Lecture 03
Learning to Program

- One of the best (maybe the best) way to learn to program starts with downloading source files of software.
- But: It is NOT just compiling and playing with the software.
- It is: Figure out how the program works LINE BY LINE.
- See my scribblings on the next slide!
Lecture 03

• End of Lecture 02: Sketch in NUMBERED STEPS how the Turtle draws the B on Project 1 assignment sheet.

• Want to see that again?
public class PrintAndTurtleDemo{
    public static void main(String[] a){
        System.out.println("Prof. S. Chaiken");
        System.out.println("sdc");
        System.out.println("348F05BE");
        World w = new World();
        Turtle tu = new Turtle(w);
        tu.forward(-30);
        tu.forward(60);
        tu.turn(120);
        tu.forward(30);
        tu.turn(120);
        tu.forward(30);
        tu.turn(-120);
        tu.forward(40);
        tu.turn(120);
        tu.forward(40);
        tu.turn(-150);
        tu.penUp();
        tu.forward(50);
        tu.penDown();
        tu.turn(-90);
        tu.forward(60);
        tu.turn(90);
        tu.forward(30);
        tu.turn(180);
        tu.forward(30);
        tu.turn(-90);
        tu.forward(30);
        tu.turn(-90);
        tu.forward(15);
        tu.turn(180);
        tu.forward(15);
        tu.turn(-90);
        tu.forward(30);
        tu.turn(-90);
        tu.forward(30);
    }
}

1. Your figures (like my B) don't have to be perfect, just recognizable.
I invented a visual language for the path of a Turtle when the Turtle's pen is up (so it doesn't draw anything):
DOTTED LINE
Lecture 03

- Somebody attempted to draw a golf club and go back to where he started:
  
  tu.forward(100); tu.turn(30); tu.forward(20);
  tu.forward(-100); tu.turn(-30); tu.forward(-20);

- His idea was to do the reverse of each step.
tu.forward(100);
tu.turn(30);
tu.forward(20);
tu.forward(-100);
tu.turn(-30);
tu.forward(-20);

Sketch the action on paper!!!
Do it now! Really!!
Clicker Question?
The (stupid) computer does the drawing steps in the order it is programmed to!

tu.forward(100);
tu.turn(30);
tu.forward(20);
tu.forward(-100);
tu.turn(-30);
tu.forward(-20);
NOT what you or other thinking, loving, intelligent person WANTS or INTENDS the computer to do!
Smart idea: Reverse the steps!

- After Turtle tu is commanded to go forward by `tu.forward(100);`
  
  We can make reverse its motion with `tu.forward(-100);`

- Similarly, `tu.turn(30);` is reversed with `tu.turn(-30);`

- BUT: Reverse a series of moves by writing the reverse of each **IN REVERSE ORDER**!
  
  `tu.forward(100);
tu.turn(30);
tu.turn(-30);
tu.forward(-100);`
Lecture 03

• DONE:
  - Created 9 lines of code to draw one club; turn $45^\circ$
  - Two golf clubs drawn by two separate copies of the same code. See that code under Lect02.
  - A loop that made one copy of golf club code be run 7 times, to draw 7 golf clubs.
  - The loop used a variable to count how many clubs remain to be drawn. Code subtracted 1; made it get “smaller and smaller” (as that kid said) AFTER EACH club was drawn.
Lecture 03

• DONE:
  – Two copies of the same code.
  – A loop that made one copy of code run 7 times.
  – A variable to count how many clubs remain to be drawn. Code subtracted 1; made it get “smaller and smaller” (as that kid said).

• Today:
  – Explain (a) a variable and (b) a loop more clearly with a while loop (instead of the for loop)
  – Use the VALUE OF A VARIABLE to specify the weight of the head.
Lecture 03

• Today:
  - Explain (a) a variable and (b) a loop more clearly with a while loop (instead of the for loop)
  - Use the VALUE OF A VARIABLE to specify the weight of the head.

• Lab 02 and next week's lecture:
  - Make a method (“subprocedure”) to draw one club.
  - Make it parametrized by weight.
VARIABLE

- NOT the same as in math. Not like x in a $x^2 + 2x + 1 = 0$ solve for x problem!

- `int numClubsLeft;` means “Get a dry-erase board named numClubsLeft”

- `numClubsLeft = 7;` means “Write a 7 on the board so everyone can see it”

- `(numClubsLeft > 0)` means “Look at the number written on the board and answer whether or not it is > 0”
Our program

- Look at this lecture's program on the web and study it!
- The number written and visible on the dry-erase board named numClubsLeft is used for everyone to know “how many more golf clubs are left for the Turtle to draw”
- We want 7 clubs; so we start with the count of 7 and subtract one after drawing each club.
- We should draw another club while (as long as) that count remains > 0.
- We then made the head width be set from that number, so the head widths varied.
numClubsLeft = numClubsLeft - 1;

- This is false (or nonsense) in math!
- It means: "Look at the number written on the dry-erase board. Tell somebody to subtract 1 from it and tell you the result. Then, overwrite the number on the dry-erase board with that result."
- That's what's abbreviated by = in Java! (A mouthful.)