

Troop Leader Supplemental Materials

# GRILL<sup>®</sup> Scout Introductory Patch

## GRILL<sup>®</sup> Scout Introductory Patch

.....

*Troop Leader Supplemental Materials*

### Discover

.....  
*Video links*  
Read or watch videos to learn about programming or 3D modeling

### Connect

.....  
*Brainstorming Prompts*  
Discuss more ways to learn about code

### Appendices

.....  
Packets, Answer Keys, and Continue Learning

### Take Action

.....  
*Example Projects*  
Write a program or build a model by putting knowledge to practice.



# Table of Contents

<b>Before You Start</b> .....	3
<b>Supplemental Materials: Discover</b> .....	4
Watch Videos to Learn 3D Modeling.....	4
<b>Supplemental Materials: Connect</b> .....	5
Brain Storm About 3D Modeling.....	5
<b>Supplemental Materials: Take Action</b> .....	6
Java Project Option 1 Example.....	6
Java Project Option 2 Example.....	7
<b>Appendices</b> .....	
Appendix A. Codecademy Answer Key	
Appendix B. Learning Java Packet	
Appendix C. 3D Modeling	
Appendix D. Continued Learning	

# Before you Start





The goals of this patch are to learn about 3D modeling or computer programming logic, and then apply that logic to a project that can be kept and shared.

This program is set up in three sections—Discover, Connect, and Take Action. Each section has different types of activities based on different levels of technology. This full patch program requires the use of some kind of technology. If there is only tablet access, complete the programming activities. If there is computer access, you can choose programming or 3D Modeling.

To complete the patch, only **ONE** activity from each section (Discover, Connect, and Take Action) needs to be completed. The choice is yours for each section.

In order to promote the widest access possible, this patch offers different levels of technology. The activities that do not require any technology are indicated with a blue pencil icon. All of the remaining icons are representative of the type of technology that can be used to perform the activity.

The continued learning options are not a requirement of this patch. These are simply activities to encourage girls to continue learning and practicing.

	Activity with no technology needed
	Activity that requires a tablet
	Activity that requires a computer
	Activity that can be completed on a tablet or computer

# Before you Start

## Setup

### Java Programming

Tablets - Download Learn Java.



### 3D Modeling

#### Blender

1 Go to <https://www.blender.org/download/>.

2 Download and install the version that coordinates with your operating system.



#### Sketchup

1 Go to [www.sketchup.com/download](http://www.sketchup.com/download).

2 Select "Personal Projects."

3 Fill in the rest of the information.

A screenshot of a web form titled 'Step 2: Tell us a bit about yourself'. The form contains several input fields: a dropdown menu with 'Personal Projects' selected, an email field labeled '\*Email', a checkbox for 'Send me SketchUp news and tips', another dropdown menu with '\*Profession/Interest' selected, and a third dropdown menu with 'Windows' selected. Below these fields is a checked checkbox for 'I agree to SketchUp Make's license agreement \*'. At the bottom of the form is a large red button labeled 'Download SketchUp Make'. Below the button, there is a text link that says 'Think you might need SketchUp Pro?'.

# Supplemental Materials: Discover



## Watch Video Tutorials

Sketchup Videos Playlist: (34 Minutes)

<http://www.sketchup.com/learn/videos/58?playlist=58>

Blender Videos:

⇒ Playlist (35 min)

<https://cgcookie.com/course/blender-basics/>

⇒ First Steps with Blender (5 min)

<https://cgcookie.com/lesson/first-steps-with-blender/>

⇒ Interface and Navigation (9 min)

<https://cgcookie.com/lesson/interface-and-navigation/>

⇒ Selecting and Transforming Objects (7 min)

<https://cgcookie.com/lesson/selecting-and-transforming-objects-2/>

⇒ Adding and Removing Objects (4 min)

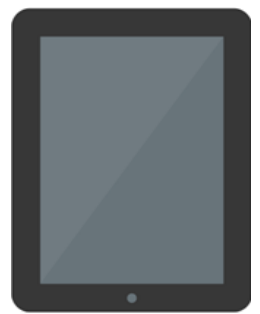
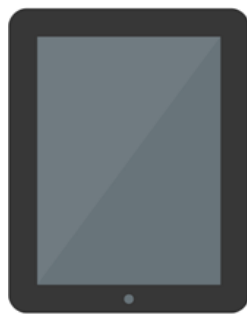
<https://cgcookie.com/lesson/adding-and-removing-objects-2/>

