CS 115 Exam 1, Fall 2009

Your name: _____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This is the only resource you may consult during this exam.
- You must show your work/explain your answers in order to receive partial credit for incorrect answers.
- All snippets of code can be assumed to be enclosed within `int main()`. You can assume that the `iostream`, `fstream`, `iomanip`, `string`, `algorithm`, and `cmath` libraries have been included at the beginning of the program.
- When you are asked to write *a snippet* of code, you may also assume that it is enclosed within `int main()` and that any necessary libraries have been included.
- When you are asked to write *a complete program*, you must write the `#include` statements, the `int main()`, etc. in your solution to receive full credit.
- A line consisting solely of "..." represents one or more unspecified C++ statements, some of which may change the values of program variables.

Grade (instructor use only)

Problem 1	
Problem 2	
Problem 3	
Problem 4	
Total	

Problem 1: 24 points.

What does each of the following snippets of code print to the screen?

(a)

```
string a = "b";
string b = "a";
cout << a << endl;
```

(b)

```
string a = "b";
string b = "a";
cout << a << b << endl;
```

(c)

```
int a = 5;
cout << ++a << endl;
```

(d)

```
for (int i=0; i < 2; i++) {
     for (int j = 0; j < 2; j++) {
          cout << j << endl;
     }
}
```

(e)

```
bool b = true;
if (b) {
     b = b && false;
}
cout << b;
```

(f)

```
int i = 4;
while (i > 0) {
     cout << i << endl;
     i -= 2;
}
```

Problem 2: 16 points.

State whether each segment of code is valid C++, meaning that it will compile. If it is valid, write "valid." If it is invalid, write "invalid" and fix it so that it will compile.

(a)

```
int a;
cin >> a;
if (sqrt(a)) {
     cout << "Moo!" << endl;
}
```

(b)

```
int i = 5;
while (i--) {
     cout << i << endl;
}
```

(c)

```
int q = 2;
if (q > 0); {
     cout << "Baa!"
}
else {
     cout << "Grrr!"
}
```

(d)

```
int i = 25;
int j = 100;
i *= j;
```

Problem 3: 30 points.

Write short snippets of code to accomplish the following tasks:

(a) *Assume:*
- An integer variable `N` has already been declared and defined.

Your task:
- Print the word "Hello!" `N` times (if `N` is negative, do not print the word at all).
- Each "Hello!" should be on a separate line.

(b) Repeatedly ask the user to enter an integer. You should stop asking when *either*
- The user enters something that could not be read, *or*
- The user enters a negative number

Other than prompting the user for an integer, your program should not print anything to the screen.

(c) *Assume*: Float variables `a`, `b`, and `c` have been declared and defined.
Your task:

- Print "Yes!" if `a`, `b`, and `c` could be the side lengths of a triangle. For this to be the case, each number must be less than the sum of the other two numbers.
- Print "No!" otherwise.

(d) *Assume*: Float variables `a` and `b` have already been declared and defined.
Your task: Print `a`, `b`, and a^b (that is, a raised to the power of b) as shown in the example below for a=2 and b=3.5:

```
2 to the 3.5th power = 11.3137
```

You do not need to worry about formatting the answer to a specific number of decimal places.

Problem 4: 30 points.

For this problem, you must write a **complete program** that does the following:

- Prompts the user to enter 100 integers. It should prompt for each integer individually.
- If the user enters an invalid input, the program should stop asking for integers and should not print anything else.
- Otherwise, after the user has entered all 100 integers, the program should print the maximum of the 100 integers.