	Scenarios (pink is Beth isn't sure she agrees, green is a "good one")
NOT OK	The code has a bug that is a **fundamental part of the program which requires changing multiple lines of code**. It is not ok to ask another CSE8B student to help you figure out what is the right way to write the code.
NOT OK	A scenario that would not be okay would be if the debugging **involves errors in most parts of the program** so that none of the program is executing. This is because then a peer reviewer would have to go through and understand all parts of the students program and would see exactly how they did their work. I think that would make it too easy and tempting for the reviewer to copy if they hadn't started
NOT OK	We get an assignment and two people decide to discuss it before they start working. One person is lost while the other is fairly confident with the program, and by the end of the discussion, **the confident person has shared all their plans for how to do the program and the other person now has all the same ideas without trying to figure it out on their own.**
NOT OK	I think one reading other classmates code for debugging is "ok" as long as he/she does not physically type on classmate's code or give exact code that he/she did. For example, if a guy name Peter gets a run time error in while loop and asks classmate to debug it. Classmate looks at the code for a while and sees that way Peter set up condition for while loop is improper. Since it was a small mistake (perhaps he used "<" instead of "<=") classmate just fix it by typing in the code, and Peter compiles it and sees that it works fine. This crosses the line, **because Peter would not learn what he did wrong in the code and understand how that simple change fixed the error.** Instead, if classmate explained him that while loop has to stop when variable is 8 and code is <=9 which would stop when variable gets to 9, Peter would understand his mistake and next time he gets to same problem, he would be able to fix the error by himself.
NOT OK	**Student A and Student B both work together to do the PSA assignment as if it was still pair programming. However, before each student submits it, they change the names of their variables and do other slight altercations to prevent from getting caught submitting the same code.**
NOT OK	I have finished my programming assignment in the lab and have an adrenaline rush, as well as being extremely happy. My best friend in the class, who has not finished the assignment yet, *pokes me on the shoulder and sniffles.* She tells me that she is still stuck on figuring out the algorithm for how to finish the project. I show her the code on my screen, eager to prove that I know how to do the assignment, while her lightening fingers take it all down.
NOT OK	**Combining bits and peices of code to create a finished peice. This includes anything originally created by the other student**
NOT OK	It's not OK for the person helping you to know what the error is and then just fixing the entire thing, even if it's just a missing semi-colon or something. **They should only tell you (more like hint at) what's wrong.**
NOT OK	Asking someone else for help debugging when you have **not exhausted your own resources of finding the bug**
NOT OK	Student 1 asks Student 2 for debugging help. Time is short, so Student 1 offers to **email their code to Student 2,** so they can look at it later. Student 2 is able to fix the bug, then sends the working code back to Student 1. Student 1 checks to see if the code works, and upon seeing that is does in fact work, turns it in as their completed assignment. Student 2, who wasn't as far along on their assignment, uses the same general format and problem-solving technique as Student 1 when completing their assignment. Student 2 may have done the assignment much differently had they not seen Student 1's code.
NOT OK	When you have spent minimal time examining the problem yourself and **ask for help after like 5 minutes of looking at it.** When you know the other person is already done and maybe willing to help push you in the right direction.

NOT OK	Student A is struggling on how to start writing a program. **Student A knows that student B was writing a few notes how to start but not the actual code itself and thought it was ok to borrow his notes.** However, student B's notes mention on what kind of codes to use and student A ends up using the kind of codes that student B used. Student A could have stopped himself instead and should of use the idea to create notes for himself to help with creating the program, but instead, he crossed the line by using student B's codes.
NOT OK	You've been trying to finish your psa but are stuck on one of the methods and cant figure out how to do it. You call over another student and ask for help and they **let you see what they did for there method to help you get started.**
NOT OK	If one student approaches another for help, and the helping student **simply tells the other what to do,** then that would be an example of a debugging scenario that is not ok.
