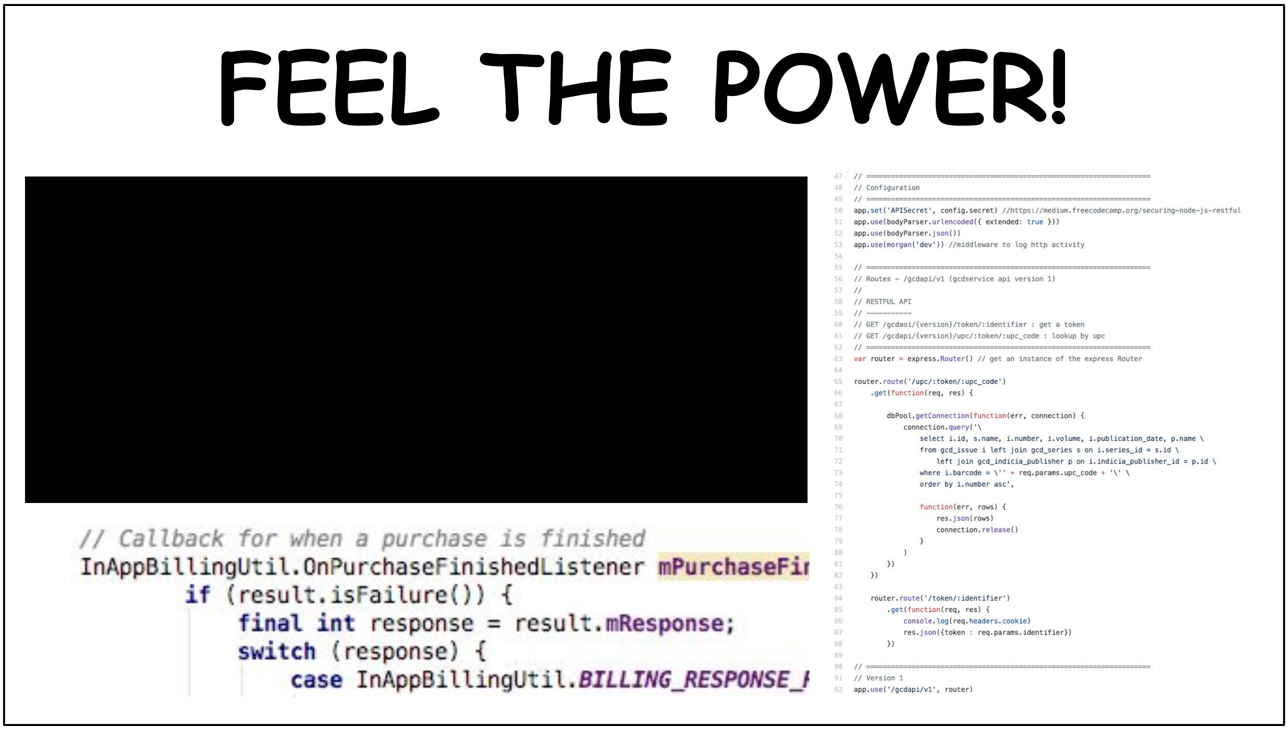


FEEL THE POWER!

[redacted]

```
// Callback for when a purchase is finished
InAppBillingUtil.OnPurchaseFinishedListener mPurchaseFi
        if (result.isFailure()) {
            final int response = result.mResponse;
            switch (response) {
                case InAppBillingUtil.BILLING_RESPONSE_
```

```
// =================================================================
// Configuration
// =================================================================
app.set('APISecret', config.secret) //https://medium.freecodecamp.org/securing-node-js-restful
app.use(bodyParser.urlencoded({ extended: true }))
app.use(bodyParser.json())
app.use(morgan('dev')) //middleware to log http activity

// =================================================================
// Routes - /gcdapi/v1 (gcdservice api version 1)
//
// RESTFUL API
// ----------
// GET /gcdaoi/{version}/token/:identifier : get a token
// GET /gcdapi/{version}/upc/:token/:upc_code : lookup by upc
// =================================================================
var router = express.Router() // get an instance of the express Router

router.route('/upc/:token/:upc_code')
    .get(function(req, res) {

        dbPool.getConnection(function(err, connection) {
            connection.query('\
                select i.id, s.name, i.number, i.volume, i.publication_date, p.name \
                from gcd_issue i left join gcd_series s on i.series_id = s.id \
                    left join gcd_indicia_publisher p on i.indicia_publisher_id = p.id \
                where i.barcode = \'' + req.params.upc_code + '\' \
                order by i.number asc',

                function(err, rows) {
                    res.json(rows)
                    connection.release()
                }
            )
        })
    })

    router.route('/token/:identifier')
        .get(function(req, res) {
            console.log(req.headers.cookie)
            res.json({token : req.params.identifier})
        })

// =================================================================
// Version 1
app.use('/gcdapi/v1', router)
```

***** Make sure you have a sign-in sheet!!! *****
https://docs.google.com/document/d/1C9n6aMJss3ahvAUojt93NZQ6o4gMlTRjShyblWA0jO0/edit?usp=sharing

***** Take a group Photo with the screen in the background showing the merit badge patch *****

<u>Ground Rules</u>

1. This is a Scouting Event - ALL the associated rules apply. Including 2 deep leadership and taking a Buddy if you leave the area.
2. Can not help yourself to snacks or drinks - we need to respect allergies.
3. Class B Uniforms required
4. Recite Pledge of Allegiance to the Flag
5. Recite Scout Oath
6. Recite Scout Law
7. Phones are allowed and encouraged - this is my challenge to the Scouts! - Just make sure you're still listening so you can complete the requirements :)