⇒ Using the Interface (8 min)

<https://cgcookie.com/lesson/using-the-interface/>

⇒ Wrapping Up (2 min)

<https://cgcookie.com/lesson/wrapping-up/>

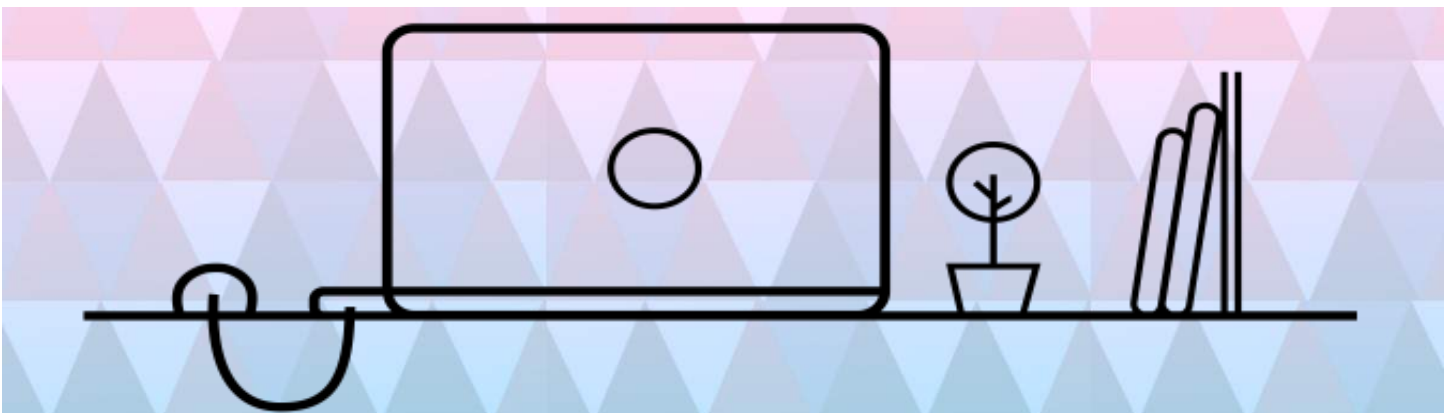


# Supplemental Materials: Connect



## Brainstorm about 3D Modeling

- What are some other ways that 3D modeling could be used?
  - 3D modeling can be used for training simulations, modeling prosthetic limbs, and more!
- How can you continue to improve your skills after completing this patch?
  - You can add to the model that you'll be making later, keep making bigger and more intricate models, or download a 3D modeling program that has more tools.
- Out of the 5 ways 3D modeling can be used, which one interests you the most? Why?
  - Artistic girls might be interested in all or most of the applications. More detail-oriented girls might enjoy modeling for movies and TV so that they could make intricate models. Still, other girls might like the challenge of making models for games.
- Would you rather model, create textures, or animate? Why?
  - This is just a personal preference.



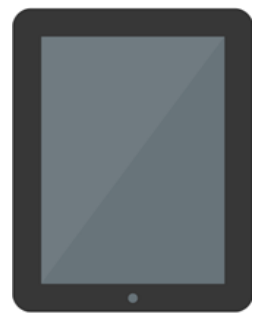
# Supplemental Materials: Take Action

Most software development is done in teams, so encourage girls to work together and help each other with their projects.



