Replace Temp with Query Bad Example

\*\*\*Assume all code is encapsulated into classes, I forgot to do this and ran out of time.

This method will generate two random doubles, print their product, then print the product of the minimum value squared, then return the square of the minimum product.

public void printTwoRandomDoubleProductsAndReturnMinProduct()

{

double x = Math.Random();

double y = Math.Random();

//Calculate product and print

double randProd = x \* y;

System.out.println("Our random product is " + randProd);

x = Math.min(x,y);

randProd = x \* x;

System.out.println("Our minimum random product is " + randProd);

return randProd;

}

One would think that this is appropriate to refactor randProd into a new method x \* y, as so:

public double printTwoRandomDoubleProductsAndReturnMinProduct()

{

double x = Math.Random();

double y = Math.Random();

System.out.println("Our random product is " + getRandProd());

x = Math.min(x,y);

System.out.println("Our minimum random product is " + getRandProd());

return getRandProd();

}

public double getRandProd()

{

return x \* y;

}

However, of course, this new printTwoRandomDoubleProducts() method will no longer have the same result. Of course, one should realize why this refactoring would be erroneous; we can possibly modify x in between calls to getRandProd() without changing y appropriately. So, this is an example of a possible side-effect of the Replace Temp with Query refactoring.