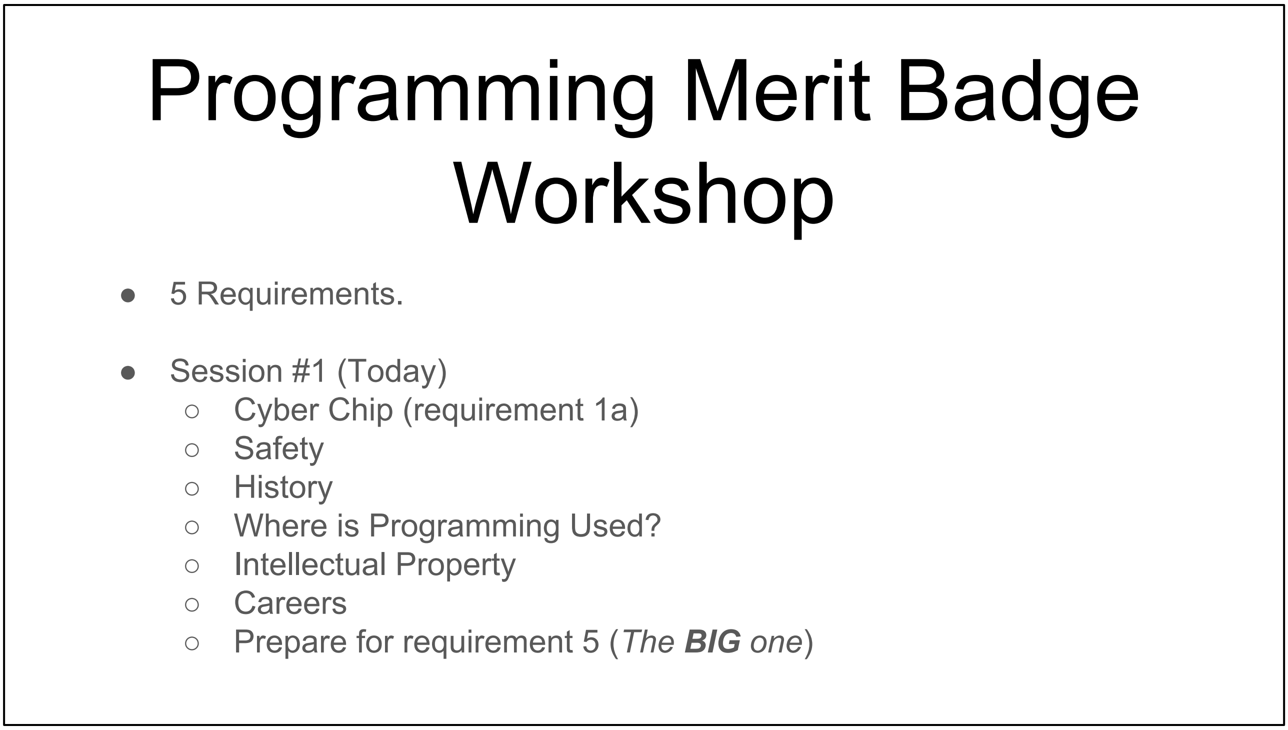
<u>Initial Introduction</u>

- Lights can be dimmed to get full attention and focus

- Do an introduction geared to spark their interest at the very beginning of the Workshop. It also covers most of the *"What is Programming?"* section.

- Have the Scouts gather around my workstation:

    - Run the NumberExplosion iOS app in xcode on the Simulator (or some other program you have source code for)
    - Run it again on a real mobile device.
    - Let them glance at the source code. See if anyone knows the name of the programming language?
    - Set a breakpoint and explain to them the meaning if a debugger.
    - Explain to what an IDE is to them.
    - Show and explain to them the github repository.
    - Show them the app in the app store
    - Show them itunes connect and explain to them what it is used for.
    - Explain to them that as a programmer this is how you would spend a good part of your work day.
    - Talk to the Scouts a little about how you transfer an idea - to requirements - to a program. (“*What is Programming?”* Section)



Programming Merit Badge Workshop

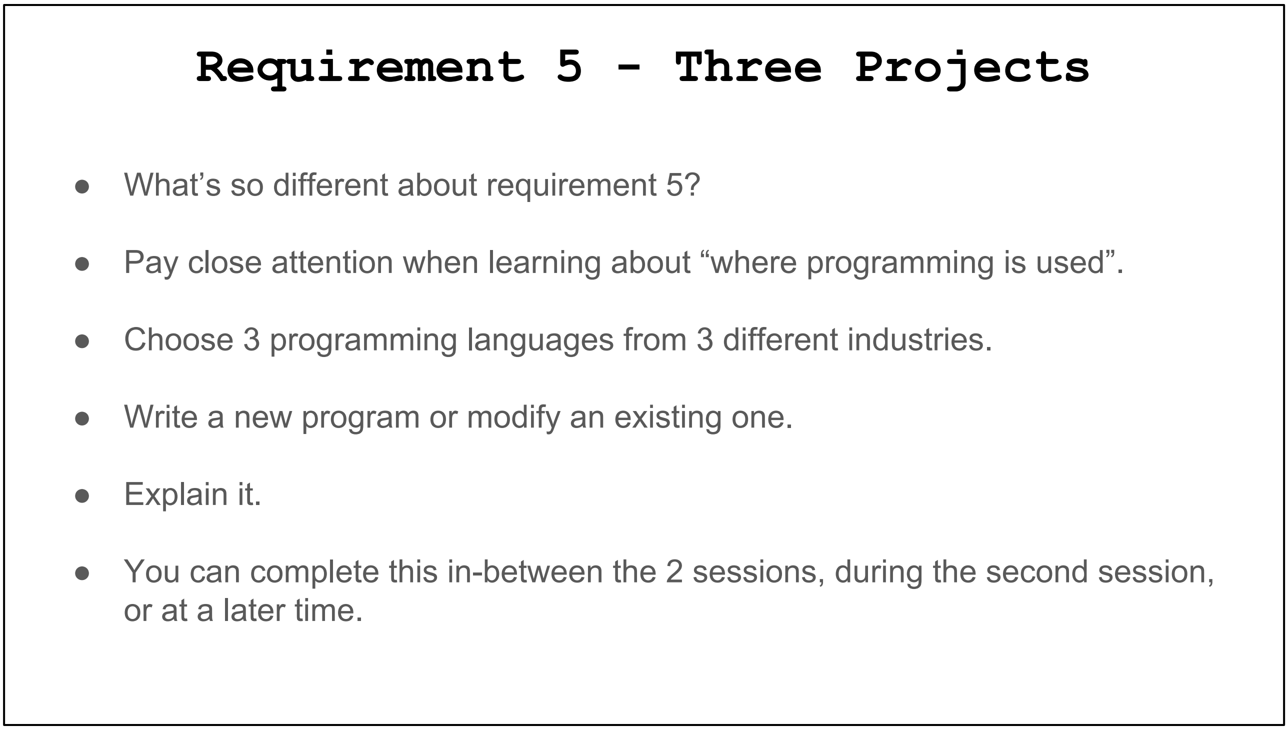
- 5 Requirements.

