IF statements, part 1: Drills

Syntax and conditions

1) Have the user enter two integers, x and y. Translate the following specifications into code. You may choose to write this question in separate files or a single file.
    a. Output a suitable message to indicate which variable is larger. How will you handle the case where the two are equal?
    b. Output a suitable message to indicate whether either of the two variables is negative. In the event that both are, the user should see two separate messages.
    c. In the event that y is larger than x, x should be set equal to y.
    d. In the event that x is larger than y, y should be set to zero.
2) Have the user enter a number that represents the total amount of money spent on food at a restaurant (what type of variable should this be?). This amount will be the pre-tax total. Compute the tax based on the following rules. Note: the tax rates apply to the **entire** amount spent.
    a. Any amounts less than $5 are not subject to any tax
    b. Any amounts between $5 and $10 are subject to a 7% tax
    c. Any amounts above $10 are subject to a 13% tax.
    d. Any amounts less than $0 are invalid and the user should be told
3) A computer can be used to evaluate a piecewise function. Have the user enter a double, x. Compute the value of *f(x)* defined as:

$$f(x) = \begin{cases} \dfrac{\sin(x)}{x}, & x \neq 0 \\ 1 \quad , & x = 0 \end{cases}$$

Output the value of *f(x)*.

4) Have the user enter the coefficients a, b, and c of a parabola ($f(x) = ax^2 + bx + c$). Using the quadratic formula and if statements, determine how many real roots the parabola has.
5) What will be output when the user enters the value 5 for the variable x?

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x < 0)
{
    x=x+1;
}
if(x >= 0)
{
    x=x-1;
}
cout << x;
```

Common coding errors

1) Your friend shows you two code segments, shown below, and claims that they are equivalent. Determine at least one test case that proves that the two code fragments do not produce the same output.

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x < 0)
{
    x=x+1;
}
if(x >= 0)
{
    x=x-1;
}
cout << x;
```

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x < 0)
{
    x=x+1;
}
else if(x >= 0)
{
    x=x-1;
}
cout << x;
```

2) Is it possible to re-write the code on the left, above, so that the two code fragments do, in fact, produce the same output for all test cases, without using the else keyword? If it is possible, re-write the code.

3) Below are two code fragments. Fill in the correct conditions in the second code fragment so that they produce the same output.

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x > 10)
{
    cout << “Big!”
}
else if(x > 5)
{
    cout << “Medium!”
}
else
{
    cout << “Small!”
}
cout << x;
```

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x > 10)
{
    cout << “Big!”;
}
if(FILL_IN_CONDITION_1!)
{
    cout << “Medium!”;
}
if(FILL_IN_CONDITION_2!)
{
    cout  << “Small!”;
}
cout << x;
```

4)

A very common error is to repeat code inside of an if statement. If statements should be used only to contain code that runs when the condition is true. Re-write the following code so that there is no repetition of code.

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x > 10)
{
    x = x + 1;
    cout << “Big!”
}
else if(x > 5)
{
    x = x + 1;
    cout << “Medium!”
}
else
{
    x = x + 1;
    cout << “Small!”
}
cout << x;
```

5) It may not seem like it, but sometimes using nested if statements is not necessary. The following code can be re-written without using a nested if statement. Determine how to do so, then test it to prove to yourself that the new code is correct.

```
int x = 0;
cout << "Enter a value for x: ";
cin >> x;
if(x > 10)
{
    if(x > 20)
    {
       cout << “REALLY big!”
    }
}
cout << x;
```