Name Landon Moon ID:1001906270

Scanner in=new Scanner(System.in);

int [] bowls=new int[3];

System.out.println("Enter the number of candies in bowl 1: ");

bowls[0]=in.nextInt();

if(bowls[0]<5) {

System.out.println("Exiting.");   
}

else {

System.out.println("Enter the number of candies in bowl 2: ");

bowls[1]=in.nextInt();

if(bowls[1]==bowls[0] && 100<bowls[1]) {

System.out.println("Lots of candies!");

}

else {

System.out.println("Not a lot!");

}   
   
System.out.println("Enter the number of candies in bowl 3: ");

bowls[2]=in.nextInt();

}

1. Both Lots of candy! and Exiting. could be printed to screen since they are both in if statements.
   1. False. If exiting is printed, then the else is never run.
2. The purpose of the line bowls[1]=in.nextInt(); is to output to a value to screen.
   1. False, in.nextInt() returns the next integer.
3. The same number of relational operators as logical operators are used in this program.
   1. False, there are 3 (<, ==, <) relational operators and 1 (&&) logical.
4. If the first user input into the program is 3, Lots of candies! can possibly be output to screen.
   1. False, 3<5 so Exiting is printed and the else is not run.
5. There are 3 lines of code saving a method return value.
   1. True, all being in.nextInt(). New Scanner is a constructor/object declaration, new int[] is a array declaration
6. Exiting. will be output to screen if the element at index one (of the bowls array) is less than 5.
   1. False, index one is the second index since arrays start at 0. Bowls[0]<5 is testing the first index or index 0.
7. Enter the number of candies in bowl 3: will definitely be printed to screen.
   1. False, if bowls[0]<5 is true then the else statement is never run.
8. If Not a lot! is printed to screen, we can say for sure what the value of bowls[0] is.
   1. False, we know its not less than 5.
9. boolean b=( bowls[1]==bowls[0]); would be a valid line of code.
   1. True, a boolean is a primitive type (being back to java is nice instead of c) and setting it equal to a boolean is valid.
10. bowls[3]=in.nextInt(); would be a valid line of code since we have 3 elements in our array.
    1. True, the line is valid but would give an outofbounds exception every time, exceptions can be dealt with unlike errors.