- Session #1 (Today)
  - Cyber Chip (requirement 1a)
  - Safety
  - History
  - Where is Programming Used?
  - Intellectual Property
  - Careers
  - Prepare for requirement 5 (*The **BIG** one*)

Welcome to this Programming Merit Badge workshop! The workshop consists of 2 four hour sessions. This is the first. The other one will be in a couple of weeks.

There are 5 requirements. In this first sessions you will:

1. Get started on earning your cyber chip (if you haven't already earned it)
2. Complete requirements 1-4
3. Prepare for requirement 5 which can be completed in between the two sessions or during the second session if you need help.



Requirement 5 - Three Projects

- What’s so different about requirement 5?
- Pay close attention when learning about “where programming is used”.
- Choose 3 programming languages from 3 different industries.
- Write a new program or modify an existing one.
- Explain it.
- You can complete this in-between the 2 sessions, during the second session, or at a later time.

The last requirement of this Merit Badge is completely “hands-on”.

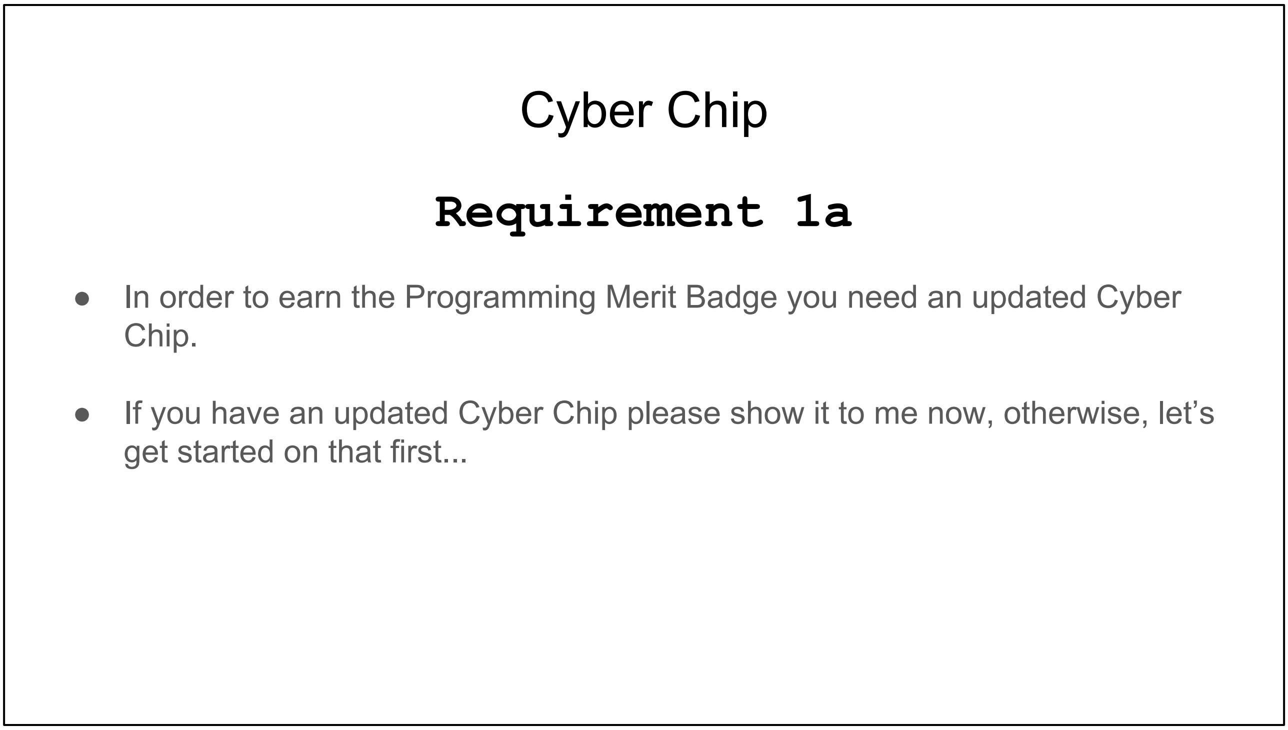
You need to pay close attention to the section of the pamphlet that talks about “Where is Programming Used?” I’ll be going over this section later.

You will be asked to choose 3 different programming languages (Java, Javascript, etc..) from 3 different industries (Engineering, Robotics, Mobile Devices, etc) and either write a new program or modify an existing one. Then you have to explain it.

Programming Merit Badge

Welcome! And ...

Have the Scouts watch this welcome video.



Cyber Chip

Requirement 1a

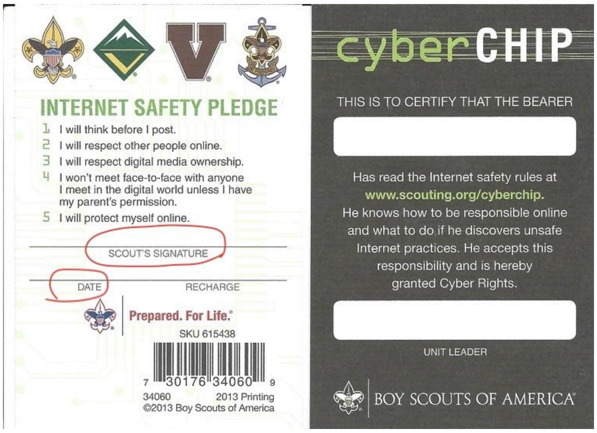
- In order to earn the Programming Merit Badge you need an updated Cyber Chip.
- If you have an updated Cyber Chip please show it to me now, otherwise, let's get started on that first...

Turns out for this initial workshop all the Scouts need to earn their Cyber Chip. In the future scouts who have completed their Cyber Chip can begin tinkering with an example program I can have running in a debugger off to the side (Language and env are TBD. This is a TODO item for the next workshop).

