Coding Challenges

Wait until you are told by your instructor to start!

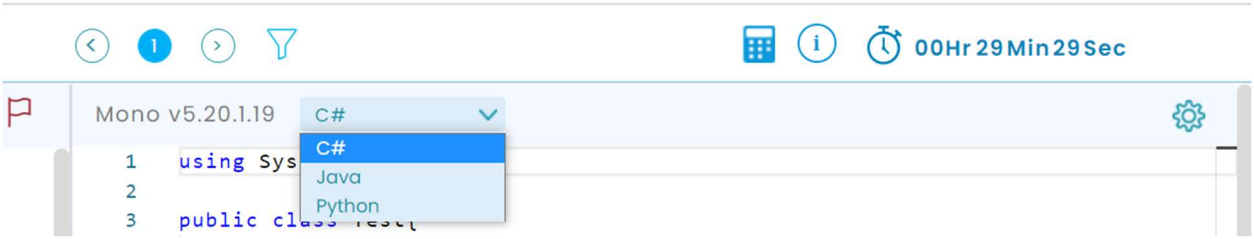
There is a time frame for when the challenges can be taken. There is also a time limit (usually 1 hour for 2 questions). For example, the timeframe may be from 2-4 pm. That does NOT mean you have 2 hours to take the challenge. The time limit is also in effect. If you open the challenge at 2:03, it will stop at 3:03. SO, do not try to access the challenge before your trainer tells you to, or your timer may start early!

Choose your LANGUAGE!

Your tech stack will have a core language. You MUST use your core language! Let's look at an example. You are familiar with Python, but you are learning JAVA in Revature training. You take the challenge in Python and pass all cases in both questions. You will score ZERO because Python is not your core language. Java is!

*****You will need to select the appropriate language <u>yourself</u>. When you do, there will be a warning that you are switching languages. Select OK. Take challenge in correct language.** 😊

In the image below, can see that there are 3 choices (C#, Java, and Python). The actual coding challenge will probably have a few more options.



DOs and DON'Ts

1. Solve the problem, not the expected output.
    a. You only see 3 out 5 test cases.
2. Fully understand the problem
    a. Read the problem and think of how to solve it. Then, look at the example input/output and read the explanation to make sure it fits with your solution
3. Use pen and paper
    a. Work out your algorithm completely before starting to code
4. Know your syntax
    a. iMocha does not "Help" you. It will not underline syntax errors in red. It will not make suggestions. There is no auto-complete option that will show you available functions/methods. YOU need to know it.

5. Learn to read the stack trace
    a. If your code does not compile, read the output in the console. It is telling you what the problem was so you can find the error and fix it.
6. Make sure you know what to return (value, format, data type)
    a. You are writing the logic in a method and returning a value. Make sure you know what you should be returning. Not sure if you should return a String (“123” or an int (123)? Hard code the answer (return “123”;) to check, but make sure to switch it back!
7. You can switch between questions before submitting the full challenge
    a. Problem 1 to tough? Work on problem 2 first, then go back to problem 1!

Do NOT touch the Stub Code!!!

```
public static int exampleMethod(int inputValue) {
    //WRITE YOUR CODE HERE




}
//DO NOT TOUCH THE CODE BELOW
```

The stub code (the code that is already there when you start a question) is important. Do NOT change it. It is used to make sure that the proper input and output are used to run the testing. If you change it, it will not help you. It can only HURT you.

Your code will be written inside a method/function. You will return a value. That’s it.

The stub code will call the method and take care of output.

One More Reminder: Choose the correct language for your cohort’s tech stack.