Java Project option 1 example solution:

```
12 public class EvenOrOdd {
13
14     /**
15     * @param args the command line arguments
16     */
17     public static void main(String[] args) {
18         //Set num equal to the number you want to test
19         int num = 58562;
20         //rem is the result of num modulo 2
21         int rem = num % 2;
22
23         //if num modulo 2 results in a zero, the number is even
24         if(rem == 0){
25             System.out.println("The number " + num + " is even");
26         }else{
27             System.out.println("The number " + num + " is odd");
28         }
29     }
30 }
31 }
```



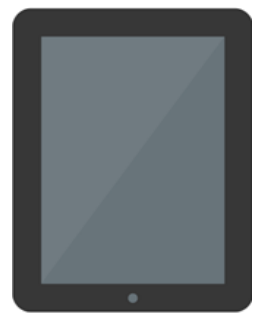


# Supplemental Materials: Take Action Continued



Java Project option 2 example solution:

```
12 public class IsItATriangle {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         //the sum of the lengths of any two sides of a triangle must be
19         //greater than or equal to the length of the third side.
20         int sideA = 4;
21         int sideB = 9;
22         int sideC = 6;
23
24         int sumAB = sideA + sideB;
25         int sumAC = sideA + sideC;
26         int sumBC = sideB + sideC;
27
28         if(sumAB < sideC || sumAC < sideB || sumBC < sideA){
29             System.out.println("This cannot be a triangle");
30         }else{
31             System.out.println("This is a triangle");
32         }
33     }
```



# Codecademy Answer Key

Some answers involve specific values that can vary from person to person.

1.1 <pre>public class YourName {   public static void main(String[] args){     System.out.println("Gilberto");   } }</pre>	1.2 <pre>public class DataTypes {   public static void main(String[] args){     System.out.println(7);   } }</pre>
1.3 <pre>public class DataTypesB{   public static void main(String[] args){     System.out.println(true);   } }</pre>	1.4 <pre>public class DataTypesC {   public static void main(String[] args) {     System.out.println('A');   } }</pre>
1.5 <pre>public class Variables {   public static main (String[] args) {     int myNumber = 42;     boolean isFun = true;     char movieRating = 'A';   } }</pre>	1.6 <pre>public class WhiteSpace {   public static void main (String[] args){     boolean isFormatted = true;     System.out.println(isFormatted);   } }</pre>
1.7 <pre>public class Comments {   public static main (String[] args) {     //System.out.println("Noise!");      /*     This     is     a     multi-line     comment!     */   } }</pre>	1.8 <pre>public class Atithmetic {   public static void main (String[] args){     int myNumber = 123 * 456;     System.out.println(myNumber);   } }</pre>

# Codecademy Answer Key

<p>1.9</p> <pre>public class Modulo {     public static main (String[] args) {         int myRemainder = 7 % 5;         System.out.println(myRemainder);     } }</pre>	<p>1.10</p> <pre>public class RelationalOperators {     public static main (String[] args) {         System.out.println(2 &gt; 1);     } }</pre>
<p>1.11</p> <pre>public class EqualityOperators {     public static main (String[] args) {         System.out.println(true == false);     } }</pre>	<p>1.12</p> <pre>public class Generalizations {     public static main (String[] args) {         //A comment!         boolean isComplete = true;         int awesomeLevel = 121;         int epicLevel = awesomeLevel * 2;         System.out.println(epicLevel);     } }</pre>
<p>2.2</p> <pre>public class And {     public static main (String[] args) {         System.out.println(1 &gt; 2 &amp;&amp; 3 &gt; 4);     } }</pre>	<p>2.3</p> <pre>public class Or {     public static main (String[] args) {         System.out.println(10 &lt; 20    30 &gt; 40);     } }</pre>
<p>2.4</p> <pre>public class Not {     public static main (String[] args) {         System.out.println(!false);         System.out.println(!(5 &gt;= 1));     } }</pre>	<p>2.5</p> <pre>public class Precedence {     public static main (String[] args) {         boolean riddle = !(1 &lt; 8 &amp;&amp; (5 &gt; 2    3 &lt; 5));         System.out.println(riddle);     } }</pre>

# Codecademy Answer Key

2.6

```
public class If {
    public static main (String[] args) {
        if (1 > 0){
            System.out.println("Access granted");
        }
    }
}
```

2.7

```
public class IfElse {
    public static main (String[] args) {
        if (7 < 6){
            System.out.println("Try again...");
        } else {
            System.out.println("Success!");
        }
    }
}
```

2.8

```
public class IfElseIf {
    public static main (String[] args) {
        int round = 6;
        if(round > 12) {
            System.out.println("The match is over!");
        } else if (round > 0) {
            System.out.println("The match is underway!");
        } else {
            System.out.println("The boxing match hasn't started yet");
        }
    }
}
```