Supporting handouts here:
https://docs.google.com/document/d/1epsq0Iro8mmGrvjfPaOXp_p5JtfvKQPALNjTwPaEEIg/edit?usp=sharing

Cyber Chip (Grades 6-8) Requirement #1

1. Read, commit to, and sign the Internet Safety Pledge on your Cyber Chip.



INTERNET SAFETY PLEDGE

1 I will think before I post.
2 I will respect other people online.
3 I will respect digital media ownership.
4 I won't meet face-to-face with anyone I meet in the digital world unless I have my parent's permission.
5 I will protect myself online.

SCOUT'S SIGNATURE

DATE

RECHARGE

Prepared. For Life.

SKU 615438

7 30176 34060 9

34060 2013 Printing
©2013 Boy Scouts of America

cyberCHIP

THIS IS TO CERTIFY THAT THE BEARER

Has read the Internet safety rules at www.scouting.org/cyberchip. He knows how to be responsible online and what to do if he discovers unsafe Internet practices. He accepts this responsibility and is hereby granted Cyber Rights.

UNIT LEADER

BOY SCOUTS OF AMERICA

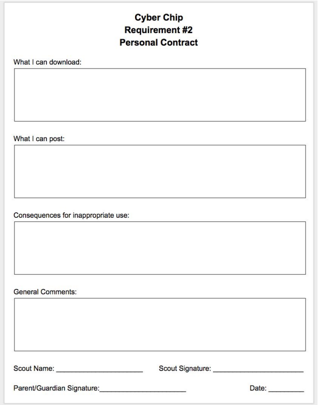
Have them do this now. Go over the pledge items with them. Make sure they understand what it means.

Cyber Chip (Grades 6-8) Requirement #2

Write a personalized contract between you and your Parent/Guardian that outlines rules for using the computer and mobile devices. You may use the following template:

Take Home Work

Take this home, fill it out and sign it with your Parent/Guardian.



Cyber Chip
Requirement #2
Personal Contract

What I can download:

What I can post:

Consequences for inappropriate use:

General Comments:

Scout Name: ________________ Scout Signature: _________________

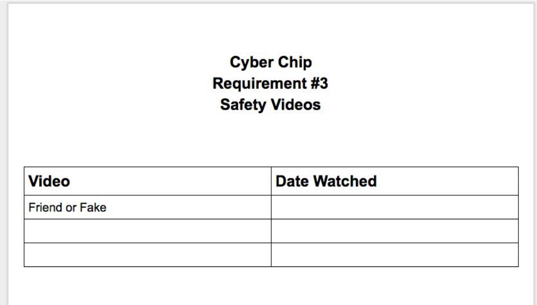
Parent/Guardian Signature:________________ Date: _______

Have the Scouts create the contract and place it in their Cyber Chip take home folder.

Cyber Chip (Grades 6-8) Requirement #3

- Let’s watch a few videos on Internet Safety! :)

- You’ll need to keep track of what you’ve watched for the Cyber Chip requirement #3. You can use the following template for this:

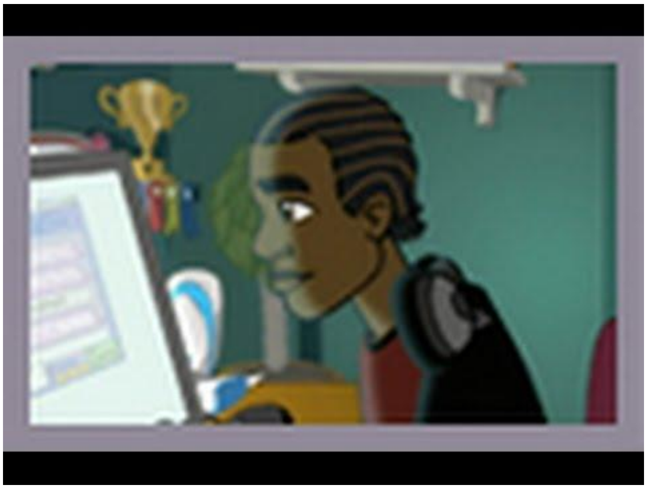


Cyber Chip
Requirement #3
Safety Videos

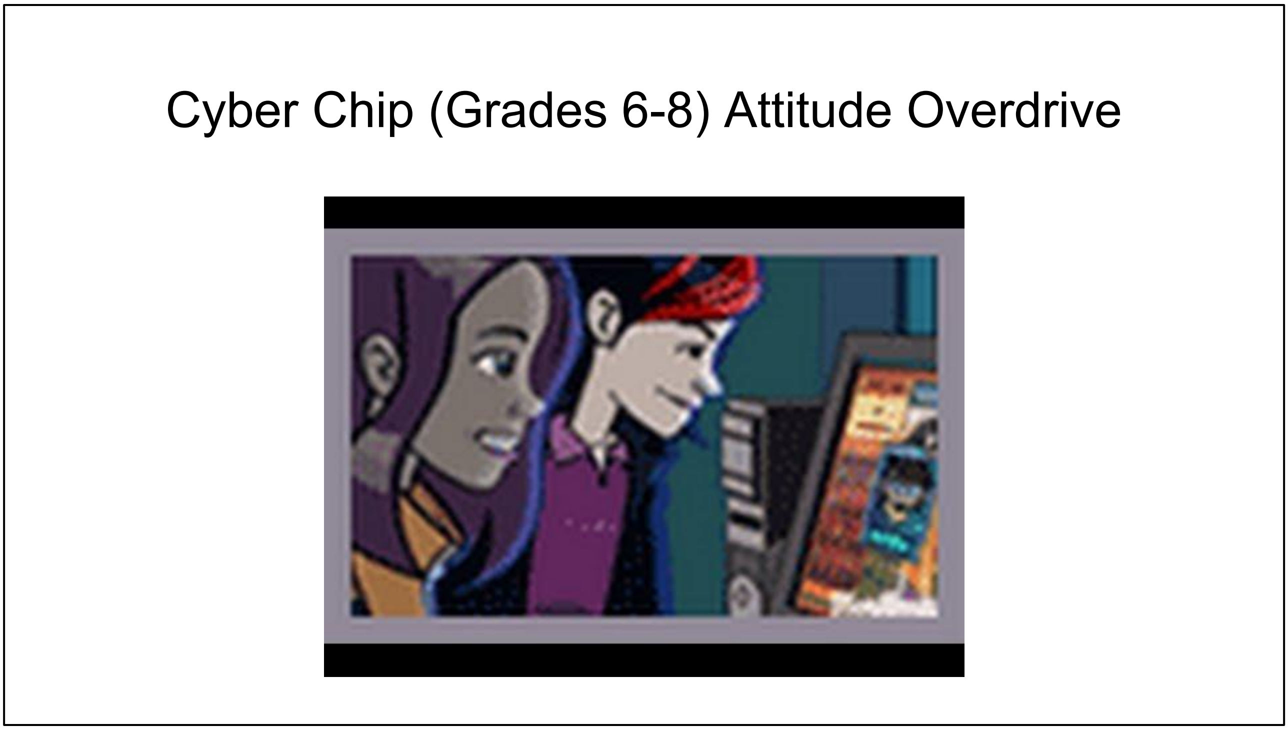
Video	**Date Watched**
Friend or Fake	

Have the Scouts write out the template and fill it in as they watch each video. Once they have completed watching all 3 videos have them place it in their Cyber Chip take home folder.

