Using the Best Resources

Our Perspective

As your instructors and support staff, we are here to help you maximize your success. Please don't hesitate to reach out for help if you feel you need it. Keep in mind that as we get further into the course, we expect you to practice greater autonomy in finding and answering your own questions. As a general practice, you will likely strike a balance between keeping up with the class pace, and giving yourself enough time to productively struggle, if you spend at least 10-20 minutes attempting to solve a specific problem before reaching out for help. In other words, 10-20 minutes with no measurable progress. Exceptions to this recommendation will certainly occur, such as requests for exercise instruction clarification, requests for additional resources, and requests for code review of a completed solution. In the beginning of the course especially, students will often request assistance with various steps in the problem solving process. This is natural and to be expected, especially if you are less experienced in working through technical problems. As the course progresses, we expect the following steps increasingly to be worked through on your own.

The Exercise Completion / Problem Solving Process (PSP)

(At any point during this process, taking a break may be best, especially if you feel yourself becoming frusterated or distracted.)

1. Set an overall goal and break the goal down into smaller steps / problems. In the case of curriculum exercises, this is largely done for you and the goal is to simply "complete the lesson exercise steps." In larger projects, this step will become increasingly important.

2. Understand the problem needing to be solved and how to succinctly describe it. For example...

    - "There is a syntax error in my login.js file on line 33."
    - "I'm not connecting to the database and my credentials appear to be correct."
    - "There is a compilation error when I uncomment this static method."
    - "This image's aspect ratio is distorted."

3. Research the problem and potential solutions further if you don't have clear context.

    - Reread exercise instructions (if applicable)
    - Refer to the curriculum (check the appendix too)
    - Check formal documentation
    - Google search

    - Search StackOverflow
    - Check general resource sites like MDN or w3Schools
    - Watch or read through online tutorials
    - **WARNING**: often students spend too much time on this step. Limit research time; get CODING!

4. (possibly) Break the problem down into a series of smaller, simpler problems. This will often allow you to make progress more quickly.

    - "This class doesn't compile, lets add each method and property back in one-by-one and see where the compilation error occurs."

5. Create a solution hypothesis and test it like a scientist. (try to always have an expectation for what output you will get)

    - "I postulate that if I run my code now having changed the names of my local variables, I will get an output of "test" and "loop ran" in the console.
    - "I postulate that changing the display property on this element to block, I will see the elements vertically aligned."
    - *Where possible, inject places where you can print out values to better determine where the problem is occuring and if a potential solution solves the problem.*

6. Keep track of various possible solutions tested. Be prepared to articulate them.

    - "I tried possible solution a, b, and c and got x, y, z results but I need to get a q result."

7. Analyze the output of your test; if it succeeds, check with at least two other values to make sure it is an effective solution. If it continues to succeed, HOORAY, you solved the problem! It's time to move on to the next problem. If the output is wrong, form another solution hypothosis and begin testing it.

8. (optional step to take it to the next level) Seek another solution that also solves the problem. Compare and contrast and pros and cons of the multiple possible solutions.

9. (another optional step to take it even further) Play around with other related problems and attempt to solve them too!

Asking for assistance

If you have been working through the problem solving process for at least 10-20 minutes without progress, begin reaching out for help.

1. Ask your neighbor or another classmate for help
2. Ask a student fellow.
3. Ask an instructor for help.
4. Post the question on a reputable developer / programming forum (follow their community rules and guidelines)

When you ask for help, be sure you can clearly articulate the following to the person you are asking for help:

1. The expected output
2. The incorrect current output
3. The solutions you have tried (possibly what resources you have consulted)
4. Your best guess at what the problem may be
5. Your best guess at what a possible solution may be