Expressions and Logic: Drills

Order of Operations

1) Write a program that allows the user to enter 5 doubles, which we will refer to as a, b, c, d, and e below. Using as many additional variables as you feel is necessary, output the results of the following algebraic expressions below:

   a.

$$x = b^2 + c^5 + \frac{d^2}{3}$$

   b.

$$y = a + \frac{b}{2} * 8 + 90$$

   c.

$$z = 3a + 4b + 5c$$

2) This question is a very simple game. The code below will not compile. Your job is to fix all of the errors in the code. The number of times you try to compile it is your score. You want to have as low a score as possible but still have the code compile and run. The lowest score is 1, obtainable by fixing all errors before trying to compile the code.

   The purpose of this code is to compute the following algebraic expression, after the user enters a value for the variable, *x*:

$$y = x^2 + 5x + \sqrt{x+2} - \frac{1}{2}x$$

   Here is the code:

```
#include <iostrem>

use namespace std;

   main()
  {
        double x, y
        cin >> x;
        y = x^2 + 5x + sqrt(x + 2) - (1.0/2)x;
        cout << y;
  }
```

3) Predict the output of the following expressions. **Do not use the computer to figure them out**. The goal here is to be able to understand the order of operations as they apply to your code.

   a. 6 + 5/7 + 8%2 = ?
   b. 6 + 1.0*5/7 + 8%2 = ?
   c. 6 * 6 + 4 * 7 = ?
   d. 6 % 2 + 4 % 3 = ?
   e. 19 % (17 % 15) = ?

f. 19 % 17 % 15 = ?

Combined Assignment

1) Predict the output of the following code snippets, or indicate that the code will not compile. **Do not use the computer to figure them out**.

a.
```
int x = 5;
x = x;
cout << x;
```
b.
```
int x = 5;
x = x + 5;
cout << x;
```
c.
```
int x = 5;
10 = x;
cout << x;
```
d.
```
int x = 5;
x += x;
cout << x;
```
e.
```
int x = 5;
x = ++x;
cout << x;
```
f.
```
int x = 5;
x += ++x;
cout << x;
```
g.
```
int x = 5;
x += 7 + x;
cout << x;
```
h.
```
const int x = 5;
x = 10;
cout << x;
```

Logical Expressions

1) Predict the output of the following code snippets. The output will be either true (1) or false (0). All code snippets below will compile. **Do not use the computer to figure them out**.

a.
```
bool x = 5 > 6;
cout << x;
```
b.
```
bool x = 5 > 4 > 3;
cout << x;
```
c.
```
bool x = 5 < 4 < 3;
cout << x;
```
d.
```
bool x = 5 < 4 && 4 > 3;
cout << x;
```
e.
```
bool x = 5 < 4 || 4 > 3;
cout << x;
```
f.
```
bool x = 5&&4;
cout << x;
```
g.
```
bool x = 5&&(5 || 0);
cout << x;
```