Designing fluency puzzles and games:

- Both fluency puzzles and games should be designed to have students working many, many problems in a relatively short amount of time.
- Puzzles (and sometimes games) build students observation and reasoning skills (MP.2, MP.7, MP.8) as well as their perseverance in problem solving (MP.1).
- Games add the aspect of competition, motivating an improvement in speed and accuracy.
- Use math language both to name the game and while playing the game.

Examples of Math Game Names	
Mathematical Name	**Non Mathematical Name**
Partners fo Ten	Fill the Tank
Name the Polygon	Figure it Out
Measure the Length	Roll 'Em!
Equal Sums and Products	Balancing Act
What is the Difference?	Size it up!
Make Units	Pack the Bags
Exchange to Get to 100	Race to 100
Fractions of a Set	Give Me Some!
Finding 10 Percent	Chocolate Chip Cookies
Number Line Clock	Tick Tock Clock

- Simple is better – this allows the focus to be on the math, not the technology or rules, or other aspects of the materials. Use simple cards, dice, spinners, or personal dry erase boards with uncluttered templates.
- Design the competitiveness of the game to fit well with your class culture. Games can be cooperative, team competitions, or have built in luck factors to diffuse the level of personal competition.
- After a game, use a student debrief to closely tie the experience to the mathematics.

Designing Sprints:

- Write the Sprint to with a specific lesson or topic in mind. The goal of the Sprint is to build speed and accuracy with a topic that students have already mastered.
- Write the Sprint to account for a large range of skill levels.
- Write the Sprint to be easy enough for every student to experience the adrenaline of a race and thus build confidence by getting a lot right. Every student should be able to get no less than 10-15 right. Every time you deliver a Sprint in which one or more students get less than ten correct, rewrite the Sprint to make the first ten problems easier.
- Write the Sprint to be hard enough so that no student finishes the Sprint in under a minute, thus every student will be motivated to improve on the second Sprint. Every time a student finishes a Sprint in less than one minute, rewrite the last ten problems to make them harder.
- Create sequences that will help students learn to recognize patterns and methods for solving problems on their own. For example, solving the addition sentence 9 + 2 can help a student solve problems like 29 + 2, 49 + 2, and 89 + 2. These patterns with gradual jumps in complexity help students develop deeper number sense by recognizing number relationships and tricks on their own without them ever being taught. The addition sequence below illustrates.
- Problem-by-problem, the degree of difficulty should correlate between the two Sprints.