Cyber Chip (Grades 6-8) Friend or Fake



After the Scouts watch this video have them fill in the name and date in their video safety log.

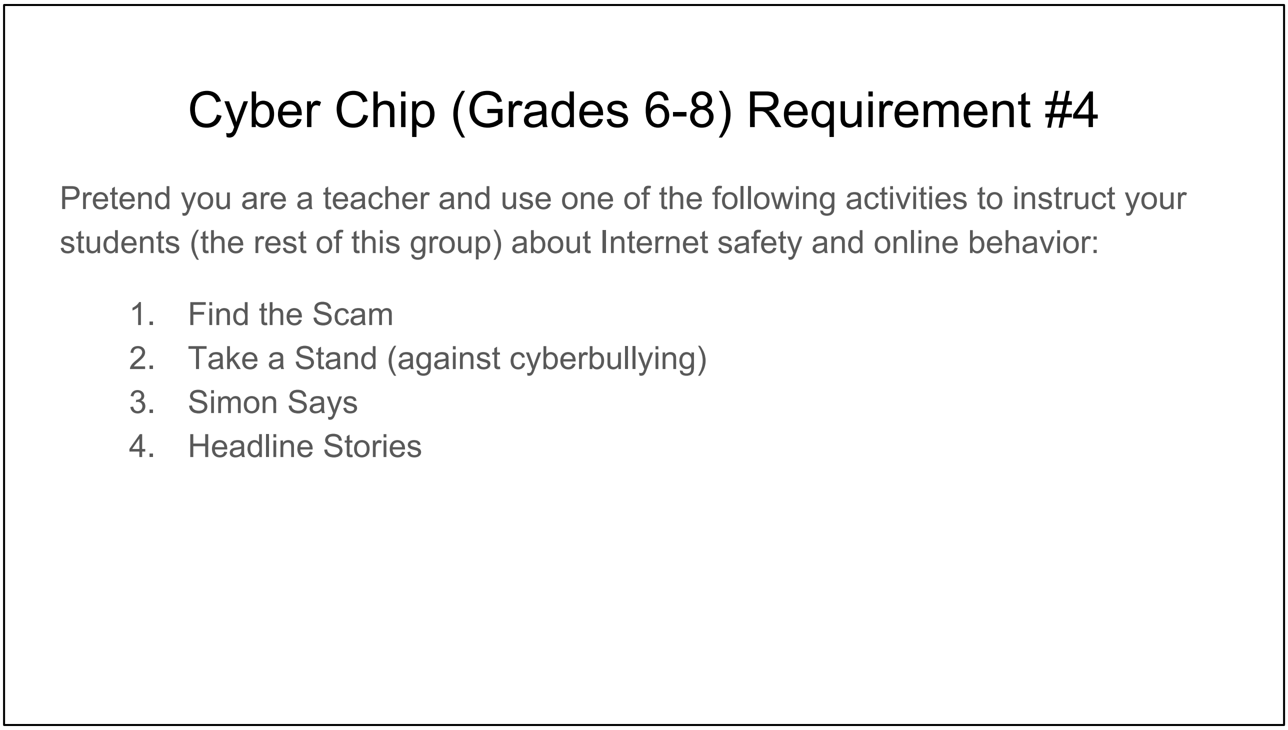


Cyber Chip (Grades 6-8) Attitude Overdrive

After the Scouts watch this video have them fill in the name and date in their video safety log.

Cyber Chip (Grades 6-8) Post to be Private

After the Scouts watch this video have them fill in the name and date in their video safety log. Since this is the last Cyber Chip video have them place their video safety log in their Cyber Chip take home folder.

