

# Problem Set 1

## *Starting From Scratch, part 2*

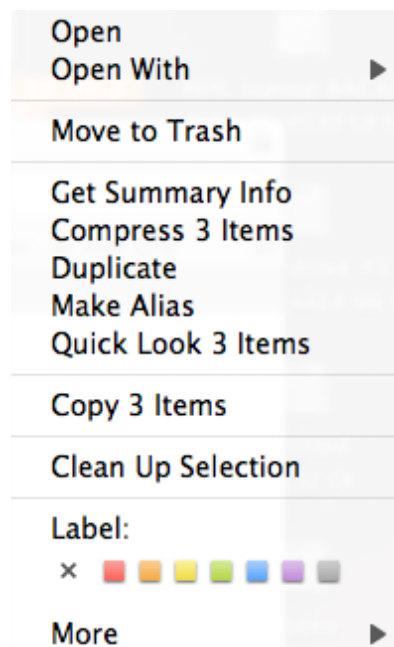
Everything must be completed and submitted prior to 5 PM on Monday, September 19.

You must submit your solution to all questions in this problem set by uploading a single “.zip” file to the “electronic dropbox” on the Computer Science E-50a course website. Note that any problem that poses one or more questions requires a written response, typed in a text file.

Here is some important information on how to accomplish electronic submission of your work.

*You should create a compressed, “.zip” archive of the files you created in solving all of the problems. Name this compressed file yourFASusername.zip*

*To create a .zip file is easy. On a Mac, select all of the files with your mouse so they are all highlighted at the same time. (In this example, suppose 3 files were selected.) Then hold down the “Control” key while clicking these files, and the following contextual menu will appear. Select the item that says “Compress 3 Items”, and a file named Archive.zip will be created. Rename it to yourFASusername.zip The menu should look something like this:*



It's just as easy to create the compressed file on Windows. Right-click the files after selecting them all, and the following contextual menu will appear:



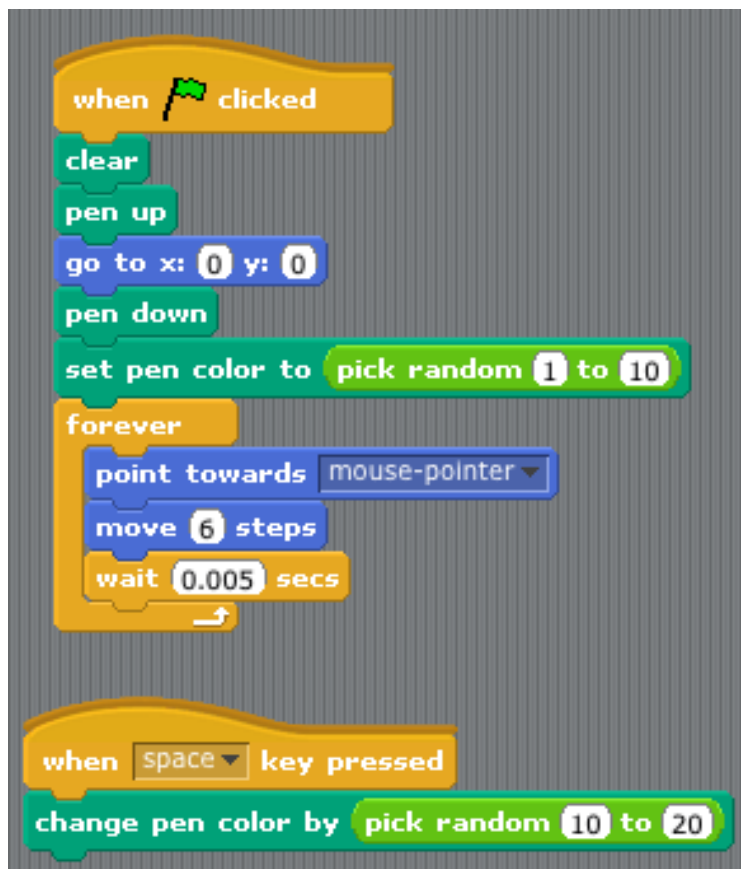
Choose the “Send To” option, and from that select “Compressed (zipped) Folder”. You will have to rename the created folder to **yourEASusername.zip**

Details on how to upload this file to <http://www.fas.harvard.edu/~libe50a> appear on this website, and will also be demonstrated in lecture and in section!

### Part A: Simple Exercises (25 points total)

1. (5 points)

Create a new *Scratch* project. Change the standard cat sprite into a “bee” (one of the standard sprites that comes with the *Scratch* software) and then construct the following two *Scratch* scripts for that sprite.

