Homework 8

1-1: Chassis, body panels, engine, drivetrain, interior, wheels, paint

Chassis, engine, and drivetrain are probably manufactured at the assembly plant

1-2: Bed, desk, chair, table, clock, computer

1-3: RTX 4090, Ryzen 7 7950X3D, whatever motherboard, 1300W PSU, 3TB NVME SSD, custom water cooling loop

None of the components are built from scratch by the assembly company, because I'm building that thing myself!!!!!1!!

2-1: To call for the use of specific components from the available java libraries

2-2: Java provides special syntax for the primitive operations, and because primitive types are part of the java language itself and do not need to be ordered with "import"

2-3: Primitive – it would use the boolean data type.

2-4:
```
if ((0 <= yourAge) && (yourAge <= 3)) {
      output.println("My, just 2 years old!");
      output.println("What a cute little baby.");
       } else {
      output.println("Thanks for entering your age.");
       }
```

2-5:
```
4 - "Oh no! A teenager!"
      "So sorry!"
   15 - "Oh no! A teenager!"
      "So sorry!"
   105 - "Oh no! A teenager!"
       "So sorry!"
```

3-1: The Math.rand() method. I understand that it provides a random number but I don't know how.

The SimpleReader type. I understand that it reads input from console, but I don't know how.

3-2: The second possibility is much more common than the first. This is because a person doesn't need to understand how a combustion engine operates in order to drive a car. The same reason shouldn't apply to software components because in order to properly implement something in a program, you need to understand how to use it so you don't implement it incorrectly.

3-3: The quadratic formula and Pythagorean's theorem

3-4: If you're able to mathematically model something, you're essentially breaking it down to it's bare structure in order to best understand each part of it.

3-5: -1, 13, 15

3-6: Because each of the values within it each have their own constraints

4-1: Set, read

4-2: Because the contract tells us

4-3: Incoming values of parameters because it tells us what needs to be input for the program to work.

setHours is not specified that way because it does not use the #new_hours variable

4-4: myClock = 3:25:48AM, newHours = 3;

4-5: `void setSeconds(int newSeconds)`

Sets `this.seconds` to `newSeconds`.

Parameters:

`newSeconds` – the new seconds for `this`

Updates:

`this.seconds`

Requires:

`0 <= newSeconds <= 59`

Ensures:

`this.seconds = newSeconds`

4-6: myClock = 11:31:48AM, newMinutes = 31

myClock = 11:52:48AM, newMinutes = 31

4-7: myClock = 11:25:48AM, am = true

4-8: boolean because it either is or isn't AM

4-9: It assigns isAM with the value of AM

4-10: The same as its incoming value

4-11: `int minutes()`

Reports `this.minutes`

Returns:

`this.minutes`

Ensures:

`minutes = this.minutes`

4-12: `int seconds()`

Reports `this.seconds`

Returns:

`this.seconds`

Ensures:

seconds = this.seconds

4-13:
```
transferMinutes = myClock.minutes();
                  myClock = (8,2,43,true)
                  yourClock = (11,18,6,false)
                  transferMinutes = 2
yourClock.setMinutes(transferMinutes);
                  myClock = (8,2,43,true)
                  yourClock = (11,2,6,false)
                  transferMinutes = 2
```