**Part 1:**

***Note****:* ***In order to stop an execution of the program, please click once on the “Python Turtle Graphics” window.***

**hit\_the\_target.py (miss)**

**A picture containing graphical user interface

Description automatically generated**

**hit\_the\_target.py (hit)**

Chart

Description automatically generated with medium confidence

***Note****:* ***In order to stop an execution of the program, please click on the “Python Turtle Graphics” window.* (P. S I did this, so that the window will not close itself right after the turtle hit or missed the target and the program stops execution)**

1. In order to determine the turtle’s coordinates on x and y plane use **.xcor()** and **.ycor()** methods on the turtle object respectively to return current position of the turtle.
2. To determine whether the turtle’s pen is up use the **not** operator along with **.isdown()** method on the turtle object as well as conditional if statement.

**if not (turtle.isdown()):**

***statement here (e.x* turtle.pendown())**

1. In order to find turtle’s current heading use **.heading()** method on the turtle object which returns a heading in degrees.
2. To determine whether the turtle is visible use **.isvisible()** method on the turtle object which returns boolean expression of either true or false.
3. If you use **.pencolor()** method on the turtle object without passing any argument and the function will return the current drawing color as a string. Similarly, if you want to determine current fill color, use **.fillcolor()** method on the turtle objectwithout passing any argument and the function will return the current fill color as a string. In case you want to determine the current background color of turtle’s graphics window use **.bgcolor()** method on the turtle object without passing any argument and the function will return the current background color of the turtle’s graphics window as a string.
4. In order to determine the current pen size use **.pensize()** method on the turtle object without passing any argument and the function will return current pen size as an integer.
5. To determine the turtle’s current animation speed use use **.speed()** method on the turtle object without passing any argument and the function will return the turtle’s current animation speed as an integer.

I decided to slightly modify the problem from the book by adding random pen color for each execution of the program by importing random module. The program finishes execution by clicking the “Python Turtle Graphics” window. When the turtle hits the target, background changes to green and “Hit!” gets printed out. When the turtle misses the target, background changes to red and “Missed!” gets printed out.

**Part 2:**

**age\_classifier.py**

**Text

Description automatically generated**

I wrote my code withing an age\_classifier function because it is better for code reusability. I did not change any aspects of the assignment when it comes to the structure and correct output. In this example, I learned basic conditionals, and how to structure logic by using if/elif/else/ structure. The age\_classifier function takes one parameter (person's age) and determines whether the person is an infant, a child, a teenager, or an adult.

***(Part 3 is on the next page)***

**Part 3:**

**roman\_numerals.py**

**Text

Description automatically generated**

In this program, I learned how to use comparison operators such as == as well as if/elif/else structure. The roman\_numerals function takes one parameter (number) within a range from 0 to 10 and returns Roman numeral version of that number.