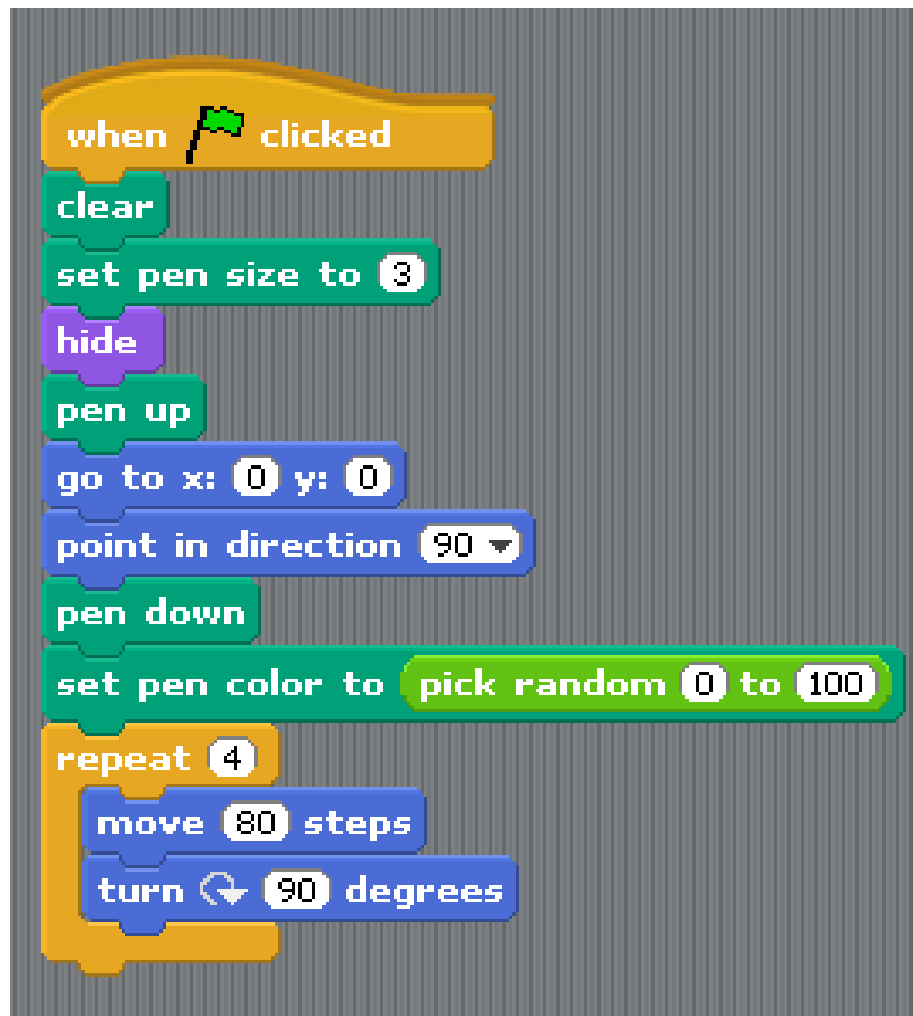


Now run the program by clicking the green flag. What happens when you move your mouse around on the stage while the program is running? While the program is executing, you should occasionally press the *space* key and watch what happens.

Finally, add some code to this project so that every time you press the letter “C” on the keyboard the stage will entirely get erased before the sprite starts drawing again. Save this project as **prob1.yourEASusername.sb**. (You *do* have an FAS account, don't you?)

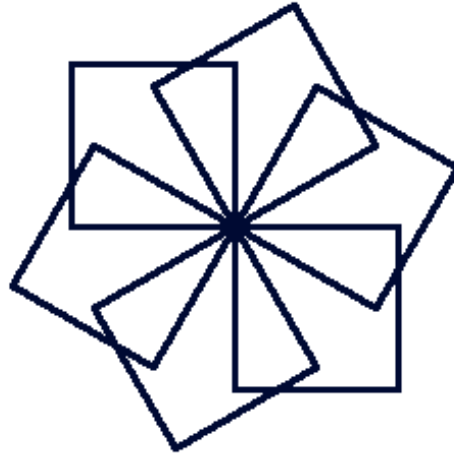
2. (10 points)

Create a new *Scratch* project, and construct the following program.



