Training - 30 min, rules, guidelines, and tips

1) Accounting adds up all the hours on Pike13 and pays us every other Friday. New hires will get paid for 2.5 hours of training. This includes a 1hr video training, a 30 min training session, and 1 hour of shadowing. It's OK to do more shadowing, but anything extra would be unpaid.

2) Always arrive on time. Be punctual.

3) Do not share personal emails with parents or students. The same goes for social media or other forms of communicating. Never meet with a student or parent outside of school.

4) Once a year you must give your students a talk about internet safety.

5) Don't create a first-person shooter game with the student or anything that depicts violence, gory, drugs, or sexual content ect...

6) When teaching a lesson with two students, share the time equally between both students.

7) Always wear your lanyard at the coder school and during your lessons.

8) Due to the pandemic, Coaches, Students, and Parents are required to always wear a mask inside the building. They must also get their temperature checked at the front desk.

9) Notes: You **MUST** always do the **notes** and **app reviews. This is part of the job!** Notes are done every day. Preferably, end the class 5 minutes early and do the notes during that time. At the latest, complete the notes by the end of the day. If you create an account with a new student, write their username, email, and password in the notes so the parents and student can access their account from home. Always keep the notes positive. If there is an issue with the student, let the GM (Marcel) know privately, do not write it in the notes.
Examples of what <u>NOT</u> to write:
      1) John had trouble focusing, he 10 min playing games, and then I got him to code...
      2) John and I copied a tutorial from the internet...
      3) John taught me and another student some things we didn't know in Python...
      4) The internet was down so we started 10 min late...
      6) The building was locked so we started 10 min late...
Everything above also applies to **app reviews.** App reviews are done every 3 months and are more in-depth with what students have worked on how they are doing in class. Notes + will tell you when an App Review is needed.

10) Teaching online over zoom:
      a) For safety, zoom records the entire session to the cloud by default.
      b) Your <u>camera should always be on</u>.
      c) Take advantage of zoom's annotate tool, enable multi-share, use comments.

12) Please follow the coder school convention for creating new accounts for students. This avoids students forgetting their passwords and allows for coaches to access their accounts when needed for reviews, or prep.
The convention is as follows:

f = first letter of their first name, **l = last name**, **keep everything lowercase**

nsd (North San Diego Location)	enc (Encinitas Location)
Email: nsd+**fl**@thecoderschool.com **User: fl**-tcssd **Pass: fl**coder	**Email:** encinitas+**fl**@thecoderschool.com **User: fl**-tcsenc **Pass: fl**coder

Example: John Smith is new at The Coder School in North San Diego and needs an account on Trinket.io. His account credentials are below:

Student **John Smith**
User: jsmith-tcssd
Password: jsmithcoder
Email: nsd+jsmith@thecoderschool.com

13) Don't share the trinket & scratch accounts we provide as a resource for your training to students.

14) Refresh/check the Pike13 calendar including the day you teach on in case of any last-minute changes made in the morning.

15) 1-hour classes during the week start on 30min increments. Example: 2:30 - 3:30, 3:30 - 4:30, 4:30 - 5:30 ect... Therefore, keep that in mind when planning your availability with the GM (Marcel).

16) Let the GM (Marcel) for no-shows. After 10 minutes, if the students haven't arrived yet, let the GM (Marcel) know either by text or email.

17) If the GM (Marcel) is not in the building and the phone rings answer it. If it is a parent take a message and relay that message to marcel later.

18) If you are teaching a lesson and person comes in asking questions or interesting in enrolling their child, tell them that you are busy teaching a class and have them leave their email and phone number on a piece of paper and the GM (Marcel) will get back to them later.

19) In order for a student to learn Roblox parent permission is required with approval from the GM (Marcel) as well.

20) <mark>No Food or Drinks inside The Coder School.   We don't want to attract unwanted visitors.</mark>

21) When teaching the student, interact with them, teach them a concept, then have them practice it, then create a game. If you think you are talking too much, this is a red flag, keep the kid engaged.

