1. The difference between the literals 4, 4.0,’4’, and “4” would be that 4 is an int, 4.0 is a double, ‘4’ is not a literal, and “4” is a string literal.
2. System.out.println(“I gotta be me!”); The System command tells the computer that it has to get ready to do an action. The .out tells it that it’s end goal is to output something. The println means to print something out and enter a new line afterward. The parentheses are for syntax. The double quotes are to indicate that a string is going to be outputted. The “I gotta be me!” is the string that will be printed by the system. The semicolon is an indicator to the system to let it know that this line of code is finished.
3. The output of the code fragment would be
   1. Here we go! 12345
   2. Test this if you are not sure. Another.
   3. All Done.
      1. The output is this because the prints do not insert a new line and hence it keeps printing on the same line. The println does put a new line.

1. The program statement System.out.println("To be or not to be, that

is the question.");

* + - 1. Is wrong because there is no concatenation that puts the entire sentence together. The way to fix this is to either put it all on one line or put a concatenation.
      2. System.out.println("To be or not to be, that"
      3. + "is the question");

1. Output = “50 plus 25 is 5025”
2. Output: He thrusts his fists
   * 1. against the post
   1. And still insists
      1. He sees the ghost
3. 15
4. 2.08
5. 1
6. 4 different program statements that increment the value of an integer variable total
   1. Count = Count+1;
   2. Count = Count++;
   3. Count+=2;
   4. Count = Count\*2;
7. Results
   1. iResult = 5
   2. fResult = 5.0
   3. iResult = 3
   4. fResult = 3.4
   5. fResult = 3.4
   6. fResult = 1.33
   7. iResult = 0.625
   8. fResult = 0.625
   9. fResult = 0.625
   10. fResult = 0.625
   11. iResult = 3
   12. fResult = 3
   13. fResult = 0
   14. iResult = 2
   15. iResult = 6
   16. iResult = 17
   17. iResult = 0
8. Operator order
   1. a (1) - b (2) - c (3) - d (4)
   2. a (1) - b (2) + c (3) - d (4)
   3. a (3) + b (1) / c (2) + d (4)
   4. a (4) + b (1) / c (2) \* d (3)
   5. a (1) / b (2) \* c (3) \* d (4)
   6. a (1) % b (2) / c (3) \* d (4)
   7. a (1) % b (2) % c (3) % d (4)
   8. a (3) - (b (1) - c (2)) - d (4)
   9. (a (3) - (b (1) - c (2)) - d (4))
   10. a (4) - ((b(1) - c (2)) - d (3))
   11. a % (3) (b (1) % c (2)) \* d (4) \* e (5)
   12. a (4) + (b (1) - c (2)) \* d (3) - e (5)
   13. (a (1) + b (2)) \* c (3) + d (5) \* e (4)
   14. (a (1) + b (2)) \* (5) (c (3) / d (4)) % e (6)
9. The role of the web in translating and executing some Java programs is that the Java applets are moved across the web. The applet is then compiled into Java bytecode and the Web takes that bytecode, interprets it using a Java interpreter, and executes it.
10. The traditional coordinate system starts from the bottom left corner with x increasing upward and y increasing rightward. The Java coordinate system starts from the top left corner with x increasing rightward and y increasing downward.
11. 8 bits for 3 values. 400 x 250 pixels = 100,000. 100,000 pixels x 3 colors x 8 bits = 2,400,000 bits.
12. page.drawLine(20, 30, 50, 60);
13. page.drawRect(10, 15, 35, 70);
14. page.drawOval(30, 30, 40, 40);
15. The eyes have a diameter of 5. Because the first value is the upper-left corner of the eye and not the center, then the first eye has a center at -7.5 units from mid and the second eye has a center at 7.5 units from the mid.