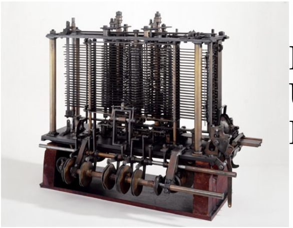
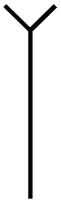
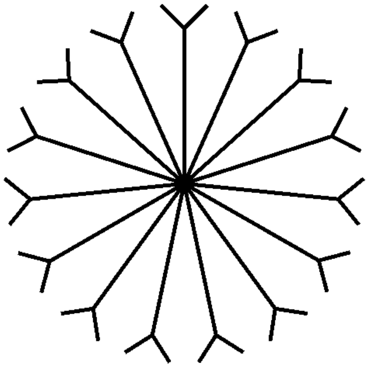


Lab 06
Introduction to Computer Science
UAlbany, College of Computing and Information
ICSI 201
Fall
2012





Programming questions will ask you to write METHODS to make the Turtle graphics shown here. Also, to make and use PARAMETRIZED methods, and start PART 2 of the current project, Project 03.

INSTRUCTIONS: THERE ARE 3 parts.

1. The TA will lead a discussion of **what** the last two exam. questions **asked** and guide you to do them on the computer. The framework:
public class Turtle extends SimpleTurtle

```
{//Lots of GE code omitted
 public void drawFork( )
 {  this.forward( 200 ) ;  //You code the rest
 }
 public void aNameStudentMadeUp( int nForks )
 { System.out.println(
     "Hey, I'm going to draw a number of forks on your World.");
   System.out.println( nForks );
   //You code a loop
   { //Somewhere in the body, there is the CALL
     this.drawFork()
     //code to update the count should be AFTER the fork drawing
   } //end of the loop
 } //end of the aNameStudentMadeUp method
 public static void main(String[] a)
 {
   Turtle tu = new Turtle( new World());
   //code a loop to the run the body below 15 times:
   {
      tu.drawFork( );
      tu.turn( 24.0 );
   }
} //end of the definition of the Turtle class, end of Turtle.java
file.
```

2. If you upload a PERFECT SOLUTION to BOTH the last 2 midterm questions THAT IS COMMENTED with CORRECT (metacognitive) EXPLANATIONS OF everything you misapprehended about the two questions when you took the exam, you will recoup up to half (23.5 points total!) the points for those questions on your midterm grade! (Usual due date for lab followups).

3. Get started w/ Proj03:
   - Create a **Proj03** dir/folder.
   - Set up the 4 subdirs. for the 4 parts.
   - Set up a 5th subdir. for **Lab05**.
   - In **Lab06**, set up sub-subdirs **Start** and **Release**.
   - Get a copy of GE's **Picture.java** file into **Proj03/Lab06/Start**
   - Code in **Picture.java** a **scribble** method that USES your **drawFork** method. ALSO, code a **main** method that tests it! Test and debug. The first version should ignore the scale parameter. When it works, develop a new version that actually uses the scale parameter.
   - Your actual Project 03 work must draw something DIFFERENT from my 2 prong fork!
   - New policy for followup: The sooner you submit it after your lab, the more credit!!! (14 points per day.)

Reprinting of revised questions below:

1. (A1 8) Make your own **Turtle** method for drawing a 200 pix. tall two-prong fork (or snake's tongue), similar to what's on the cover of this exam. (A2 8) Make sure that before returning, your method makes **this** (a **Turtle**) go back to its original location and head pointing direction.

   Your method should begin:

```
 public void drawFork( )
{ this.forward( 200 );  //You write the rest!
}
```

   (B 8) Code a main method that makes a **World**, then a **Turtle** in that **World**, and then uses a loop to draw a circle of 15 forks **using the method you have made.** (360 degrees/15=24 degrees)

2. Assume you solved the previous problem correctly. (6) Make a parametrized **Turtle** method with one parameter **int nForks**. When called, (A 3) it should **first** print "**Hey, I'm going to draw a number of forks on your World.**" (B 3) **Second** it should print what number of forks it should draw. (C 11) **Third**, it should use your **drawFork** method in a loop to make **this** (a **Turtle**) draw a circle of forks with **nForks** (number of) forks spaced at equal angles. Strategy hint: Program it to divide (with **/**)360.0 degrees by **nForks** to calculate how many degrees the **Turtle** should turn after each arrow.

   Notes: (1) Do NOT recode the body of **dofork()** again.

   (2) Do NOT code another **main( )** method! Only code a new method that gives **Turtle** the capability to draw a circle of ANY, unpredictable number of forks. The number is determined by the caller, NOT the callee code. You write the callee code.