

## Objectives

- Create a Java program that produces output using an IDE for development
- Learn how to submit to and interpret the I/O test cases in Gradescope

## Introduction

In this assignment, you will be introduced to one of the software development tools that we will be using in the remainder of the course. An Integrated Development Environment (IDE) is a software application that software programmers use to allow them to productively develop new software. It includes a text editor and tools to help resolve programming errors (“bugs”).

Visual Studio Code is a popular IDE that runs on Windows, Linux and Mac. The software is installed in the lab and at the clusters. You can also download to your own computer to use for the course. We recommend following the instructions at this site to set up your environment. <https://code.visualstudio.com/docs/java/java-tutorial> . If you have a laptop, you can bring it to the lab and ask your TA for assistance if you run into difficulties.

## Steps

1. Click on the Fish Tank Lab in MyCourses.
2. Create a folder for your Java program, you can name it whatever you like
3. Start Visual Studio Code (VS Code)
4. In VS Code, Open the folder.
5. Create a file and save it with the name `FishTank.java`
  - make sure your file name has the exact capitalization as above, Java is a case-sensitive programming language
6. In your java file enter the following code

```
1 public class FishTank {  
2     public static void main(String[] args)  
3     {  
4         System.out.println("Hello");  
5     }  
6 }
```

7. Click the run triangle icon in the upper right hand corner of VS code or press the F5 key. You should see in the terminal that comes up that **Hello** gets printed out (similar to the below). For reference the equivalent code in python would be `print("Hello")`.

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\matt\Desktop\FishTank> & 'C:\Program Files\Java\jdk1.8.0_144\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:59334' '-cp' 'C:\Users\matt\AppData\Roaming\Code\User\workspaceStorage\37558bc199b38a671305fb72e155d45a\redhat.java\jdt_ws\FishTank_c4eadf31\bin' 'FishTank'
Hello

```

If you have an error message or hello does not print out, talk to your TA or another student in the lab to see if they know why your code isn't working correctly.

8. Now you want to update your code so that it prints out the following (see below). Note the line numbers are just for reference, you don't print those. The text "art" below is meant to look like fish in a fish tank.

```

1
2      +=====+
3      |  ><((( (*>      |
4      |                  <_><      |
5      |      <)><      <><      |
6      |          ><>      |
7      |                  >('>      |
8      +=====+
9

```

There are different ways to approach this, but an easy way would be to use multiple `System.out.println` statements. This built-in method to Java prints the string you enter between the parentheses and adds a new line character to output (we see as line break in displayed output). This is a good resource to refer to if you would like to learn more about printing strings in Java at <https://math.hws.edu/javanotes/c2/s4.html>

9. Submit your file to Gradescope which will run test cases to make sure that you have produced code that is correctly written and produces the required output. You may find that your output doesn't match what is expected causing the test case to fail. Gradescope requires that you have the correct files uploaded and that they compile before it will attempt to run other tests against your code. In this lab there is only 1 test that sees if you produced the correct output.

10. Your goal is to pass all test cases, the score you see displayed in Gradescope is the grade you earn for this lab. In this case if you don't pass the output test case your score is 0. Some students in the past have thought that if they just submit the file, they are all done, you need to have your code pass the test cases to get awarded full credit for the lab. You can submit to Gradescope and update your code as many times as you need to. Just be sure to have completed by the lab due date.

TIPS for approaching the lab.

- The expected output is meant to print a blank line on line 1.
- White space and newlines matter, you may think your output is "close enough", but computers are very "picky" they need your output to exactly match what they are "expecting" in order to say it is the same. Some students find this frustrating, this is part of programming, computers are very literal and cannot infer that something is the same even if you think they should!
- It could be helpful to work on just a couple of lines at a time. This is a technique in programming where you work on little pieces at a time, gradually building to the correct solution. Below is a snapshot of some points in time while the code was developed. The test case uses a diagnostic output to help guide you in correcting your output. It's not perfect, but is still very helpful.
- Below is a walk through of some stages in the code that I produced to write a reference solution program for this lab.

#### Test 1 (0/100)

Input

Output

Line	Expected	Actual
1		
2	+=====+	+=====++
3	><((( (*> _	
4	<_><	
5	<)>< <><	
6	><>	
7	>('>	
8	+=====+	

For this first submission, I only had 2 println statements. Note: I have a 0 because my code isn't producing the correct output. The first line was correct. The second line has yellow highlights indicating areas where some changes are needed. Expected is what the "correct output" is (what the computer is using as a reference) and Actual which is what your code is producing. As your output gets closer to the expected, the diagnostic output can be more helpful. Also note that this test case has no input, meaning the test isn't acting if anything is being typed in at a keyboard. In future labs and assignments there will also be input for I/O tests and a failed test case will show what the actual input is. Stay tuned for more details...

**Test 1 (0/100)**

Input

Output

Line	Expected	Actual
1		
2	+=====+	+=====+
3	><((((*> _	
4	<_><	
5	<)>< <><	
6	><>	
7	>('>	
8	+=====+	

In this case, I made some updates to my output for the second println statement. And it's showing that I just need to add one more = to have it match the output for that line. Even though I'm getting closer to the correct solution, I still have a 0, not what I want.

## Test 1 (0/100)

Input

Output

Line	Expected	Actual
1		
2	+=====+	+=====+
3	><((( (*> _	><((( (*> _
4	<_><	<_><
5	<)>< <><	<)>< <)><
6	><>	><>
7	>('>	>('>
8	+=====+	+=====+

Here I had added in all 8 of my println statements. This indicates I'm close and that I just needed to add a space on lines 5 and 7 and delete a space in the area indicated in Actual. So close, this looks right to me, but it's just off a little and the computer isn't buying it, still a 0 😞

## 1L Fish Tank Lab

● Graded

Student

Unknown Student (removed from roster?)

Total Points

100 / 100 pts

Autograder Score

100.0 / 100.0

Passed Tests

Test 1 (100/100)

Late Points (0/0)

After I corrected those errors, Finally my 100 on the lab 😊

You want to make sure that the score under total points is 100 as that is the score that will get recorded in My Courses. There is a lag between the time you complete the lab in Gradescope and when the grade is updated in Canvas. This is because all labs (and assignments) allow for submitting after the due date (but with a point penalty)