NOT OK	Student A asks Student B for help debugging a section of a PSA. Student A has completely written the code, and Student B has done most of the PSA but has not completed it. **If Student A asks B for help on a section that B has not completed yet, that is "crossing the**
NOT OK	One person does not understand how to do a part of the PSA and calls another person over the phone or in the lab and **asks them how to do it.** This second person suggests that a while loop should be used to perform this function of the program.
NOT OK	I believe if one student **has done little to no work, if it his/her first day in the lab for example, the day before a PSA is due**, he should not be able to collaborate with others who have put in the time and effort and worked a great deal more than him/her.
NOT OK	You get a compiling error and then you ask another CSE8B student who finished the assignment if they had gotten the same compiler error. The student says yes. Then you ask the student if he can tell you what he did to fix the error.
NOT OK	You are asked to help another student debug their file. Looking through it, you notice that their 'while' loop doesn't have an incrementing line. So you inform the student of their mistake. While looking through the student's Java file, however, **you realize that their 'while' loop is better constructed and more efficient than you own 'for' loop**. Thus, you change your own Java file so that it is a 'while' loop that matches the other student's 'while' loop. This scenario crosses the line and is NOT OK.
NOT OK	It is not okay to not have started or worked for a considerable time on the PSA and to go up to a classmate and look at their work "just to get a feel for what to do." This is how tutors help us too. We have to have applied some logic and have to have developed the core, and then they helped us with marginal things.
NOT OK	Both students are working in the lab. You overhear a student having a similar problem. You ask them what they are thinking, and the two of you discuss it. In the process, you see a portion of their code that seems well put together, and use it in your solution to that problem, despite it being vastly different from your own, such as using an entirely different way to do what the assignment asks.
NOT OK	Student A has just finished their PSA and is about to leave the lab. Student B asks Student A to e-mail the code, and says it's not cheating because he/she will use it as a reference and not copy it word for word.
NOT OK	A CSE8B student has finished his/her PSA and is looking for a peer who can collaborate on what problems they have encountered. The student finds a peer, who hasn't finished the assignment, and they discuss the errors the student who finished had overcame. The peer looks on and gains an understanding of how to do the PSA, therefore "crossing the line" by using the ideas of the student who finished.
NOT OK	Peter needs help with his PSA assignment. He calls his friend Mike. Peter describes his bug, but Mike says he cannot provide adequate aid over the phone. They meet up, and Mike compares his code to Peters. Although Peter and Mike's code ends up being different in names and minor structures, Peter was ultimately only able to finish his PSA because he viewed Mike's code in order to solve his own.
NOT OK	I didn't start on my assignment yet and my friend asks me to look at his code for the assignment to see how to fix his error. I thought his way of of doing the assignment is very efficient so I use the same way on my assignment.

NOT OK	When you ask someone for help that clearly has not yet started on the PSA. By letting them help you out, you are allowing them to get some ideas on how to start and finish the project fastest and technically not on their own. However if they have already started and have done a fair amount or at least is almost equal to what you have done, then it is okay and fine to ask them for help in debugging your code. In this way they should already have an idea of what they want to do and what should be done.
NOT OK	You have not finished your PSA and you are starting to stress. Your friend has finished and is waiting on you to finish so you can both go eat. You ask him to take a look at your code and tell you what your missing. He tells you exactly what you are missing and what is different
NOT OK	Your friend is a "method or two ahead" of you and he asks you too look at his code to help him debug. To help someone debug you both should be at the same level in code, i.e same method pr level of completeness.
NOT OK	Student A asks student B to debug a code. Student A gives student B the whole code through e-mail because it's late at the lab. Student B does debug some of the code, but there is a risk of student B keeping most of the code for himself.
NOT OK	Student A needs help debugging and Student B realizes he has made a similar mistake. Both students have completed their PSA. They work together to solve the bug but student B also uses the same code for his own.
NOT OK	Classmate A has a runtime error on his code and doesn't know what's wrong. He is about 90% done, and needs someone to skim over his code to see what he did wrong. He contacts his friend, classmate B, to check over his code. He sends it to Classmate B via e-mail, and just asks him to find out what's wrong and email him back.
