Homework\_4 Submission

**Question 1**: What is the output of the following Java program? Explain in terms of how parameters are passed in Java.

Output: (x,y) = java.awt.Point[x=1,y=1] p = java.awt.Point[x=2,y=2] q = java.awt.Point[x=1,y=1]

**Answer**: Java uses a pass-by-value form of parameters. (it can also be called a pass-by-copy-of-the-variable-value)

Integers x and y get created and initialized to 1.

Points p and q get created and initialized to (x,y) which is (1,1)

doubleScale is called and the parameters x,y,p,q are passed which are (1,1,(1,1),(1,1))

x and y are multiplied by 2, and p.x , p.y are multiplied by 2

q is created and initialized with (x,y) = (x=2, y=2)

this finishes the method go back to main

main prints : ("(x,y) = " + new Point(x, y) + " p = " + p + " q = " + q)

which outputs: (x,y) = java.awt.Point[x=1,y=1] p = java.awt.Point[x=2,y=2] q = java.awt.Point[x=1,y=1

+ new Point(x,y) is creating a new point using the original x and y which are still 1

P is the point created a in main then doubled in doubleScale so it outputs java.awt.Point[x=2,y=2]

Q is a point created in main, then re-created as a new object in doubleScale. And nothing is done with it. The Q which is called in the println is the q in main because the one in doubleScale doesn’t go back to main.

**Question 1b**: Suppose a similar program were written in C# in which all the parameters were ref parameters. What would the output of that program be?

Answer: with the references being passed instead of the values.

Output: : (x,y) = java.awt.Point[x=2,y=2] p = java.awt.Point[x=2,y=2] q = java.awt.Point[x=2,y=2]

I think the original x and y will get updated to 2 then when a new point is created within the printline it wil be (2,2), p’s x and y were updated to (2,2) but q’s is re initialized but after x and y are each 2 so now its (2,2)

**Question 2:** 2. Examine the following C++ program, in which a IntList class is defined that contains an overloaded [] operator. What is the output of this program?

**Output**:

0

1

**Question 2b**: Notice that the overloaded [] operator returns a reference. Change the [] operator to return a value instead and explain what happens in that case. Explain why the ability to return by reference is essential to be able to implement such an operator. Can anything similar be done in Java?

**Answer**: Off of not knowing much about C++ I’m going off of the research I’ve done.

change int& operator[] -> int operator[] (removing the &)

Without returning the reference I believe it is trying to do something like (list[0] = 1;) which it takes as 0=1 which isn’t true. And it doesn’t know what to do with it.

I’m not sure anything like that can be done in java. It would seem not sense everything is a “pass-by-copy-of-the-variable-value” and your only changing a copy of the variable value. You “would never have this problem” as it were.