

ASSIGNMENT 1

COMP-202, Winter 2015, All Sections

Due: January 20th, 2015 (23:59)

Please read the entire pdf before starting.

You must do this assignment individually and, unless otherwise specified, you must follow all the general instructions and regulations for assignments. Graders have the discretion to deduct up to 20% of the value of this assignment for deviations from the general instructions and regulations.

Question 1: 70 points

Question 2: 30 points

100 points total

It is very important that you follow the directions as closely as possible. The directions, while perhaps tedious, are designed to make it as easy as possible for the Teacher Assistants (TA) to mark the assignments by letting them run your assignment through automated tests. While these tests will not determine your entire grade, it will speed up the process significantly, which will allow the TAs to provide better feedback and not waste time on administrative details. Plus, if the TA is in a good mood while he or she is grading, then that increases the chance of them giving out partial marks. Marks can be removed if comments are missing, if the code is not well structured, or if your solution does not follow the assignment specifications.

Assignment

Question 1: Scratch Project (70 points)

You are to head to <http://scratch.mit.edu/> and create an account. Start by exploring some projects, and click on the **see inside** button in order to understand the logic behind each project. Feel free to make any changes to the scripts to see how it affects the overall behaviour of the program. Once you can honestly say to yourself, “Okay, I think I get this,” you are ready to proceed.

Now, it is time to choose your own adventure! Your mission is, quite simply, to have fun with Scratch and implement a project of your choice (be it an animation, a game, interactive art, or anything else), subject only to the following requirements.

- Your project must have at least two sprites, at least one of which must resemble something other than a cat.
- Your project must have at least six scripts total (i.e., not necessarily six per sprite).
- Your project must use at least one condition, one loop, and one variable.
- Your project must use sound.
- Your project should be more complex than most of those demonstrated in lecture (many of which, though instructive, were quite short) but it can be less complex than, say, Scratch Scratch Revolution. As such, your project should probably use at least 40 puzzle pieces overall.

Feel free to peruse additional projects online 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. But do not try to implement the entirety of your project all at once—pluck off one piece at a time. It is important to take your time and break the overall task into sub and simpler tasks.

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 that you find fun, you should not find it hard to satisfy this assignment requirements.

Once finished with your project, click **See project page** in Scratch’s top-right corner. Ensure your project has a title (in Scratch’s top-left corner), some detailed instructions (in Scratch’s top-right corner), and some notes and/or credits (in Scratch’s bottom-right corner). Then click **Share** in Scratch’s top-right corner so that others like your **TA** can see your project. Finally, take note of the URL in your browser’s address bar. That’s your project’s URL on MIT’s website, and you will need to send it to us in the `Question1.txt` file you will create.

Oh, and if you would like to exhibit your project in Winter 2015’s gallery, head to <http://scratch.mit.edu/studios/788562/projects>, then click **Add projects**, and paste in your own project’s URL.

Question 2: Simple Java Program (30 points)

You are to write your very first interactive java program!

Part 1

The program, we’ll call it “Hal”, asks the user to input the number of laws of prediction according to Arthur C. Clarke. This is achievable using the **Scanner**. You will have to save the input into a **String** variable, and then Hal will repeat back the answer (no matter what it is).

For example, one run of your program might output the following:

```
Hal: How many laws of predictions are there according to Arthur C. Clarke?
2 // Input by the user
Hal: There are indeed 2 laws of prediction.
```

A second example:

```
Hal: How many laws of predictions are there according to Arthur C. Clarke?
no // Input by the user
Hal: There are indeed no laws of prediction.
```

In order to produce this output you will need at least three instructions:

- one instruction to print the sentence to ask for a response, although it could be a letter or multiple, letters, *e.g.* see the second example;
- one instruction to save that information in a **String** variable;
- a third instruction to print Hal’s response.

Please submit this part as `Question2a.java`

Part 2

You might have noticed that this program is not that interesting as it always agrees with the user. For this section, you are to extend part 1 to include an **if-else** statement that verifies if the input is correct. If it is correct, Hal agrees with the user as in part 1. If the input is incorrect, Hal states the input is incorrect and then responds to the user with the correct number of laws what they are.¹ The tricky part is that your program should work for both Arabic numerals and spelled out responses. For example, if the correct answer is one, “Hal” should accept both “1” and “one”.² Please submit this part as `Question2b.java`

¹*N.B.* You have to find out how many laws there are and what they are.

²Please use the `.equals` function with Strings. Do not attempt to cast the String into another data type. You don’t have to make sure the response works for upper case responses such as “One”, just make sure it works for lower case.

What To Submit

You have to submit one zip file with all your files in it to MyCourses under Assignment 1. If you do not know how to zip files, please ask any search engine or friends. Google might be your best friend with this, and a lot of different little problems as well.

`Question1.txt` - Scratch URL
`Question2a.java` - The complete Java code to run the program
`Question2b.java` - The complete Java code to run the program

NB: Do not submit .class files as part of your assignment. They are useless to the TAs.

`confession.txt` (optional) In this file, you can tell the TA about any issues you ran into doing this assignment. If you point out an error that you know occurs in your problem, it may lead the TA to give you more partial credit. On the other hand, it also may lead the TA to notice something that otherwise he or she would not.

Marking Scheme

Up to 20% of question 2 points can be removed from bad indentation of your code as well as omitting comments, or missing files. Marks will be removed as well if the class name rules are not respected according to the general assignment guidelines.

Question 1

TA discretion, has to be at least more complicated than simple examples
and less complicated than the *Scratch Scratch Revolution*

70 points

Question 2

Get the input of the user	5	points
Some comments and right indentation	5	points
Displays the right dialogue	5	points
Both programs work as mentioned	5	points
Use <code>if-else</code> to check valid input	10	points
	30	points