NOT OK	One student has already finished theirs, and is telling another how to do it.
NOT OK	You are confused about how to program some part of a PSA and there are no tutors available, some other students are at the labs along with you. You decide to ask one of them what you should do and they come over and type the program needed for the PSA to work.
NOT OK	...when you are NOT finished, and look over to the student next to you and bribe him or her for answers. simply put.
NOT OK	Copying and pasting an entire .java file asking for help on debugging. The better scenario would be to simply copy and paste the error message given by the compiler.
NOT OK	When a student does not get the assignment and ask others for help, but whoever is helping ends up giving explanation to the point where the answer is obvious and then the student who needed help ends up typing the exact same thing.
NOT OK	A student receives a lot of help from a peer and virtually has the program written for him by the peer. He makes no changes to the program once the peer helps him write the code. The code turned in is exactly the same as the peer's.
NOT OK	It is not okay to copy another persons full or part of his/her code when you become frustrated and can't figure out the problem of your
NOT OK	Student A asks Student B to debug his code on an assignment Student B has not yet started, and Student B accepts. (As any glimpse of completed code would be like looking at someone else's completed exam, one student should not be able to look at another's code until his has been completed and turned in.)
NOT OK	One student encounters a failure with his code and seeks help from surrounding peers. He finds another student who has had the same failure and that student has identified the defect. The student asks this student to tell him exactly what it is that he changed to make it
NOT OK	Bill is just about to start his project and realizes that Bob is just finishing. Bill politely asks Bob if he could see his sample code or perhaps have it emailed to him for reference on his assignment. Bob is hesitant because he knows that even though Bill would use it as a "reference," his work that is out there; and what if Bill becomes so stressed he just starts to copy. So, Bob says that he can't do that, but he can instead have Bill email him if he has any questions or concerns with the assignment.

NOT OK	A student is very busy working on his program the night before it is due. He can't fix this one problem in a certain method. He calls his friend to get help. He asks the friend to e-mail that method to him so he can figure out what is wrong. Whether or not the friend agrees, this clearly "crosses the line".
NOT OK	Showing another classmate the whole program.
NOT OK	A student is stuck on a particular section of code, and goes to a classmate to ask about it. He shows only that one section of the code to his classmate, but it was a part that the classmate hadn't even started on yet. The classmate helps, but has seen the rest of the code that went into that one part.
NOT OK	Bob writes a program that runs and is almost perfect but there is a defect in the code. He asks his friend Sally who is also in his class to look over the code before she has started on the program herself. She starts on the program immediately after she has helped Bob fixed his
NOT OK	Allowing someone to look over your entire code for bugs, especially if that fresh pair of eyes hasn't started yet.
NOT OK	You have just finished with your lab, and scream out "YES YES OMG YES! I'M DONE LIKE ANY TEAM WHO GOES AGAINST THE LAKERS IN THE PLAYOFFS THIS YEAR!" Billy Bob there, sitting next to you, screams back, "You're done?! Dude help me out!" You give him some pointers and tips on the areas where he asks you questions, and after you are done, revert your attention back onto your own accomplishment. Suddenly, Billy Bob asks you, "Hey, I still do not really understand how you did that right here. Think I can use a bit of your code?" As heartfelt and innocent-sounding his light request is, you know that you are not allowed to give out any code. He turn him down, and go out to buy yourself a ticket to fly back home in San Jose.
NOT OK	Showing the other person your entire program and asking him or her to debug it for you
OK	**Both students have completed their programs, but one of the students seem to have a few bugs left in their programs,** keeping them from finishing the assignment. In this scenario it would seem appropriate, since both students have for the most part completed their programs. The student with a few bugs still in their program just needs another pair of eyes to fix them.
OK	If my code compiles with no errors, but the program doesn't run properly and brings up an output that isn't correct, then I may ask a fellow student to look over my code. He or she must then look at my code and find a place where I may have made a mistake. He or she may not type in anything nor tell me the exact code; however, if he or she simply **explains to me how I can go about to solve my problem in English words**, then that would be acceptable.
