An introduction to programming concepts with Scratch
(in 5 hours + homework)
Who computes?

- Scientists
- Engineers
- Businessmen
- Social scientists
- Artists
- FBI agents
- Brain surgeons
- Gamers
- Grandparents

Everyone uses information in some way.
Computers do information processing.
What can we do with Scratch?

• Learn some computing concepts.
• Learn some practical algorithms.
• Use Scratch as computing tool.
• Have fun with Scratch creating stories, games, art.
Part I: Looks and Motion

• Goals:
  – Learn Scratch programming environment
  – Learn the looks and motion menus
  – Write a sequence of instructions
Start scratch and let’s go!

• Click on the cat icon
• Scratch programming environment comes up
• We will first do simple things
Our Scratch environment

- **Stage** is at upper right (where actors act and information is displayed)
- **Sprite** is another name for actor.
- Instruction menus/panel at left (instructions for the actors)
- A **script** is another name for program or method; a script tells the actor what to do.
- Programming area in center; here is where we construct scripts for the sprites
We start with the cat sprite

We will learn to use many other sprites later. For now we use the cat as our only actor. We script the BEHAVIOR of our cat.
The LOOKS menu

Has instructions for setting the color, size, and visibility of a sprite. Costumes will be used later.
The “hello” script (program)

• Choose the Looks menu
• Click on the “say hello” lego block
• Check your sprite behavior at the right
• Then click “say hello for 2 secs”

Your very first Scratch program!
Try some other looks operations (click on menu items)

• Change color effect by 25
• Change color effect by 25 again
• Hide
• Show
• Change size by 10
• Change size by 10 again
• Set size to 100%
Let’s write a script to

• Say “hello” for 2 seconds
• Then change color by 25
• Then think “Hmm..” for 4 seconds
• Then change color by 75
• Then change size by 200

Drag each instruction from the menu to the center script area. Connect them into a single block. Edit the parameters to get the numbers we want.
Our script (program)

- say **Hello!** for 2 secs
- change **color** effect by 25
- think **Hmm...** for 4 secs
- change **color** effect by 75
- change size by 200
A sequence or block is a simple script or program

- The first instruction is done first
- The second instruction is done second
- The last instruction is done last.
- (if any one instruction is done, then every one of them is done)
The WAIT instruction

• WAIT is needed to slow down the acting so we can see or hear it properly (computers are too fast sometimes)

• Get the wait instruction from the CONTROL menu. Ignore the other menu options for now.

• Insert a wait in our looks script
3 second pause between changing color and size
Student exercise: write a script to do the following

• Double the size of the sprite
• Wait 2 seconds
• Change the color of the sprite to green
• Wait 4 seconds
• Change the whirl effect to 100
• Say “That’s all folks!”
The MOTION menu

How to locate and orient a sprite; moving a sprite.
Position on the stage

• Using the **Looks menu**, shrink our cat to 25%.
• Click on the **Motion menu**.
• Click to check the box for x-position and y-position
• Drag your cat around and note its x-y position.
Exercises: goto instruction

• In the Motion menu, drag the “goto xy” instruction to the script panel.
• Edit the coordinates and click to see the sprite’s position
• A) goto x=200, y=0
• B) goto x=-200, y=0
• C) goto x=200, y=-100

Where does the cat go?
The “glide-to” instruction

• Drag the “glide-to” instruction into your script panel.
• Edit the coordinate values and click to see where your sprites goes.
Exercise

• Create a script to glide the sprite along the sides of a triangle. The first vertex of the triangle is (-100, -100). The second vertex is (200, -100). The third vertex is (50, 100). Make sure you complete the triangle.

• Change the speed of gliding and run again.
Angles and directions

Making your sprite go this way and that way
Exercise Script

• Start at (-100, 100)
• Move 200 steps
• Turn right 90 degrees
• Move 200 steps
• Turn right 90 degrees
• Move 200 steps
• Turn right 90 degrees
• Move 200 steps

Use waits if needed to make the motion look smooth
Exercise/Homework Script

• Sprite starts at home base at x=-100; y=-50
• Sprite says “I hit the ball” for 2 seconds
• Sprite runs (east) 200 steps to 1st base
• Sprite changes color
• Sprite runs 200 steps to 2nd base (north: left turn from 1st base)
• Sprite doubles size
• Sprite says “I hit a double” for 5 seconds
End of Part I: Outcomes

Student should be able to control the looks of a sprite and its location on the stage. Students should be able to construct sequences of instructions controlling looks and location on the stage.
Part 2: Repetition and Variation

• Goals:
  – Learn how to program with repetition
  – Learn how to change costumes
  – Learn to control execution (behavior) using special keys
The sneezing cat script

• We repeat 20 times

  Say “AAAAHHHHH” for a short time

  Then grow the cat 5 bigger
Exercise: bring the cat size down to normal

Repeat 20 times

Briefly say “CCHHEEEEW”

Then shrink the cat’s size by 5

Repeat N times and repeat forever are in the Control menu
Exercise: Have the cat do a flip by rotating 20 degrees 18 times.

