Problem: Program the computer to make a Turtle draw a circular arc.

Q1: Which arc?

- Turtle's heading in degrees \( H \)
- Radius of the arc \( R \)
- Angle amount of arc \( AA \)
- \((x_{\text{start}}, y_{\text{start}})\)
public void drawArc(
    double R, double AA
) {
    double H = this.getHeading();
    double Xstart = this.getX();
    double Ystart = this.getY();
Q2. How will another programmer use my code? How do I start?

/*new method: in Turtle.java*/
public void drawArc(  
  double R, double AA )
{
  double H = this.getHeading();
  double Xstart = this.getXPos();
  double Ystart = this.getYPos();

  System.out.println("Trying to draw an Arc.");
  System.out.println("Xstart=", + Xstart);
  //etc
Turtle's heading in degrees $H$

Radius of the arc $R$

Angle amount of arc $AA$

When $AA$ is expressed in radians

Arc length is $AA \times R$
When AA is expressed in radians, one length is \( AA \times R \).

Q: Convert AA degrees to radians?

\[
\text{Circumference} = 2\pi R.
\]

\[
360^\circ \text{ is } 2\pi \text{ radians. a whole circle 'round!}
\]

\[
\frac{AA}{360^\circ} = \frac{AA \text{ Radians}}{2\pi}
\]
Also code a call in main so we can test our work and see the results!

```java
public static void main(String[] args) {
    World = new World();
    Turtle mt = new Turtle(w);
    mt.turn(30.0);
    mt.drawArc(100.0, 50.0);
}
```
Q: Convert AA degrees to radians?

A: 360° is \(2\pi\) radians, a whole circle 'round!

\[
\frac{AA}{360^\circ} = \frac{AARadians}{2\pi}
\]

\[
\frac{2\pi AA}{360} = AARadians
\]

Computer: Please change your value of AA to express the angle in radians.

\[
AA = 2.0*\text{Math.PI}*AA/360.0;
\]
When $\text{AA}$ is expressed in radians, one length is $\text{AA} \times R$.

Q: Convert $\text{AA}$ degrees to radians?

Let's settle for an approximate solution: make the turtle go forward $R \times \text{AA}$ steps.

```python
this.forward((int)((R*AA)));
```
Let's settle for an approximate sol'n:

Turtle go forward \( R \times AA \) steps!

```java
this.forward((int)(R*AA));
```

That line's too skinny to see — make it fatter, please.

```java
public static void main(String[] a) {
    World = new World();
    Turtle mt = new Turtle(w);
    mt.setPenWidth( 5 );
    mt.turn(30.0);
    mt.drawArc(100.0, 50.0);
}
```
HA!? That's cheating. I want a round, curved arc rat a stupid line!
But at least it's a start.

Not good

Maybe?

Looking better!
Break arc into 4 (say) parts

$\text{each } \text{length} = \frac{\text{length}}{4}$

$\text{each } \text{AA} = \frac{\text{AA}}{4}$

$R$

Radius

Circle center
int nParts = 4; // for now???

double eachLength = AA*R/nParts;
double eachAA = AA/nParts;

for(int i=0; i<nParts; i=i+1)
{ this.forward((int)eachLength); }
int nParts = 4;  // for now??
double eachLength = AA*R/nParts;
double eachAA = AA/nParts;
for(int i=0; i<nParts; i=i+1)
{
    this.forward((int)eachLength);
}

Bug it still just draws one line!
for(int i=0; i<nParts; i++)
{
    this.forward((int)eachLength);
}

This loop is hard to understand.

Act like a computer to get a clue!

i = 0
this.forward( some distance );

i = 1
this.forward( some distance );

i = 2
this.forward( some distance );
for(int i=0; i<nParts; i=i+1) {
    this.forward((int)eachLength);
}

this.forward( some distance );
this.forward( some distance );
this.forward( some distance );
this.forward( some distance );

4 < nParts
\[ 4 < n_{\text{Parts}} \]

\[ \text{Strictly less than} \]

\text{iClicker}

(A) true (Run body again)
(B) false (Finish the loop again)
```javascript
this.forward( some distance );
this.forward( some distance );
this.forward( some distance );
this.forward( some distance );
```

What does the stupid Turtles do?

Not what I want!
for(int i=0; i<nParts; i=i+1)
{
    this.forward((int)eachLength);
}

for(int i=0; i<nParts; i=i+1)
{
    this.forward((int)eachLength);
    this.turn(360.0/(2.0*Math.PI)*eachAA);
}

/* Turn wants degrees! */