22) <mark>Don't continuously quiz the kid. For, example, you just showed them a concept and then the next thing you do is engulf them with quiz questions on that concept. We want the kids to be relaxed and enjoy learning.</mark>

23) Always be prepared to teach. Have in mind what topic, or project you will teach them when coming to class. Don't ever ask the student; "hey what do you want to do?" This gives a bad impression makes you look unprepared and most of the time the student won't know anyway. If you are really not sure what to do with them, give them 2 or 3 options and have them pick.
        Example: A coach calls out and you must substitute for them. Then arrives their student John, and you are not sure what to teach them. **You can say:** Hey John, shows me some of the python projects you've worked on (so you can see how advanced they are) ... Oh that's cool, looks like you have some basic experience. So, in terms of what to do today, there are couple things we can do, I will let you decide; 1) if there is a python concept you have trouble with, we can review that to help you understand it better, 2) Or if you feel comfortable with you programming skills, we can create X project together.

24) <mark>Before introducing a new topic/subject, ask the student if they have worked with that topic/subject. If they have, ask them about some of the concepts they learned and how comfortable they feel with the topic/subject so you can judge how advanced they are.</mark>

TIPS:
- ❖ As a coach, it is ok to know everything 100%. We don't expect you know all the programming languages we teach here, or all the tools, or all the concepts ect... You will learn as you coach. Start off with the basics, create a few projects in scratch and python, teach it, and go on from there.
- ❖ There are some kids with learning disabilities, some have genuine problems focusing, it's ok to give them a 5 min break in the middle of the lesson to help them recuperate themselves. Take it easy with those students.
- ❖ There are some students (typically students age 10 and younger), that just want to goof off, play online games, and do whatever they want. Obviously, we NEVER want to punish the student or yell at them, but it is important to present yourself as assertive. A good coach has the perfect balance of assertiveness, respect, and friendliness towards the student. Be

- cool with them and have fun, but don't let them control the lesson, after all you are the coach.
- ❖ Be aware of the learning style of the student. Ask about their interests, some kids are gamers, other like learning about algorithms, some simply enjoy creating any project.
- ❖ Be aware of the student's age, you may come across a student that is 7 years old and barely knows how to read, use a keyboard, and mouse. You may be explaining something but they don't know the meaning of the technical words you use, like cursor, interface, coordinate, ect..
- ❖ Take in account their interests. It's good to ask them about sports, or things they like to do and try to incorporate that with their projects. Girls tend to like more creative projects while boys tend to like games.
- ❖ My advice is to create projects with the student that take around 1 - 3 lessons to complete.
- ❖ Every coach is different but typically when I have a new beginner student learning Python, I make a couple cool but basic programs for the first couple weeks to capture the student's interest, then after that I interchange between teaching them a concept and creating a project using that concept. **Here is an example below:**

Week 1: Create a python program that draws cool shapes on the screen. — the goal here is to just to have fun and show them some cool stuff
Week 2: Create a bear by using circles. — the goal here is to have them practice with the coordinate system for placing circles
Week 3: Create a tree — Very similar to the bear project but this time we are using **loop** to create several leaves.
Week 4: Introduce them to **if-statements**
Week 5 - 7: Create a project using if-statements — The project I always do is generating a maze using if-statements, random number generators, and loops
Week 8: Introduce them to **functions**
Week 8-11: Create a project using functions — I do a tic tac toe game with them using functions
Week 12: Introduce them to Lists
Week 13-15: Create project using lists
And so on ….

- ❖ For scratch students, lessons don't have to be as structured since most kids are under 10 years old and they don't have the capacity to focus on specific topics. They mainly just want to create games. As long as you start off creating simple games with beginners and gradually more complex games as they learn than that should be fine.

In reality, 95% of the students are super easy going and great to work with. The first month of a new coach is always the trickiest since you are new, but you will learn quickly, and you will get to know your students well. It gets easier the more you teach and watching you kids grow is rewarding. The Coder School truly is a great place to work and learn! Please reach out for any further questions, or advice. We are here for you.

Important links:
Toolset: https://server.thecoderschool.com/toolset/
Notes: https://server.thecoderschool.com/notes/noteSelector.php
Find A Student: https://server.thecoderschool.com/portal/portalsearch.php
Schedule Encinitas: https://tcs-encinitas.pike13.com/
Schedule North San Diego: https://tcs-northsandiego.pike13.com/