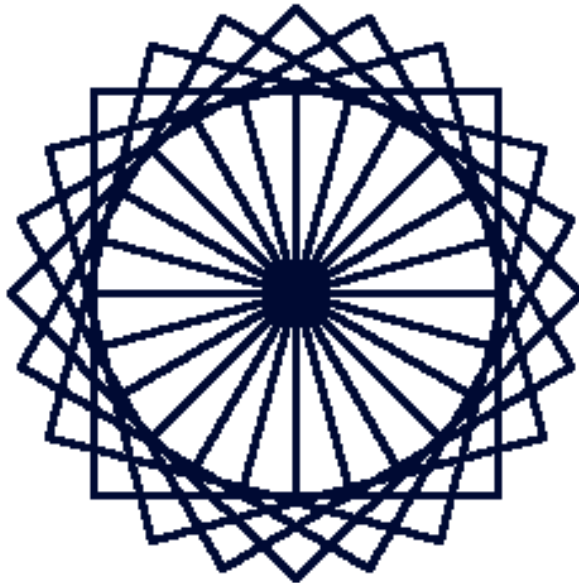
Before executing this project, try and predict what will happen.

Now modify this program so it can produce output that looks like this:



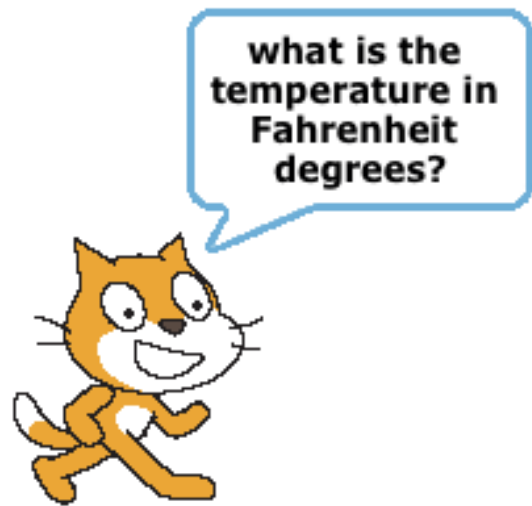
If you look carefully at this figure, the simple square shape that the original program would draw just once is now being drawn *six* times (each one rotated a bit more than the previous one), so we get a circular overall shape. The modification you need to make should be really simple: it will consist of adding a **repeat** loop which will enclose the existing **repeat** loop (in other words, you'll end up with a repeat loop inside of another one). In addition, you will need to add a “turn” command.

Here is what the modified program would produce if we simply increase the number of repetitions in the new **repeat** loop and correspondingly decrease the number of degrees that the sprite rotates before drawing another square:



(Hint: the number of repetitions multiplied by the number of degrees the sprite turns before the next repetition should equal 360 in order to accomplish the overall circular shape.) Save this project — with your solution to the figure at the top of this page only — as **prob2.yourEASusername.sb** .





212

If the user responded by typing **212** (as in the above example), the response that gets output would be this:



5. (15 points)



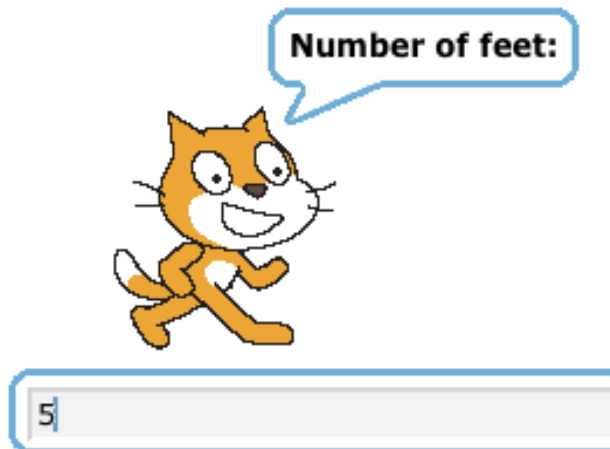
A man should weigh 106 pounds for the first 5 feet of height, plus 7 pounds for every inch above that; a woman should weigh 100 pounds for the first 5 feet of height, plus 6 pounds for every inch above that.

Write a program that determines how much an individual person should weigh.

