Lecture 02

- Video: Turtle Object, PROCEDURES to help draw a dinosaur.
- Plan of next few weeks.
- Reading guide.
- iClicker quiz
- Figuring out how my example program worked.
- What we did.
-

Chief Learning Goal

Given a description or your own dream of what a computer might do, YOU WRITE/CREATE a program that demonstrably makes the computer do it.

Surprising Examples

- http://www.youtube.com/watch?v=xMzojQFyMo0
- After about 1 hr of play, I discovered that each turn was a little under or over 120 degrees, and three successive turns were a little different, makes a cool figure. I think they increased length a little after each line.
- Get at my Java program, my try to reproduce this, from the Wed. It has a **loop**.

Surprising Examples

http://www.youtube.com/watch?v=xMzojQFyMo0

- The kid talked about “SUB-PROCEDURES” with “INPUTS” to draw
  1)The back of the dinosaur.
  2)The fuzzy spikes coming out of the back
- Circle of Golf Clubs Demo
  - You'll write programs LIKE THIS and beyond for your midterm exam!!! (October 10).
- Java-lingo for “procedure” is **method**

Plan...

- I'll demonstrate

  figuring out

  HOW the code on the Proj 1 handout

  MAKES

  a Turtle DRAW a B

- YOU do that for the E

  so you will understand

  how to do the homework!

Needed materials and “habits”

- iClicker & textbook (used/shared OK)
- **(BY MONDAY!) Way [WORKFLOW] to use computers for lab follow up homework, experimentation, project assignments and submit work to Blackboard.**

Project Assignments Soon

- Proj1 (NOW) & this week's Lab: you program ONLY a sequence of method calls.
  - LAST TIME it will be that simple!
- Proj2 (due Sept 19??) MUST
  - create YOUR OWN methods (not just use given ones)
  - your own methods MUST have parameter
  - you program calling your own methods.
- Labs 2, 3 & Proj2: Program calculations with variables.

Intro. reading:

The whole preface, INCLUDING the material for teachers.

explains/motivates content and approach—what you might get out of the course, the book and why.

But you will not USE that content directly.

SO: START PROJ. 1 FIRST!!

1. We and the book try to explain “concepts” just when they are needed, not bundle the concepts first, then apply them together later.
2. Arrays are covered VERY EARLY (Ch. 3)

Project 1 reference reading

See the pages listed in the assignment!!
You MUST USE some of the
DETAILS
written on those pages, and given in lectures too.

Spelling and punctuation
REALLY REALLY MATTER!

`tu.turn(90);` NOT **`tu.Turn( 90 ).`**

iClicker Question
What best describes Profs. Guzdial's and Ericson's approach?

A) One computing topic at a time is taught in great detail in each chapter, separately.

B) Hard concepts are introduced only AS NEEDED for problems that students find interesting to solve.

Turtle Drawing: How to figure out the example in the homework assignment.

- Number each method call.
- Mark and identify by number

WHICH METHOD CALLS

MADE THE TURTLE

DRAW **WHICH** LINES OF THE B

- I will scan and post my pencil scratchings!
- YOU do the same for the E--BY YOURSELF!!!

What is the best way to learn from a worked example (like my B E drawing code)

A) Figure out, MENTALLY, what it does, by concentrating ONE LINE OR COMMAND AT A TIME, together with notes you wrote about the results you already figured out from the previous lines or commands.

B)Carefully type it in, compile, fix syntax errors due to mistyping, and then run it, to see it do what it does.

Making programs work yourself

- Get (download/install) the Java Development Kit (if your computer doesn't have it already)
  - Lab, Library computers and Macs have it already
- Get (download) the DrJava (Beta version!) Software (if your computer doesn't have it already)
  - Lab and Library computers have it.
- Get the “Book Classes”
  - In Lab you will get them onto your UA “C:” drive.
  - On your computer, you download them.
- Set DrJava's CLASSPATH to your Book Classes folder. (Instructions in Lab and the text.)

Testing your setup

- DrJava for generic Java
  - Launch Drjava
  - Type and save a Hello program in Definitions pane.
  - Click Compile and then Run
  - All is well if you see the Hello printed.
- DrJava for the Book Classes
  - Make sure the CLASS PATH is set.
  - Add to your Hello program right under the println method call, Compile, Run

```
World w = new World( );
```

  - All is well if there are no error messages and you see a new window appear.