# Learning Java

## **General**

- When naming things in Java, instead of using spaces between words, capitalize the first letter of all words except for the first. For example, if you need a name for the distance to a house you could name it distanceToHouse. This is called writing in camelCase.
- Statements end with a semicolon. They are typically only one line.
- When in between curly braces, lines are indented one extra tab.
- Comments, anything written in the program that isn't part of the code, start with "/\*" and are followed by any documentation or an explanation that would be useful for someone in the future looking back at the code.
  - ◆ Comments can also be multiple lines like this:

```
/*
comment
here
*/
```
- Java Documentation <http://docs.oracle.com/javase/8/>

## **Data Types**

- Data types are exactly what they sound like. They're different types of data capable of holding different kinds of values. Their names are case sensitive.
- int - Can be any whole number from -2,147,483,648 to 2,147,483,647
- double - Used for decimal values
- boolean - Holds either true or false
- char - A single character defined with single quotes ''
- String - A list of one or more characters. Typically used for words and sentences. Defined with double quotes ""
- These are the most common and most problems can be solved using these, but more data types as well as more info on those listed can be found here:  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

# Learning Java

## Variables

Variables must be assigned a certain data type and given a name. The name should be meaningful, use camelCase, and start with a lowercase letter or underscore followed by any combination of letters, numbers, and underscores. Two variables cannot have the same name.

## New Variables

A variable needs to be declared and initialized before it can be used. These can be done on one or two lines. A declaration says "This is the datatype of the variable and its name." Initialization says "This is the first value that the variable will hold." We use an equal sign to assign value for initialization as well as afterwards to change the value.

Ex) `int count = 0;` stores the value of 0 in count  
`int otherCount = count;` stores the value of count in otherCount, so otherCount also holds zero. Updating count will not update otherCount.

Ex) `char _uppercaseC = 'C';`

A new string can be made from multiple strings added together.

Ex) `String greeting = "Hello World" + "Good Morning";`

Ex) `boolean open = true;`

Variables can also be declared first and defined later.

Ex) `double speed;`  
`speed = 55.37;`

More Info: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

More on Strings: <https://docs.oracle.com/javase/tutorial/java/data/strings.html>

# Learning Java

## Output-Print Statements

System.out.print(); prints only what is in the parenthesis.

System.out.println(); prints what is in the parenthesis followed by a new line.

Variables or a literal string can be put in the parenthesis. To print multiple things, put a plus sign between them.

Ex)

```
String s = "!";
```

```
System.out.print("Hello ");
```

```
System.out.println("World" + s);
```

```
//The above will print out:
```

```
Hello World!
```

## Conditions/Logic/Operations

Symbol	Meaning	Example	Result
>	Greater than	5 > 7	false
<	Less than	5 < 7	true
>=	Greater than or equal to	4 >= 4	true
<=	Less than or equal to	6 <= 4	false
==	Equal to	0 == 0	true
!=	Not equal	0 != 0	false
!	NOT – Makes the statement false if it was true and true if it was false	!false	true
&&	AND – Both statements must be true for the condition to be true	true && false	false
	OR – One or both statements must be true for the condition to be true	true    false	true
%	Modulo – Divide two numbers and take the remainder as the answer	5 % 2	1

Just like in math, parenthesis can be used to evaluate certain expressions first. There are also certain operations that will be evaluated before others. NOT (!) will be done first, then AND (&&), followed by OR (||). Relational operators such as greater than and equal to are evaluated in the order they are presented.

