Review Questions

1. You can define variables of different types in Java. List at least three types

   We discussed five data types: `int` (for integers), `double` (for decimals), `boolean` (for either "true" or "false"), `char` (for single characters), and `String` (for lists of characters). In fact, we also mentioned "`float`" which stands for a 'short' decimal. The most common ones are `int`, `double`, and `String`.

2. What are the results of the following computations for z1, z2, z3, and z4

```
int x = 10;
int y = 6;

int z1 = x * y;
int z2 = x + y;
int z3 = x / y;
int z4 = x % y;
```

   z1 and z2 are pretty obvious. z1 is 60 and z2 is 16. You would think that z3 is 1.6666 because that's what 10 / 6 works out to be on a calculator, but in fact z3 is 1. That is because x and y are integers, so their division will be an integer, too, and it is simply the "actual" number with all decimals chopped off. z4 gives you "the remainder after integer divison of x by y. But since 10 / 6 = 1 remainder 4, that remainder is the answer for x % y. (So just for practice, 12 % 5 = 2, 14 % 2 = 0, 14 % 3 = 2, 14 % 5 = 4, and 14 % 7 = 0)

3. What are the advantages of defining and using *functions* in your program? What is a *function header*?

   The function header is the first line of a function, where you define its name, its input (if any), and its output types. It basically tells you how to use a function: call its name, provide values to the input variables, and wait for the result of the function. The function body, on the other hand, defines exactly what the function does and how it does it.
   Functions are useful because they group together some code into subtasks that are easier to program and can be reused as many times as you wish. They are used to break up a complex task into many simpler subtasks.
   Functions, once defined, are pretty easy to use because all you need to know is the function header to use it. But creating your own functions is much harder, since you need to figure out which aspects of your overall task should be grouped into a function, exactly what should be the input and output for your function, and then of course you need to write the code to get the function to work properly.

4. What are the names of *all* functions defined in the program below, and what input do they require:

```
public class WhoIsWho
{
   public static void driveForward(int distance)
   {  // code deleted
   }

   public static void tunBy(int angle, double speed)
    { // code deleted
   }

   public static void main(String args[])
   {  // code deleted
   }
}
```

There are three functions in the above little program. The first one is called “driveForward” and takes as input an integer, the second one is called “tunBy” (which is a typo and should have read “turnBy”) and takes as input one integer and then one decimal. The last function is our familiar “main” function, which is where every Java program starts. It takes as input a String (or more precisely because of the brackets [] an “array” of Strings – we did not talk about arrays yet).

5. What are the benefits of a 2 wheeled differential drive robot over a 4-wheeled car-like robot?

   A 2-wheeled diff. drive robot can turn in place, a 4-wheeled car has a non-zero turning radius. The diff. drive robot is much easier to model.

6. If a wheel with radius 4 cm rotates 180 degrees, what distance will it cover? Note that 180 degrees are π radians. How many degrees does that wheel have to rotate to cover 25 cm?

   The “arc length” of a segment of a circle with radius R and angle t (in radians) is t*R. For example, to go once around a circle of radius R, your angle is 2*Pi, so that the length of that circle is our familiar 2 * Pi * R. So, to figure out how much distance a rotation of 180 degrees would lead, we first convert 180 degrees into radians: degree in radians = Pi / 180 * (degrees), so that in our case: angle = Pi/180 * 180 = Pi, so that 180 degrees will get us a distance of 4 cm (Pi * R). To figure out which rotation will get us a distance of 25 cm, we first figure out the angle of rotation in radians: 25 = R * t (where R is the radius and t is the angle). Since R = 4, we have that t = 25/4 = 6.25 radians. Now we need to convert that into degrees (we can see right away that since Pi = 3.1415 we have that 2 * Pi = 6.283 so that our angle of 6.25 radians is just about “once around”, so about 360 degrees. Precisely we have: (angle in degree) = 180 / Pi * (angle in radians) so that the angle in degrees to move 25 cm in distance is 6.25 * 180 / Pi = 358.1 degrees

7. Suppose a differential drive robot has the *left* motor attached to motor port A and the *right* motor attached to port C. Describe the way the robot moves if you issue the commands:

```
        Motor.A.rotate(-720, true);
        Motor.C.rotate(720, false);
```

The left motor rotates backwards while the right one rotates forward. That will turn the robot **left** in place.

8. What are “state variables, and where in a program are they defined?

   State variables are variables that specify certain properties of the entire robot, things that define a robot. For a diff. drive robot it would be the radius of the wheels and the length of the axles (there might be more). They are defined at the top of the program, before you define any functions, and are valid throughout the entire program. They are somewhat similar to named constants, but they are specific to the type of robot you are defining. For example, the value of Pi would be defined as a constant (since it is universally true) while the radius of the wheels applies only to the particular type of robot you are creating and should be stored in a state variable.