVER print `Freezing` no matter what value is entered for temperature by the user:

```
int temperature;
cout >> "Enter the temperature";
cin >> temperature;
if (!temperature >= 32)
  cout << "Freezing" << endl;
```

3. Section 3.2 discusses multi-way branches and the "dangling else" problem, multi-way branches, and the switch statement.
   a. (2 pts) What is the dangling else problem, and how is it solved?

   b. (2 pts) What is the purpose of a "default" clause in a switch statement?

   c. (2 pts) What happens if you forget a "break" at the bottom of a "case" in a switch statement?

   d. (2 pts) In display 3.8 on p. 136, at line 28, there is a variable called "subtotal" that has an arrow pointing to it with the label "local to the block". What does this "local to the block" business mean?

   e. (3 pts) When we talk about a "block" in C++ programming, what are we talking about?

4. Section 3.3 goes into more detail that previous sections about while loops and introduces the for loop.
   a. (6 pts) A for loop header has three parts:

```
for (part1;part2;part3) {
   doSomething;
}
```

   What are these three parts, and what does each part do?

      a. part1:

      b. part2:

      c. part3:

   b. (2 pts) Typically you should NOT put a semicolon at the end of a for loop statement. But if you do, it isn't a syntax error. And, unlike Python, C++ ignores indentation (although proper indentation should STILL BE DONE for the sake of humans reading your code.) How many times will "Hi" be printed by this code (which has a semicolon where it SHOULD NOT BE)?

```
for (int i = 0; i < 5; i++);
   cout << "Hi!" << endl;
```

   c. (2 pts) How about now---how many times will "Hi" be printed by this same code, with the semicolon removed?

```
for (int i = 0; i < 5; i++)
   cout << "Hi!" << endl;
```

5. Section 3.4 discusses the design of programs that use loops. Even though C++ may be new to you, the ideas about loops in this section should be somewhat familiar to you from previous coursework in programming, including how to use loops to accumulate a sum, how to use nested loops—even so, there may be some new and useful ideas in this section even for very experienced programmers. It is well worth reading in detail, not just for the answers to the questions below.
   a. (2 pts) When accumulating a sum, we initialize the sum variable to what value?

   b. (2 pts) What is a "sentinel" value?