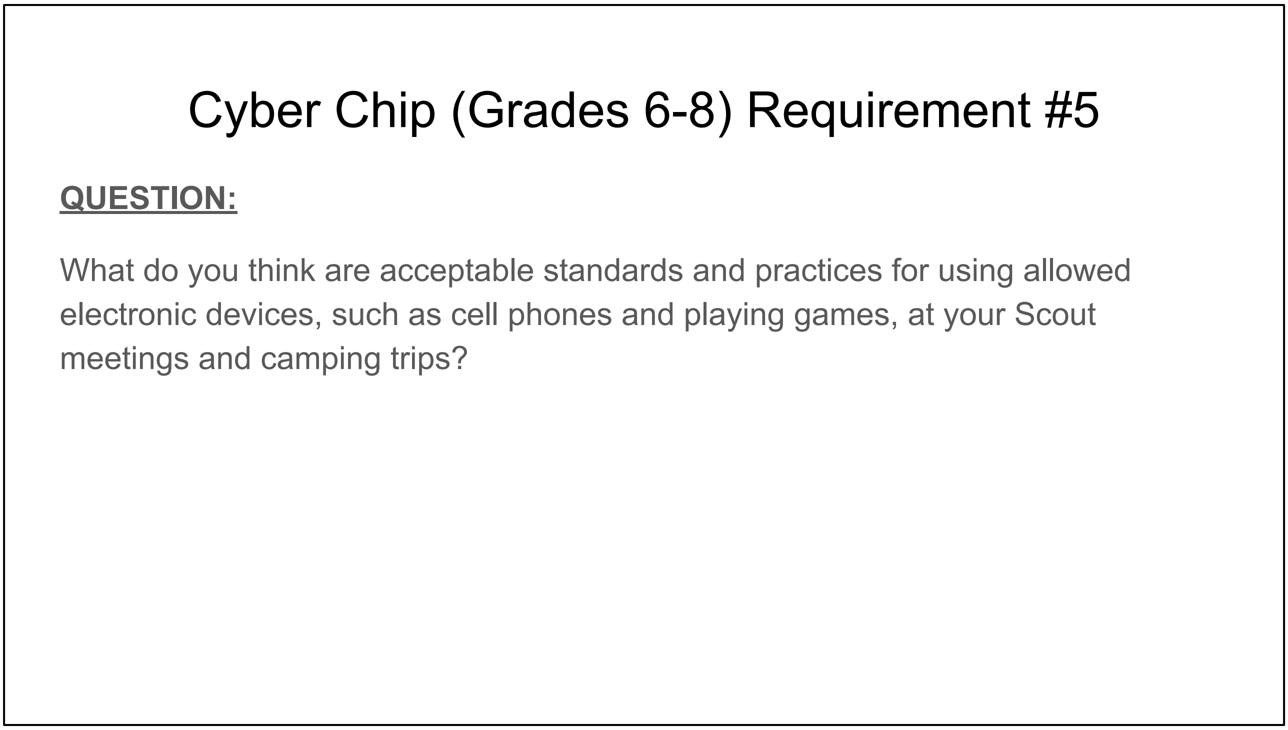


Cyber Chip (Grades 6-8) Requirement #4

Pretend you are a teacher and use one of the following activities to instruct your students (the rest of this group) about Internet safety and online behavior:

1. Find the Scam
2. Take a Stand (against cyberbullying)
3. Simon Says
4. Headline Stories

Explain and demonstrate what needs to be done here. The scouts should choose one and place it in their Cyber Chip take home folder. Instruct them that they will complete this at our next session.

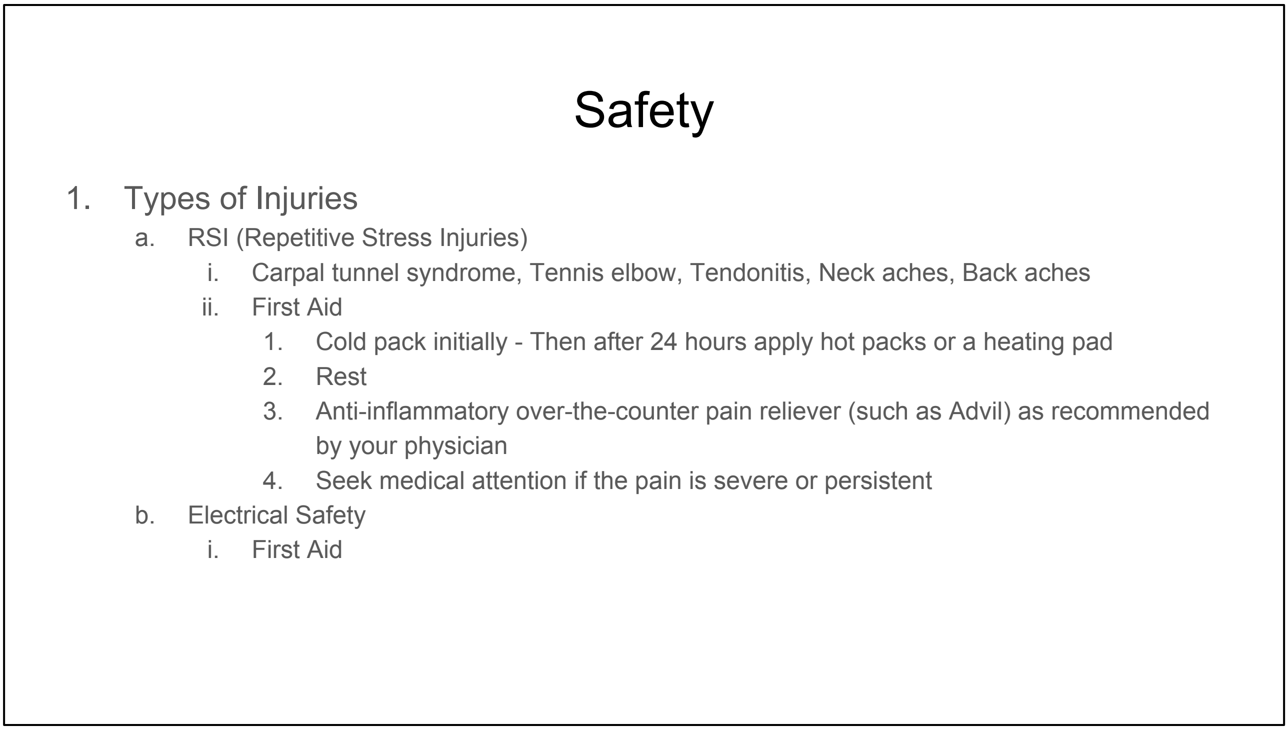


Cyber Chip (Grades 6-8) Requirement #5

QUESTION:

What do you think are acceptable standards and practices for using allowed electronic devices, such as cell phones and playing games, at your Scout meetings and camping trips?

