Java Review Homework: 1, 5, 7, 8

**Question 1:**

class BadNews

/\* a really terrible mess **🡨 1. Single line comment using multiple line comment opening without being closed properly, meaning the rest of the code will be ignored when by Java.**

{

public static void main (string[] args) 🡨 **2. ‘spring’ should have a capital ‘S’.**

{

int i = 34.0, j = 2; **🡨 3. Wrong data type assigned to variable ‘i’.**

System.out.println(‘Values are ’, i); **🡨 4. Single quotations are being used., 5. “i” is concatenated incorrectly.**

System.out.println(j);

}

}

Rewritten:

class BadNews

// a really terrible mess

{

public static void main(String[] args)

{

double i = 34.0;

int j = 2;

System.***out***.println("Values are " + i);

System.***out***.println(j);

}

}

**Question 5:**

MEMORY AND OUTPUT TRACING

Memory:

myNum = 5, 6

yourNum = 4.5, 7.5, 5.5, 0.5

Output:

5

11.0

6

0.5

**Question 7:**

a. long to double: automatic – significant digits may be lost

b. byte to char: explicit

c. float to double: automatic

d. byte to long: automatic

e. short to char: explicit

f. double to long: explicit

**Question 8:**

Using the following declarations, rewrite the statements to include the

appropriate type casting, most accurate rounding where necessary. If type

casting is not necessary, explain why.

int j = 5;

double k = 1.6;

int y;

double z;

1. y = j \* k; 🡪 y = (int)j\*k;
2. z = j \* k; 🡪 Cast not needed: the product of ‘j\*k’ is a double and it’s being assigned to a double.
3. z = k \* k; 🡪 Cast not needed: the product of ‘j\*j’ is a double and is being assigned to a double.
4. k = j; 🡪 Cast not needed: ‘j’ is an integer ; the less precise data type is being converted to a higher precision data type.
5. y = j + 3; 🡪 Cast not needed: the sum of ‘j+3’ is an integer and is being assigned to an integer as well.