OK	When it is a bug you have narrowed down where it can be to a **small portion of the code. Just a couple of lines perhaps, or maybe its just in one method**. This way hopefully you don't to show your whole code to the other person.
OK	Student A and Student B are both assigned a PSA. Student A completes the PSA and it works perfectly. Student B is struggling with the assignment and asks Student A for help. Student A comes over and skims the code, gives advice, but not directly telling Student B what to do. (ie. "**Take another look at your loops") or, Student A can also draw a diagram or something to help Student B understand the assignment more clearly.**
OK	Well in case you are really struggling with a bug in your program and there is no tutor around to help you out. Then you should be able to ask someone who looks like has already finished that part of the assignment, to help you out. If he is able to help you **without having you explain, your whole program to him**. Then in that scenario, I think it would be alright for that person to help you fix your bugs.
OK	You compile a code and it shows you the error and highlights it for you. **You are not sure what the error message means or how to solve it, so you ask a friend what the error message might refer to.**

OK	Most of my code works, but an **s.o.p. statement** indicates I have an error in a small block of code and it shows when I run it. However, I can't seem to identify what the error exactly is where the s.o.p. is present. Thus, if still stuck, I would ask another student for **help on that specific small portion of code.**
OK	A CSE student Alice was working on her PSA and she was almost done with it, spending many many hours on it. However, **on the very last step she found that the code did not compile**, and realized with horror that her code had a bug somewhere. She spend many hours trying to look for the bug without success. So she called her friend Jake who already took the class before and asked him to look it over(OK situation). But Jake couldn't find the bug either. So she called her another friend Charity who was in the same class but had already finished the assignment and asked her to find the bug(also an OK situation). Fortunately Charity found the bug as soon as she laid her eyes on the code and Alice was able to finish her PSA on time!
OK	I think one way of a legal helper is when a person is done with their PSA and is able to help the questioner whether debugging or logically is not to physically help or even say the right answer but **help facilitate and lead the questioner into the right method**. For debugging maybe they can have a little more direct aid because of time consuming frustration.
OK	A student has completed writing the entire program for the PSA and only has the debugging part left. If another student is nearby **who has already completed writing his program and is in the process of debugging**, the first student may ask the student nearby for assistance in debugging his program.
OK	I'd find someone (either arrange to meet someone at the labs, or find someone in the same class who came in some time right before or right after I did) and ask them what time they are leaving. Hopefully, we both agree to work apart for some number of hours and then come together to debug each others stuff for a brief spell. Maybe in 2 or 3 hours. By that time, it would be okay to have someone else see your code since **they too have been working on their own for that amount of time** Admittedly, we don't all work at the same pace,
OK maybe	so when i'm debugging my code and someone is struggling (and i can see HAS been for a while) and they ask for help...i can ask them in specific WHAT they want to see or get help with, but i will NOT let them take my mouse and look at whatever they want to...that has already happened to me once and it offended me quite a bit because i never gave them permission (and when i had asked them for help before they grudgingly tried....it came out more like mumbling so i could tell they didn't want to help)...aaanyways... i think it is ok to get a little help, but not to take over and look through the whole code to get to what they want...they can say what they need and the person who is offering help can show them if they feel comfortable to do so. :]
OK	Working with others is OK.
OK	When my code does not work, I would ask my friend to give me a hint. So, when my friend comes by to help me on the code, he and I would generate a code as we discuss through the process.
OK	Share your compile-time error messages with your classmates, if you have trouble figuring out why the error is occurring.
OK	Student A is working on his PSA but he has no idea what to do, then he asks student B about it. After listening to the explanation of the PSA and student B's idea of how to do it, student A is inspired and starts to do his own work without copying exactly what B tells him.
OK	If you are both working on the same part of a program, and have both written at least some code, comparing code to help each other and get ideas is ok. The amount of work done by each student should be roughly equal.
OK	Well, I feel like it's alright to debug and even work on the problem together is alright, as long as you do the work individually. Talking in ideas and the concept is fine, but not going into he specifics, basically, as long as there isn't an "identical answer" it's ok.

