1. What is the output of the following code?

   ```
   a = 3
   b = (a != 3)
   print(b)
   ```

   A. True  
   B. False  
   C. 3  
   D. Syntax error  

   b is assigned the result of a relational operator, so it must be a boolean. Here a != 3, so b is False.

2. What is the output of the following code?

   ```
   a = 3
   b = (a != 3)
   print(b)
   ```

   A. True  
   B. False  
   C. 3  
   D. Syntax error  

   b is assigned the result of a relational operator, so it must be a boolean. Here a == 3, so b is True.

3. What is the output of the following code?

   ```
   a = 3
   b = (a == 3)
   print(b)
   ```

   A. True  
   B. False  
   C. 3  
   D. Syntax error  

4. What is the output of the following code?

   ```
   a = 3
   b = (a == 3)
   print(b)
   ```

   A. True  
   B. False  
   C. 3  
   D. Syntax error  

   b is assigned the result of a relational operator, so it must be a boolean. Here a == 3, so b is True.
5. I would like an expression that evaluates to True exactly when at least one of the following two conditions is true:
(1) a and b are equal,
(2) when a has value 5.

Which of these expressions does that?

A. \(a == b == 5\)
B. \((a == b) \text{ or } (a == 5)\)
C. \((a == b) \text{ and } (a == 5)\)
D. \(a == (b == 5)\)

6. I would like an expression that evaluates to True exactly when at least one of the following two conditions is true:
(1) a and b are equal,
(2) when a has value 5.

Which of these expressions does that?

A. \(a == b == 5\)
B. \((a == b) \text{ or } (a == 5)\)
C. \((a == b) \text{ and } (a == 5)\)
D. \(a == (b == 5)\)

Based on “at least one of” we will need an OR operator, that is \((a == b) \text{ or } (a == 5)\). Recall that chaining introduces an AND, so option A is \((a == b) \text{ and } (b == 5)\), while option D will compare a to boolean produced by b==5. This is valid but not good Python, we should avoid comparing equality of different types. It is however OK to compare equality of numeric types like integers and floats.