For this project I decided to look at my first major Java programming assignment.  
It is a program that simulates an ATM.

The basic idea is that you input an amount of money and it gives you the optimal breakup of that amount using the least number of bills possible.

* Are you able to tell what you were working on with your old code? Why or why not?

Yes, I am able to tell what I am working on with the code. One problem which might make understanding the code difficult is that I did not include the problem prompt as a comment. That might have been a good idea to make the code clearer. I do think the code is pretty self-explanatory but the actual question would have helped.

* Do you know why you implemented your solution a certain way?

As for why I implemented the solution I did I think it was because it seemed to be the most obvious solution. Also, at some parts I can see I did some things because ignoring them would lead to errors. For example, floating points can lead to problems with certain numbers. Like 0.3 for example. While I do not think I did the most optimal solution I do think my solution was workable.

* How long did it take you? What resources did you use?

I do not remember how long the project took me as it was about 2 years ago. As for my resources I would guess that I used my textbook as well as the internet to look up any questions that I might have had.

* Issues with my old code

When looking at my old code one of the main issues that stands out to me is that I did not make any functions. I can understand why I might not have done so since I was doing each transaction only once. Also, there is another reason I might have not done so because even though the Euro and Dollar break up code is extremely similar there is some difference in the matter of printing. That means that the code is not as universalized as it could be. Another possible reason is that I simply did not know how to make functions in Java yet.

Another issue is that I did not make use of loops to make the program shorter. I could have stored the denominations in an array and used that to perform the calculations.

Also, I could have had a condition to break out when the amount reached 0 instead of doing unneeded actions.

Also, my code lacks any error checking. To be honest I do not really see this as an issue though. That is because usually error checking is a pretty time-consuming process as there are so many wrong possibilities to account for. Not to mention I once again probably was unaware of how to do this in Java.

* What do you know now that you didn’t know before?

One thing that I learned that I didn’t know before was the different ways of handling the floating-point error. All computer languages have this problem because it is a fault in binary language but there are some languages better at handling it.

* What has changed about your technique when approaching programming projects?

One thing that I changed about my technique since then is probably my thought process to programming. The reason I say this is because after my first Java class for my second one I had Professor Ziegler. I think he was an amazing professor and gave my coding a lot of direction and structure that it was lacking before.

* What was the "problem" you were working on?

I was working on an ATM change machine program.

* + Which programming language(s) did you use to solve the problem(s)?

I used Java as this was my intro to programming class.

* + Why did you choose a particular language? *(There is no right or wrong answer here, choosing an implementation is an art form)*

I had no choice here since it was one of my first programming classes.

* Evaluating the choices you made...
  + Was there any difficulty or ease with using that language for solving the problem?

What I like about Java is that because it is statically typed it makes it very clear what you are doing. There was a bit of an issue regarding handling the floating points though. I do remember struggling on this because we did not discuss how to fix it yet in class.

* + What was your approach to writing that code? Try to think about how you started the coding process.... such as did you first think of variables needed or did you break up the data flow into the functions? Did you work by trial and error or did you plan/pseudocode prior to beginning to code?

When writing the code, I think I just stuck with the first solution that popped into my head. That is because the code is very linear and simple in nature. That does not make it wrong it just means it is not as optimal.

* Next steps, future...
  + If you had to solve the problem again, would you choose the same language or a different language?

Solving the code again I used the Python programming language. The reason is that Python has access to the Decimal library which helps with floating point errors. Also, I think Python is a really good language for experimentation and has some features of list traversal that Java does not which makes the code a little easier to understand and implement.

* + Would you organize the code differently? How so?

Honestly if I was given this problem again, I am not sure how I would solve it. That is because I do not know if I would remember having solved it before or not.

If I do not remember solving this problem before I would probably go with the more basic solution that I did the first time around.

If I do remember I would probably try to optimize and improve the code.

That is because I would be looking at the problem with prior knowledge.

* Critique of my old code

All in all, I would say my old code is decent and if anything is a good prototype. Figuring out how to optimize the code can be done after a working model has been made.

* Possible insights

One thing this code makes me wonder is which code is better written my old code or my new code.

That is because while my old code is not as fancy it was written from scratch.

The new code while being more optimal required knowledge of my old solution.

In some ways I think I would still probably go with the most obvious solution and then try to optimize it.

Also, I feel that creating code and optimizing code are 2 completely different thought processes. When creating code, you try to be safer and more obvious. When optimizing code, you try to remove redundancies and look for patterns.

* Further Improvement

Looking at my new code I think I made some substantial improvements. I was able to cut down significantly on the number of lines of code and even add in new features. For example, I also added in how to break up the coins which is absent from the original.

One thing I noticed though is that in making the code sleeker it loses out on certain specificity.

For example, I cannot refer to the coins by their names like pennies like I originally wanted.

Also, the code still has no error detection so that could be worked on.

Also, another way to improve the code might be to use dictionaries. I say this because then I can relate the name of the coins and the values so I can bring in more specificity into the code.

This shows how I can optimize my code even more and it might be a fun side project.

This assignment also made me think of other assignments I have done where I try to repurpose old code instead of building from scratch.