• Repeat from the Control menu: set the number of repetitions to 18
• Rotate from the Motion menu: set the size of each small rotation to 20 degrees
Changing costumes

Another way to change the appearance of a sprite.
Making a new costume to animate the bat

1) Get bat2-a sprite from the “sprite box”.
2) Click on sprite
3) Click on “Costumes”
4) Click on “Import”
5) Now get bat-2b sprite from the sprite box
Use a loop to make the bat fly!

1) **When** and **repeat** from Control menu and 2) next costume from Looks menu.
Changing coordinates

• We can *randomly* set the location of the bat so it will “flutter”.
• Set X to a random number
• Set Y to a random number
• Move the bat to location (X,Y)
• Of course, the bat should remain on stage!
Using *pick random*

Click the stop sign at the stage upper right to stop the forever loop.

Do we need a wait in the loop? Try it to see the change in flying.
Controlling the bat’s direction

- Let’s create **multiple scripts**.
- Click space bar for random moves
- Click right arrow to move right
- Click left arrow to move left
- The bat will behave differently depending upon which key is typed!
- (So, could a gamer catch the bat?)
Multiple interactive bat flight

3 different behaviors for the bat
Adding a **background**

- Click on the **Stage** icon at lower right
- Click on **Backgrounds**
- Click on bat icon (Sprite1)
- Click on **Scripts**
- FLY AGAIN!
Homework/Exercise:

• Add a behavior (script) to move the bat up with the up arrow.
• Add a behavior to move the bat down with the down arrow.
End of Part 2: outcomes

Student should be able to write a program controlled by a loop, and execute a script by clicking special keys. The student should also know how to use “costumes” to change the appearance of a sprite.
Part 3: Input/Output, Variables, Simple computing

• Goals:
  – Learn how the user and the program can communicate
  – Learn how to compute with formulas
  – Computing an average, perimeter of a geometric figure, etc
The SENSING menu

Getting information from the user or some other machine
Asking the age of the user

• Program wants age of user (maybe to set a level of difficulty in a game)
• Program asks user for age
• User types in age
• Program stores the answer in a variable named “answer”
What is your age?

• Depends on who you are
• For one person it’s 12
• For another person it’s 27
• For yet another, it’s 19
• IT’S VARIABLE!
Use the SENSING menu

1) Ask the user for age;
2) user types in age;
3) result stored in “answer”
User types “13” and the script stores the answer

“answer” is a Special Scratch variable
OUTPUT: Giving the user information with **say answer**

*Say* is in Looks menu; drag “answer” from Sensing menu; click
Using the Variables menu

We can save an answer in our own named variable;
Then we can program with many data items using many variables
Making an “age variable”

1) Click Variables menu; 2) click “Make a variable”
2) Type in “age” and click OK

Do you want to display for all sprites or just one?
We have an “age variable”

Operations for “age”

Displaying variable “age”
We can **join** text and a number to say things.

- **From Looks**
  - `say Hello!`

- **From Operators**
  - `join hello world`
  - `answer`

- **From Sensing**
  - `join answer is your age`

Build this instruction from the parts above it.

13 is your age.
Storing a value in a variable; then saying it to the user

1) **Set** from Variables menu; 2) **Say** from Looks menu; 3) **Join** from the Operators menu joins your text with the “answer”
Using **variables** in programs

- A script might have to remember information
- How far is the cat from the dog?
- How fast is the rocket moving?
- Where did the user click on the stage?
- What is my score in the game?
- What is the user’s skill level?
Computing \( C = A + B \) (1 of 3)

- Make 3 variables, A, B, and C
- Type in the values of A and B
- Set the value of C to the sum of A and B
Computing $C = A + B$ (2 of 3)

- Make 3 variables, A, B, and C
- Type in the values of A and B
- Set the value of C to the sum of A and B
Computing $C = A + B$ (3 of 3)