OK	A scenario where I think working with another student would be okay would be if maybe you didn't understand what type of error you were getting, and needed someone to explain and help point it out in the program.
OK	I think it is OK to ask another student to look at your code for debugging only if that student has already finish with the PSA homework. But the student helping to debug the other student code must only look for mistakes he has make in his code and give him advices what he might have done wrong. Other than that, it is preferably to ask a tutor, TA, or the instructor for help if you are not understanding how you can write your code, or questions that are more complex.
OK	When two students are stuck at the same part of the PSA, it is fine for both of them to ask each other for help. Otherwise, it is crossing the line since there may be parts of the PSA that one or the other haven't finished yet.
OK	/* I can't remember if I did this survey or not */ It would be okay if you got help from another student that does not take Java 8B. I know a few people that are in CSE12 and ECE who learned C programming but is willing to help me as I am willing to help with them. Although those who learned C did not learn Java, I think it's okay for them to at least try and help me, or give me an idea how to craft the method of executing the commands.
OK	Student A is working on the first part of a PSA and becomes stuck because of a debugging error, and asks student B for help. Student B looks at the code and the error, and is able to give help because he/she has already finished that part of the PSA. (If Student B had not already started working on that part, it would be giving B an unfair advantage to see what A has written.)
OK	You get an error while running the program and you ask your peer to explain what the error means but does not tell you where the error is.
OK	The two students are both working in the CSE lab and one of them comes across a problem and asks the other student a question about the program. The other student can help the student by answering the question, but not looking at their computer screen so they don't get
OK	if both students are stuck on the same part, or if both are done and do their debugging last, then they can look at each others work to try to help, by making suggestions, or pointing them in the right direction, but not specifically doing the work for them
OK	There are 2 students who both finished working on their program but are encountering the same error. These 2 students can work together on figuring out the error. This scenario is ok because both students have worked on their program alone.
OK	Even though everyone has to work on their own, it will be more effective to work with others.
OK	A student describes an example of a problem that he or she is having with their code, and someone else gives the student a generic way of figuring out how to solve the problem. In this way, the helper and one being helped do not see the others code at any time.
OK	If both students have very similar progress done on the assignment (in the same areas of the assignment), or if one of the students who is ahead of another student helps/ works together with that student.
OK	Student A needs help with a bug/ small bug on her PSA. She asked her friend, student B for help. Student B looked over the PSA and quickly identified the error. On the process of doing so, student B learned a more efficient way of using the code but does not use it to
OK	Both students are in about the same position, and one student is having a hard time figuring out the bug in his or her code. Then, she or he can ask the other student about his or her bug in the code. I think the importance is that people making sure that the other student is in about the same place as the person who needs help. In that scenario, then the person cannot cheat off from the another person since they are already in the same position; there is nothing that fresh to copy off from.

OK	Since it is very hard to draw the line, it is all depended upon the students themselves to be honest. However, we can set a rule that students, who helps other to debug their codes only allows if and only if they have their own work done and turn in already. We might need to have a web that displays those people who had their work done and willing to help other students, then students who need help can send their whole program to the finished student for help.
OK	Usually I get asked help from a friend of mine who is taking C, and does not know any Java. Usually when I write code, it takes me a long time and after around an hour or two of programming I start to make very careless mistakes and my friend helps me with some debugging. I help her with C, even though I don't know C, but such careless mistakes are easily overlooked.
OK	Matt has been working very had on his PSA for many days and he has been stuck for hours on a certain error. Matt's friend Richard has already finished, and decides to ask him to look over his code. Richard realizes he made a rudimentary mistake and has created an infinite "for loop". Richard suggests he should check his for loop. Matt quickly realizes he has an infinite "for loop", fixes it, and eventually finishes
OK	When a bug pops up and assistance is needed it is OK to show your code to a friend but both must be comfortable with it and the one who is helping should be honorable and not steal ideas
OK	A student who is almost finished with the program asks another student who already finished the program and turned it in.