

# Activity 13 Assessment

## Classifying Triangles

### Analyzing and Classifying 2-D Shapes and Using Algebraic Thinking

Reads and alters code by testing out various values or blocks until desired outcome is attained.

```

repeat (2)
  move (150) steps
  turn (80) degrees
  wait (1) seconds
  move (100) steps
  turn (80) degrees
  
```

"I'm going to change the steps to 50 and the wait to 2 and the degrees to 100."

Reads and alters code by visualizing and explaining the impact of changes until desired outcome is achieved.

```

pen down
repeat (2)
  move (150) steps
  turn (80) degrees
  wait (1) seconds
  move (50) steps
  turn (150) degrees
  
```

OR

```

pen down
glide (1) secs to x: (150) y: (100)
glide (1) secs to x: (100) y: (100)
glide (1) secs to x: (120) y: (150)
glide (1) secs to x: (170) y: (150)
  
```

"I'll change the steps to 50 and the degrees to 120, so they add to 180 and it's a rhombus. I'll delete the wait because it doesn't impact the end image."

OR

"This is going to make a narrow parallelogram because I can visualize the vertices."

Reads and flexibly alters code and makes sense of conditional statements related to outcomes of code when classifying shapes.

```

if (touching Basketball) then
  start sound Pop
  change size by (10)
  
```

OR

```

repeat (10)
  if (angle > 90) then
    Oblique Triangle for (1) seconds
  if (angle = 90) then
    Right Triangle for (2) seconds
  if (angle < 90 and angle < 50) then
    Acute Triangle for (2) seconds
  
```

"I'm going to make the condition that if Balloon is touching Basketball, it 'pops,' but if it's touching the edge, it gets bigger." OR "I've created conditions for the 3 types of triangles based on the greatest angle, but now I still need to account for any other values like 0, 160, and 180 or more."

### Observations/Documentation

## Activity 13 Assessment Classifying Triangles

### Analyzing and Classifying 2-D Shapes and Using Algebraic Thinking (cont'd)

Uses basic blocks to write code for a desired outcome.

“I tried using these blocks in this order, but it didn’t make what I wanted.”

Uses more complex blocks (including repeat and conditional statements) to write code for a desired outcome.

“I wrote code, but it used so many blocks. I can see that these blocks repeat. So, I used the repeat block instead and deleted these other blocks. I put it inside the ‘If, then’ because if it has 3 sides, it will draw this triangle.”

Uses conditional statement blocks to flexibly write different code related to outcomes and the classification of shapes.

“Writing code with conditional statements is like creating a flow chart. All the possibilities must be accounted for. If the condition isn’t met, then we need to have other options, with the ‘else’ block defining the other situations”

### Observations/Documentation