- Make 3 variables, A, B, and C
- Type in the values of A and B
- Set the value of C to the sum of A and B

Drag + operator from Operators menu
From Variables menu drag variables A and B into the slots
Algorithm for computing the average of two numbers

• Let the first number be A; say A=12
• And the second number be B; say B=15
• To compute the average _______________
• So the average of 12 and 15 = __________
Programming the average

• A is the first “variable”: A = _____
• B is the second “variable”: B = _____
• V is the third variable: V = _____
• Take the value of A, add the value of B, divide this sum by 2, then set the value of V to this answer.
• \[ V = \frac{(A+B)}{2} \] (how to do it in Scratch?)
Script from Variables and Operators menus

Drag the variable names into the slots of the operators

A formula from the Operators menu

Script not yet executed
After clicking to execute script

The variable V now stores the computed average of A and B
Exercise

• Put a Wait 5 secs in the script between the two set operations for variable V
• Execute the script and observe the changes in the value of V
Exercise

Change the values of A and B by editing the script (keep the Wait);

Input the age of grandchild and age of grandparent;

Click on your script to compute the average of the new A and B
Homework/Exercise: average program

- Modify the average program
- Ask the user for a value of A
- Ask the user for a value of B
- Compute V as the average
- Report the average to the user
- Run the program for different As and Bs
HW Exercises

• Average 3 numbers; say 12, 33, and 21
Computing perimeter of a rectangle: algorithm

- Ask the user for length L
- Ask the user for width W
- Compute $P = (L+W) \times 2$
- Report $P$ to the user

Example:

$W = 10$

$L = 15$

$P = (15+10) \times 2 = 50$
Review:
Getting length from user

- Ask “What’s the length” and wait
- Set length to answer
Review: Reporting the value of a variable to the user

- Say join “length =“ length for 2 sec

Looks menu
Operators menu
You type this in
Drag length variable from Variables menu
Review: Program with user input and output
HW Exercise: create the Scratch program we designed

- Ask the user for length L
- Ask the user for width W
- Compute $P = (L+W) \times 2$
- Report $P$ to the user

Show your program to your Grandparent.
End of Part 3: outcomes

Student should be able to write a program to ask the user for data, compute using the data, and report the answer to the user. Students should be able to compute using common formulas.
Part 4: working with sounds and sprite communication
Part 4 goals

Learn how sprites can communicate by sending messages
Learn how to play sounds and to input your own voice
Communicating sprites: taking turns by message passing

Get cat and dog sprites
Suppose cat speaks first
When finished, the cat can send a message to the dog to speak
When the dog is finished, the dog can send a message to the cat to speak
First behavior for the cat

When clicked, cat will:
1. Say: Hello Dog! for 2 secs
2. Broadcast: dog1 and wait

After saying hello, cat sends a message to dog (actually to all sprites)
Dog is activated by hidog1 message
Cat’s turn again after hicat2 message from the dog
Dog can have 2\textsuperscript{nd} touching behavior
Sprites can have many behaviors

Activated by clicking green flag
Activated by clicking the sprite
Activated by a key being pressed
Activated by a message from another sprite
Activated by touching another sprite
Using sounds

Sound library;
Speaking in the mike;
Importing music
Composing with drum beats
when I receive go
repeat 5
  repeat 3
    play drum 48 for 0.2 beats
  repeat 5
    wait 0.2 secs
    play drum 47 for 0.2 beats
Jodi jumps on the drum

when clicked
play sound recording1
say This is din! for 3 secs
wait 2 secs
broadcast goJodi

when I receive goJodi
broadcast godrum
repeat 110
wait 0.1 secs
next costume
Recording from a mike (or an iPod or other music player)

1. Plug in microphone
2. Pick the speaker sprite
3. Click "Sounds"
4. Click "Record"
5. Click red button and speak
6. Click square button to stop
7. Click triangle to hear it
8. Click "OK" if it's OK
9. Use name of sound in a play-sound instruction
The Friends Story (play it)
Variations in stories

Have several scenes
Scenes change with messages or variables
Make characters move
Make characters hide or show
Behavior can change with variables, touching, mousing.
Background music for each scene
Learning from examples

1. Click on the FILE tab
2. Click on the OPEN tab
3. Click on the Examples tab
4. Choose an example: for example, try STORIES and MAD LIB
End of Part 4: outcomes

Students should be able to program by passing messages. Students should be able to program their sprites to make sounds.