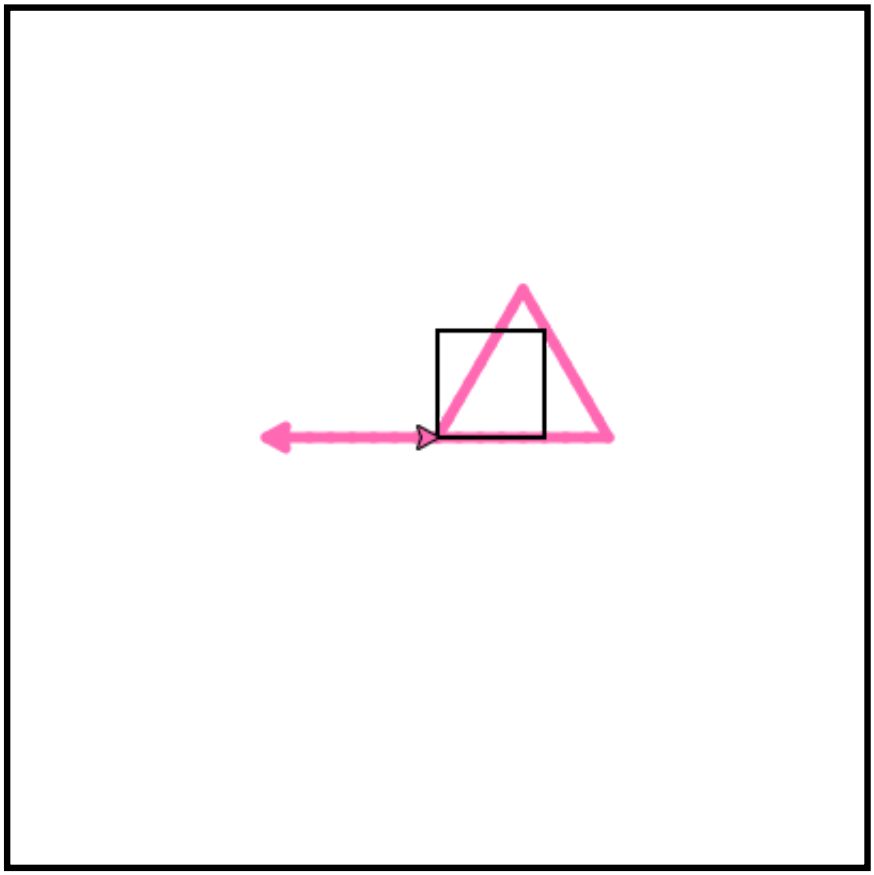
Instances — A Herd of Turtles

Just like we can have many different integers in a program, we can have many turtles. Each of them is an independent object and we call each one an **instance** of the Turtle type (class). Each instance has its own attributes and methods — so alex might draw with a thin black pen and be at some position, while tess might be going in her own direction with a fat pink pen. So here is what happens when alex completes a square and tess completes her triangle:

```
18
19 tess.right(180)                  # turn tess around
20 tess.forward(80)                 # move her away from the origin
21
22 alex.forward(50)                 # make alex draw a square
23 alex.left(90)
24 alex.forward(50)
25 alex.left(90)
26 alex.forward(50)
27 alex.left(90)
28 alex.forward(50)
29 alex.left(90)
30
31 wn.exitonclick()
32
```

ActiveCode: 1 (ch03_3)

Run | Start Audio Tour



Here are some *How to think like a computer scientist* observations:

- There are 360 degrees in a full circle. If you add up all the turns that a turtle makes, *no matter what steps occurred between the turns*, you can easily figure out if they add up to some multiple of 360. This should convince you that alex is facing in exactly the same direction as he was when he was first created. (Geometry conventions have 0 degrees facing East and that is the case here too!)
- We could have left out the last turn for alex, but that would not have been as satisfying. If you're asked to draw a closed shape like a square or a rectangle, it is a good idea to complete all the turns and to leave the turtle back where it started, facing the same direction as it started in. This makes reasoning about the program and composing chunks of code into bigger programs easier for us humans!
- We did the same with tess: she drew her triangle and turned through a full 360 degress. Then we turned her around and moved her aside. Even the blank line 18 is a hint about how the programmer's *mental chunking* is working: in big terms, tess' movements were chunked as "draw the triangle" (lines 12-17) and then "move away from the origin" (lines 19 and 20).
- One of the key uses for comments is to record your mental chunking, and big ideas. They're not always explicit in the code.
- And, uh-huh, two turtles may not be enough for a herd, but you get the idea!

Check your understanding

trl-11: True or False: You can only have one active turtle at a time. If you create a second one, you will no longer be able to access or use the first.

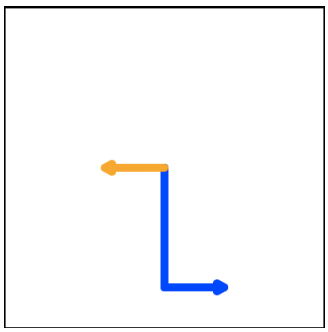
- a) True
- b) False

Check Me | Compare Me

Correct!! You can create and use as many turtles as you like. As long as they have different names, you can operate them independently, and make them move in any order you like. If you are not totally convinced, try interleaving the instructions for alex and tess in ActiveCode box 3.

Mixed up programs

trl-12: The following program has one turtle, "jamal", draw a capital L in blue and then another, "tina", draw a line to the west in orange as shown to the left, . The program should do all set-up, have "jamal" draw the L, and then have "tina" draw the line. Finally, it should set the window to close when the user clicks in it.



Drag the blocks of statements from the left column to the right column and put them in the right order. Then click on *Check Me* to see if you are right. You will be told if any of the lines are in the wrong order.

Drag from here

Drop blocks here

```
import turtle
wn = turtle.Screen()
```

```
jamal = turtle.Turtle()
jamal.pensize(10)
jamal.color("blue")
jamal.right(90)
jamal.forward(150)
```

```
jamal.left(90)
```

```
jamal.forward(75)
```

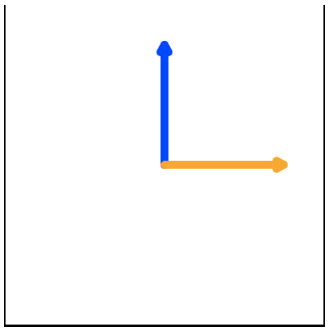
```
tina = turtle.Turtle()
tina.pensize(10)
tina.color("orange")
tina.left(180)
tina.forward(75)
```

```
wn.exitonclick()
```

Check Me Reset

Perfect!

trl-13: The following program has one turtle, "jamal", draw a line to the north in blue and then another, "tina", draw a line to the east in orange as shown to the left, . The program should import the turtle module, get the window to draw on, create the turtle "jamal", have it draw a line to the north, then create the turtle "tina", and have it draw a line to the east. Finally, it should set the window to close when the user clicks in it.



Drag the blocks of statements from the left column to the right column and put them in the right order. Then click on *Check Me* to see if you are right. You will be told if any of the lines are in the wrong order.

Drag from here

Drop blocks here

```
import turtle
```

```
wn = turtle.Screen()
```

```
jamal = turtle.Turtle()
jamal.color("blue")
jamal.pensize(10)
```

```
jamal.left(90)
jamal.forward(150)
```

```
tina = turtle.Turtle()
tina.pensize(10)
tina.color("orange")
tina.forward(150)
```

```
wn.exitonclick()
```

Check Me Reset

Perfect!