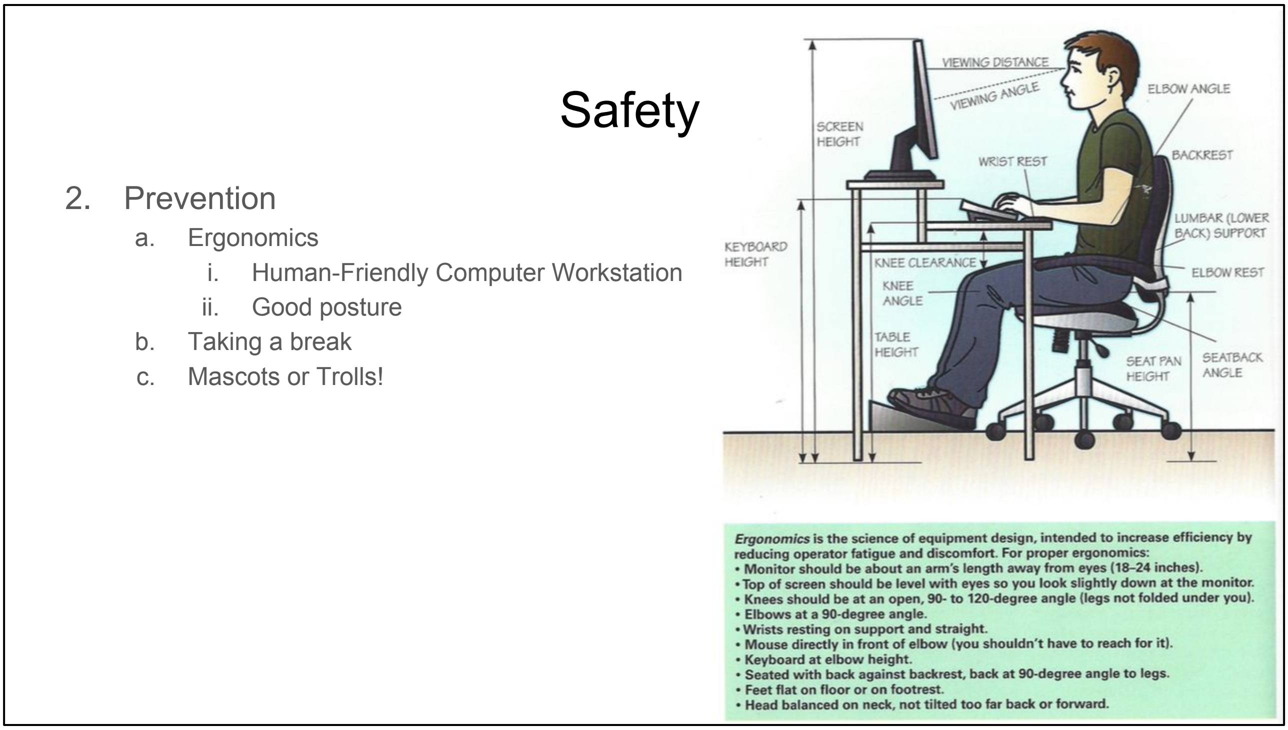
Explain to the Scouts that they must think about the answer to this at home for our next session. Give them a handout with the question and a few lines for them to answer and have them place it in their Cyber Chip take home folder. After the next session they can go over it with the Scoutmaster at one of the meetings. The SM will then sign their Cyber Chip card and this will complete the requirement.



Safety

1. Types of Injuries
    a. RSI (Repetitive Stress Injuries)
        i. Carpal tunnel syndrome, Tennis elbow, Tendonitis, Neck aches, Back aches
        ii. First Aid
            1. Cold pack initially - Then after 24 hours apply hot packs or a heating pad
            2. Rest
            3. Anti-inflammatory over-the-counter pain reliever (such as Advil) as recommended by your physician
            4. Seek medical attention if the pain is severe or persistent
    b. Electrical Safety
        i. First Aid

Talk to the Scouts about the different types of injuries that can occur when sitting at a computer workstation for long periods of time. Discuss how they can be treated.



Safety

2. Prevention
   a. Ergonomics
      i. Human-Friendly Computer Workstation
      ii. Good posture
   b. Taking a break
   c. Mascots or Trolls!

VIEWING DISTANCE
VIEWING ANGLE
ELBOW ANGLE
SCREEN HEIGHT
WRIST REST
BACKREST
LUMBAR (LOWER BACK) SUPPORT
KEYBOARD HEIGHT
KNEE CLEARANCE
ELBOW REST
KNEE ANGLE
TABLE HEIGHT
SEAT PAN HEIGHT
SEATBACK ANGLE

Ergonomics is the science of equipment design, intended to increase efficiency by reducing operator fatigue and discomfort. For proper ergonomics:

- Monitor should be about an arm's length away from eyes (18–24 inches).
- Top of screen should be level with eyes so you look slightly down at the monitor.
- Knees should be at an open, 90- to 120-degree angle (legs not folded under you).
- Elbows at a 90-degree angle.
- Wrists resting on support and straight.
- Mouse directly in front of elbow (you shouldn't have to reach for it).
- Keyboard at elbow height.
- Seated with back against backrest, back at 90-degree angle to legs.
- Feet flat on floor or on footrest.
- Head balanced on neck, not tilted too far back or forward.

Good posture habits and human-friendly equipment can help you avoid injuries when working on a computer for long periods of time.

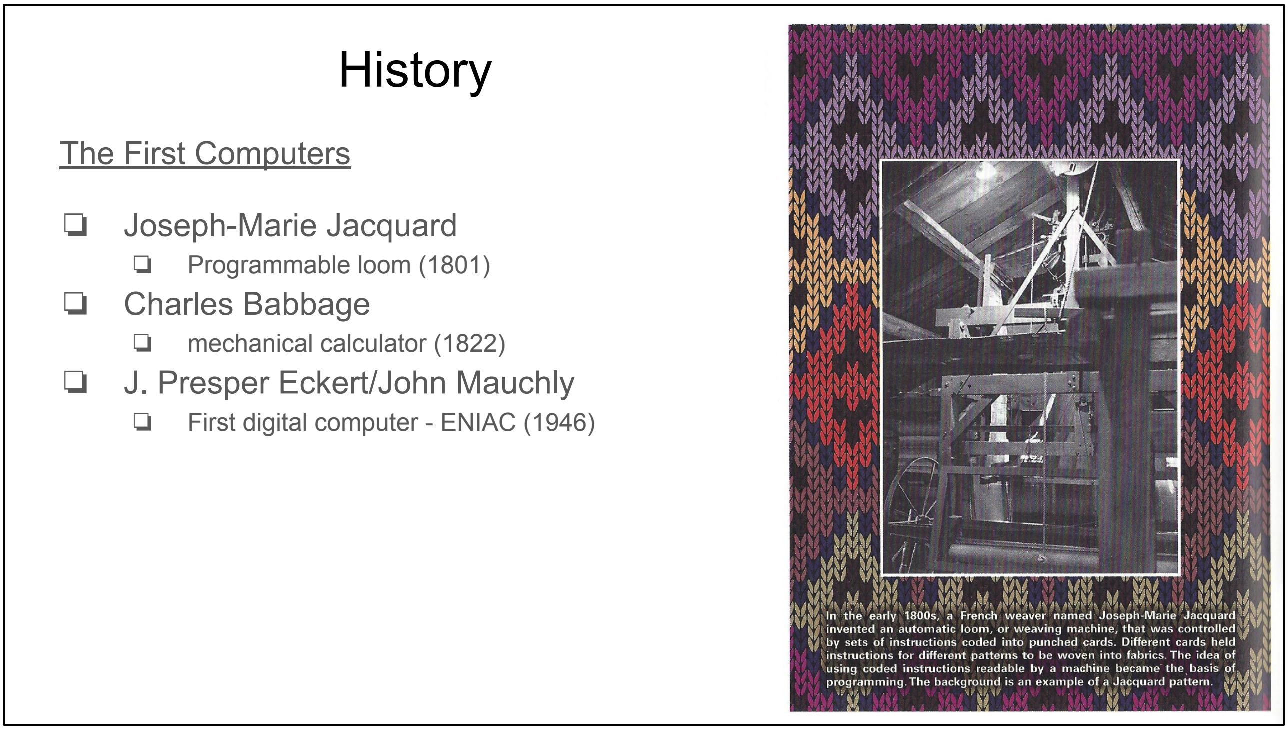
Ergonomics is the science of equipment design, intended to increase efficiency by reducing operator fatigue and discomfort.