More Info: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>

# Learning Java

## **Conditionals – If/Else**

A block is what is between curly braces. A basic if/else says that if the condition is true, do what is in the if block. Otherwise, do what is in the else block. A condition solves to a boolean value. The only semicolons are in the blocks. `doSomething();` is a placeholder for what would actually go in the blocks.

```
Ex)
if(condition) {
    doSomething();
} else {
    doSomething();
}
```

If and else if are exclusive to one another. Once one is determined to be true, the program executes what is in the first true block and then skips to what is after the else block. The else if allows for the testing of several different conditions.

```
Ex)
if(condition) {
    doSomething();
} else if(other condition) {
    doSomething();
} else {
    doSomething();
}
```

A single if can also be used without an else or else if, but an else or else if cannot be used without an if. There can also be blocks inside blocks.

```
Ex)
if(condition){
    if(condition){
        doSomething();
    } else {
        doSomething();
    }
}
```

More info: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>



# Java Cheat Sheet

## New Variable

Replace what is between the brackets with specific values.

```
[Data Type] [name] = [value];
```

```
Ex) int count = 5;
```

or

```
[Data Type] [name];
```

```
[name] = [value];
```

```
Ex) int count;
```

```
    count = 5;
```

Data Type	What does it hold?
int	Whole numbers
double	Decimals
boolean	True or False
char	A single character
String	Multiple characters

## Comments

```
//This is a comment
```

```
/*
```

```
This is
```

```
also a
```

```
comment
```

```
*/
```

Symbol	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal
!	NOT
&&	AND
	OR
%	Modulo

## Conditional

```
if(condition) {  
    doSomething();  
} else if(other condition) {  
    doSomething();  
} else {  
    doSomething();  
}
```

## Print

System.out.println(); prints what is in the parenthesis followed by a new line.

System.out.print(); prints only what is in the parenthesis.

## Official Tutorials

<https://docs.oracle.com/javase/tutorial/java/index.html>

# 3D Modeling

3D modeling is the process of taking a 2D concept and sculpting it into three dimensions using specialized software. Once the model is created, textures get added to make it more realistic. It can also get rigged with a “skeleton” so that it can be animated. Then, it gets added to a scene and needs to be rendered into frames. Rendering is the process of adding shading and lighting. The time this takes depends on the complexity of the model.

- Movies/TV
  - Having models look great is what matters most for a movie so models for movies and TV are typically intricate.
  - Special effects aren't limited by hardware and processing power, because they are pre-rendered and recorded.
  - Some high quality animated movies can take up to 30 hours to render a single frame.
- Games/ Simulations
  - For games to run quickly, models can't be as complex as they can be for a movie.
  - Game models are limited by the hardware the game is being played on and the power of the game engine used.
  - Every model in a game has to be rendered in real-time and not stop the game from running smoothly.

# 3D Modeling

- Because models can't be too complex, textures are often used to make them look more detailed.
- There are often several versions of a model created for a game with different LOD (level of detail) so that the more complex model is only shown when the player is close to the object and a simpler model is shown when the player is far away.
- Architecture/Design
  - Creating a 3D model of a proposed building can help to sell it to developers.
  - By creating a 3D model of a city, leaders can see how new buildings will affect the area.
  - 3D modeling can be used to add furniture to a room, change wall color, etc.
- Art
  - To make a piece of art, every step is typically done by one person.
  - The artist needs to decide what part of the project is the most important.
    - ◆ Should the models be very intricate?
    - ◆ Are detailed textures a top priority?
    - ◆ Will there be animation?
  - Pieces of 3D art can also be used in textbooks to illustrate concepts.

# 3D Modeling

- 3D art can be used to create eye catching advertisements.
- Prototypes
  - After a product or part has been designed, but before it is created in real life, a 3D prototype can be created.
  - These virtual prototypes are less expensive to create than a physical one.
  - They allow for easy representation of what a product does.
  - They can also be 3D printed to show an example of what it will look like. This allows for physical testing as well.

# Continued Learning

**Had fun programming? Want to learn even more?  
Here are some GREAT options!**



Complete Codecademy Java course.  
◦ When you're finished, try a new course!



Think of a new project you can program.  
◦ For example, you can write a program that will do an equation for you.



Play a programming game.  
◦ There are a lot of programming games online and apps that teach you how to write code while you play.



Watch the rest of the Sketchup tutorial videos.  
◦ Try out the techniques as they're shown.



Pick a more complicated model and try to make it in Blender.  
◦ Explore the other features in blender such as sculpting and rigging. These can take time, so be patient and keep at it! There are more tutorials here: <https://www.blender.org/support/tutorials/>.



**And of course, continue in Girl Scouts to earn the  
Ambassador Level GRILL® Scout Patch!**