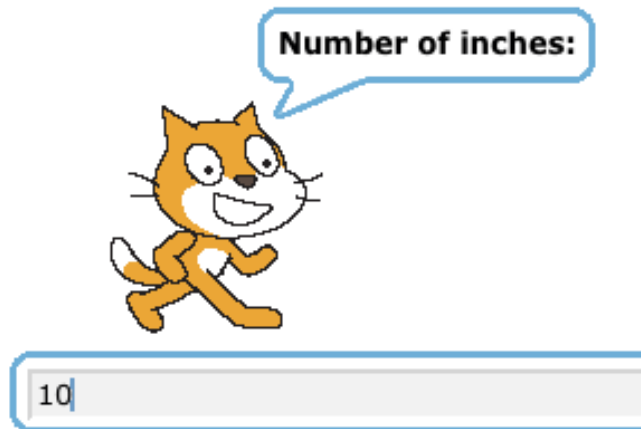
Here is an illustration of what your program might look like in action:



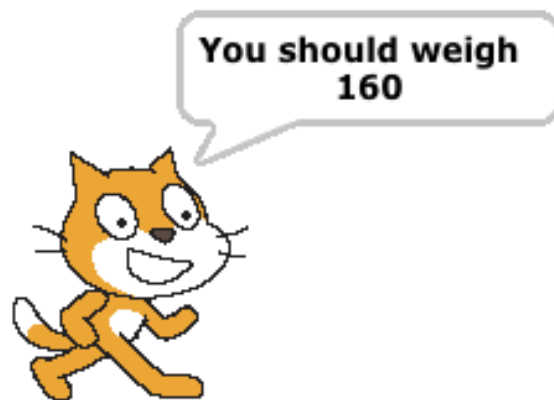
The sprite then says you need to input a *height*. First, in feet:



And then in inches:



And then the output appears:



If instead the user indicated he was “male” and was exactly **6 feet** tall, the output would be **190 pounds**.

If the height that gets input by the user is less than 5 feet, the sprite should complain that an illegal value has been input, and the program should stop at that point.

6. (15 points)

Go back to the original *Scratch* project featured in problem #2. Modify this project so that instead of drawing a single square, it will instead draw a row of 6 squares, spaced apart as shown below. You’ll need to adjust the starting (x,y) coordinate so that the pen begins drawing toward the top left of the stage.



Save this project in a file named **sixSquares.yourEASusername.sb**.

Now modify your program so that instead of 6 squares, you get 6 equilateral *triangles* (i.e., each triangle consists of sides that are equal in length). Each triangle will contain three 60-degree angles. Save this project under the file named **sixTriangles.yourEASusername.sb**.

Next, modify your program again so your program draws a row of objects containing some other number of sides (e.g., hexagons or pentagons). Save this project under the file named **sixOtherShapes.yourEASusername.sb**.

7. (10 points) *This problem is for graduate-credit students only!*

Undergraduate-credit students may attempt this problem for “extra credit.”

You are to modify your program yet again, so that it produces a row of six 5-point



“stars” like this:

Remember that the Scratch stage is 480 units wide and 360 units tall. So you may need to do some experimentation (and some basic math) to figure out the size of the stars and the spacing. Save this project under the file named **sixStars.yourEASusername.sb**.

Now that you can draw a row of stars, you need to draw a field of 9 rows in order to produce an image that resembles the upper-left of an American flag. *Do not worry about the background color or filling in the stars with white; you can just draw the stars as outlines.*



Five of the rows have six stars, and four of the rows have five stars. You could write nine complete chunks of code, one chunk for each row. If you do that, you get 1 of the five points for this section. A more concise, and more flexible solution, is to look for repetition in the pattern. Then, use loops to produce that repetition. You might use a **repeat-until** or just a regular fixed **repeat** puzzle-piece. There are lots of ways of doing this. You get full credit if the stars line up in the nice diagonals you see in the picture above, but you only lose one point if the right number of stars appear with roughly the correct spacing. Save this project as **flagStars.yourEASusername.sb**.

8. (8 points)

Write a Scratch program that asks the user for his or her age. Based on the input value, your program should print one or more of the following messages:

- ◆ If the age is less than 2, make the sprite tell you: "Sorry, you are not old enough for anything yet"
- ◆ If you are older than 2, then "you don't need diapers"
- ◆ If you are older than 15, then "you can drive"
- ◆ If you are older than 16, then "you can see an R rated movie"
- ◆ If you are older than 17, then "you can vote"
- ◆ If you are older than 20, then "you can gamble"
- ◆ If you are older than 24, then "you can rent a car"
- ◆ If you are older than 65, then "you can retire"

Each message should appear on the screen for 2.5 seconds. The categories are inclusive -- so if someone input an age of 67, then all of the messages except for the first one, should appear, one after another. Your project should be in a file named **age.yourEASusername.sb**

9. (7 points)

Write a program that displays an image of a "healthy food" (such as bananas), an "unhealthy food" (such as *Cheetos*), and the value of a variable named **Good Nutrition Points** (whose value starts at zero).