Points to remember

1) A program is a written SEQUENCE of actions the computer performs exactly in the ORDER they are written.

2)The actions (so far) result from CALLING

(a kind of commanding, invoking, making-happen)

a named potential behavior (a METHOD) FOR or ON a particular OBJECT, where that object can DO that behavior.

```
tu.turn(180);
```

Points to remember

1) program... written SEQUENCE of actions ...

2)The actions result from CALLING

(a kind of commanding, invoking, making-happen)

a named potential behavior (a METHOD) FOR or ON a particular OBJECT that can DO it.

```
tu.turn(180);
```

The Turtle object is referred to by **`tu`**

The method name is **`turn`**

A Turtle can turn.

It can turn a given **parameter** number of degrees.

The Albany Way
(NOT Georgia Tech!)

- All Java code you write is to be PART OF A
  - Complete Java Application Program..
  - that you can show to someone else...
  - can have that person RUN your program...
  - and everyone can **SEE AND DESCRIBE**

WHAT IT MAKES THE COMPUTER DO

- **(Do NOT, by hand, command the computer by typing Java into the Interactions Pane...your work is lost and that's not programming!!)**

What's the Albany Way?

To see, by experiment, what a line of Java does, we:

(A) Edit the main method of our Hello program with the definitions pane, save, compile and run.

(B) Type it in DrJava's interaction pane and press ENTER.

Next Class

- A more fundamental view of a programmed computer...
- It IS a human or robotic person or agent...
- that follows a recipe of instructions...
- one step at a time...
- to copy, calculate, erase and rewrite...
- data (ultimately numbers in binary)...
- WRITTEN ON PAPER LEDGER SHEETS or...
- stored in (today's) mostly silicon memory chips.

What our program does

- Make a new **`World`** object
- Use variable `w` to refer to it.
- Make a new **`Turtle`** object in the **`World`** referred to by **`w`**. Use variable **`tu`** to refer to it.
- Run, in the programmed sequence, dozens of method calls ON that same **`Turtle`** object, which make that **`Turtle`** draw stuff by moving and turning.
  - Those method calls have the **syntax**:

`tu.turn(` some number **`);`** or
`tu.forward(` some number **`);`**

What is "**`Turtle`**" ? Hint: Think of how we used the word **`Turtle`**.

A) **`Turtle`** is the **kind or species of objects** the program made one copy of. By referring to that **`Turtle`** by **`tu,`** the program commanded it to draw stuff by moving and turning.

B) **`Turtle`** is **one particular object** capable of drawing on command.

- Make **a** new **`World`** object
- Use variable `w` to refer to it.

- Make a new **`Turtle`** object in the **`World`** referred to by **`w`**. Use variable **`tu`** to refer to it.
- Run, in the programmed sequence, dozens of method calls ON **that** same **`Turtle`** object, which make **that** **`Turtle`** draw stuff by moving and turning.
  - Those method calls have the **syntax**:

`tu.turn(` some number **`);`** or
`tu.forward(` some number **`);`**

What is "**`Turtle`**" ? Hint: Think of how we used the word **`Turtle`**.

A) **`Turtle`** is the **kind or species of objects** the program made one copy of. By referring to that **`Turtle`** by **`tu,`** the program commanded it to draw stuff by moving and turning.

B) **`Turtle`** is **one particular object** capable of drawing on command.

Object Oriented Programming 1

- Potential actions a computer might do...
- are METHODS that belong to an OBJECT...
- think: an OBJECT, like a real live turtle animal,
- has potential BEHAVIORS...like walk forward and turn...
- we programmers can program it (the poor turtle) to DO for us.
- We program the computer to draw a line...
- by programming one Turtle to move forward.
- Our program CALLS that Turtle's forward METHOD.

What is "`Turtle`" ? Hint: Think of how we used the word `Turtle`.

A) `Turtle` is the **kind or species** of **objects** the program made one copy of. **class**

By referring to that `Turtle` by `tu,` the program commanded it to draw stuff by moving and turning. **A program could have made MANY DIFFERENT Turtles, of the same KIND.**

B) `Turtle` is **one particular object** capable of drawing on command. **NO!**

Needed materials

- Calculator: ONLY SOMETIMES (usually not)
- BRING IN A SIMPLE CALCULATOR ON WEDNESDAY
- So you can imitate the Rosie Computers of WW II.