**Good Nutrition Points** 0

Click on food to eat it.



Your program works as follows: whenever the user clicks the healthy food, the variable gets incremented by 1; whenever the user clicks the unhealthy food, the variable gets decremented by 1. In other words, the variable **Good Nutrition Points** is simply counting mouse clicks. So, if I clicked the bananas image 4 times, and clicked the Cheetos image 6 times, then **Good Nutrition Points** would show a value of **-2**.

You can use any 2 images you wish if you are not interested in the particular food items shown above.

Save the project as **foods,yourEASusername.sb**

10. (0 points)

You might also want to consider signing up for a free account at the Scratch website; this will allow you to upload and download Scratch projects and conveniently share your work with a global user community.

### *Part C: An Original Programming Project (25 points total)*

*This part of the assignment is required ONLY for graduate-credit students. Undergraduate-credit students may complete this part for "extra credit."*

11. (20 points)

And now for a real adventure! Your task for this problem is, quite simply, to have fun with *Scratch* and implement a project of your choice (be it a game, an animation, or something else), subject only to the following requirements.

- i. Your project's filename must be **project,yourEASusername.sb**
- ii. Your project must have at least two sprites, neither of which may be a cat.
- iii. Your project must contain at least three *scripts* in total (*i.e.*, not necessarily per sprite).
- iv. Your project must use at least one condition, one loop, and one variable.
- v. Your project must use at least one sound.
- vi. Your project should be more complex than the simple, short examples that are described in the lecture notes (and appear also in your **ScratchExamples** folder) but ought to be less complex than *Oscartime*. Its complexity should be more

on par with the other projects that come with Scratch. Thus your project should probably use a few dozen puzzle pieces overall.

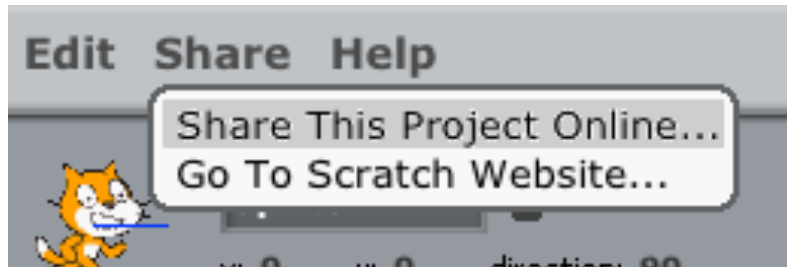
Feel free to look through the projects that come with Scratch for inspiration, but your own project should not be terribly similar to any of them. Try to think of an idea on your own, and then set out to implement it. If, along the way, you find it too difficult to implement some feature, try not to fret: alter your design or work around the problem. If you set out to implement an idea you find fun, you should not find it hard to satisfy this problem's requirements.

If you suspect your program might fall short of our expectations, feel free to ask for our opinion prior to submitting. *All right, off you go.* Impress us! A non-trivial prize shall be awarded for the best two or three programs.

12. (0 points) *This question is optional, but highly recommended!*

Once done with your project, surf on over to <http://scratch.mit.edu/signup> and sign up for a *Scratch* account. Choose any username you'd like, but be sure to remember it and your password!

Then, within Scratch itself, click the "Share" menu and select "Share This Project Online..."



to upload your project to MIT's Scratch Server, using the same username and password with which you just signed up for an account.

Once your project has been uploaded, surf on over to <http://scratch.mit.edu/galleries/view/95235> and click on the button that says "add my projects" on the right-hand side of the page.



(You'll need to be logged in to your account at this point.) Finally, you will need to click on the project you wish to upload, and then click the **Accept** button. Be sure it got added by re-visiting <http://scratch.mit.edu/galleries/view/95235> and verifying that your project is now listed!

*Your responses to problem to problem 13 should be placed in an ordinary text file named **prob13.yourFASusername.txt***

13. (5 points)

In a short paragraph, tell us what your project does (or how to use it). In one or more longer paragraphs, explain how your project works, noting the purpose of each sprite and script. Alternatively, use the new “commenting” feature of Scratch and incorporate typed comments into your actual scripts.

Roughly how much time did you spend on this project?

Did you base **username.sb** on some project that came with *Scratch* or that was demonstrated in lecture? If so, which one?

In implementing **username.sb**, what concepts or implementation details did you struggle with? Why?