Demonstrate to the Scouts the different kinds of equipment that i use: My Chair, ball chair, blue light blocking glasses, finger exerciser.

Taking a break… gives your fingers, your eyes, wrists, back and neck a rest. Stand-up and walk around to stretch your legs. Talk about the masseuse we have at the office.

Mascots help relieve stress too! A tangible/real troll, plush or toy that can help rest your mind by focusing on a non-virtual object! (Talk about the trolls I have around my workstation - pen knives, cannons, etc…)

Explain a little about the unique work environments found in development shops geared toward productivity and safety.



History

The First Computers

- □ Joseph-Marie Jacquard
  - □ Programmable loom (1801)
- □ Charles Babbage
  - □ mechanical calculator (1822)
- □ J. Presper Eckert/John Mauchly
  - □ First digital computer - ENIAC (1946)

In the early 1800s, a French weaver named Joseph-Marie Jacquard invented an automatic loom, or weaving machine, that was controlled by sets of instructions coded into punched cards. Different cards held instructions for different patterns to be woven into fabrics. The idea of using coded instructions readable by a machine became the basis of programming. The background is an example of a Jacquard pattern.

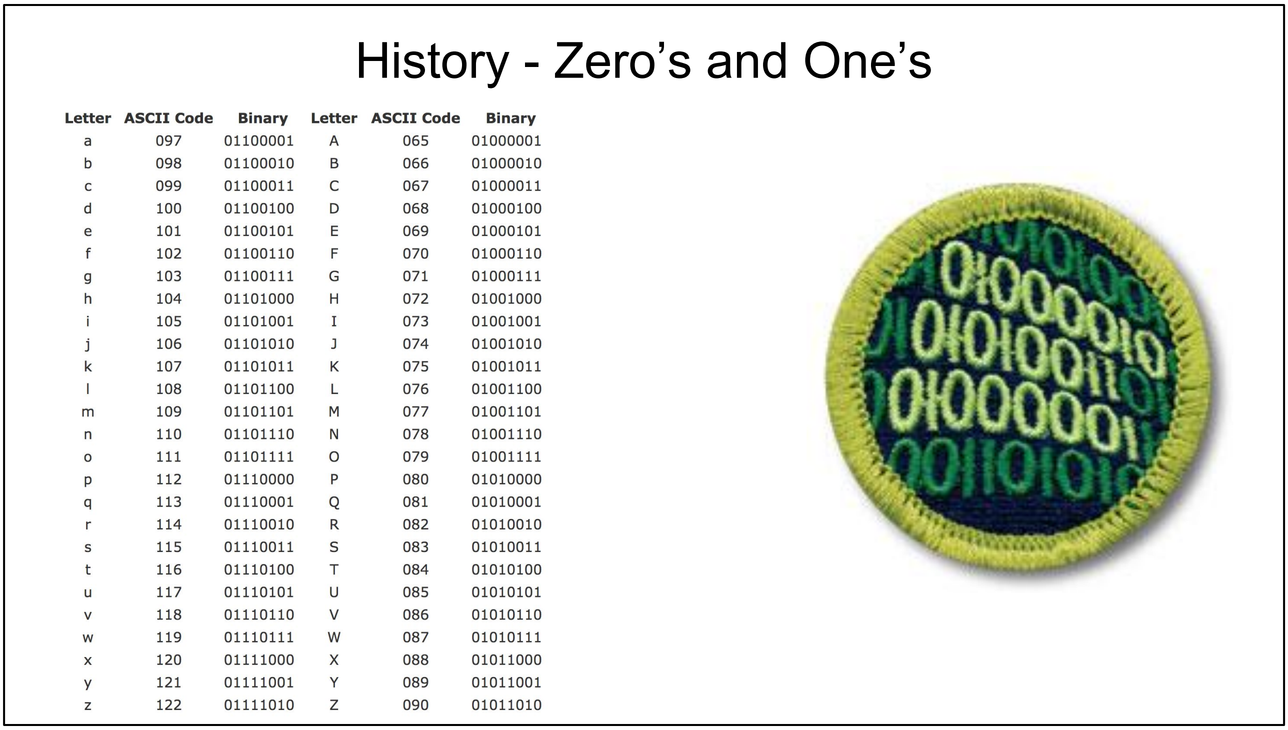
1. Joseph-Marie Jacquard was not the inventor of the programmable loom, as many people imagine, actually he created an attachment to the loom, which played a very important role not only in the textile industry, but also in development of other programmable machines, such as computers, for example the Analytical Engine of Charles Babbage.

2. The Difference Engine was the first automatic, mechanical calculator designed by British mathematician Charles Babbage, who proposed its construction in 1822 to the Royal Astronomical Society. He suggested the machine would employ the decimal number system and would be powered by turning a handle, as a method to calculate mathematical tables mechanically, therefore removing the high rate of human error.

3. The ENIAC (Electronic Numerical Integrator and Computer) was invented by J. Presper Eckert and John Mauchly at the University of Pennsylvania and began construction in 1943 and was not completed until 1946. It occupied about 1,800 square feet and used about 18,000 vacuum tubes, weighing almost 50 tons. Although the Judge ruled that the ABC computer was the first digital computer, many still consider the ENIAC to be the first digital computer because it was fully functional.

4. Explain a little bit what punch cards are.

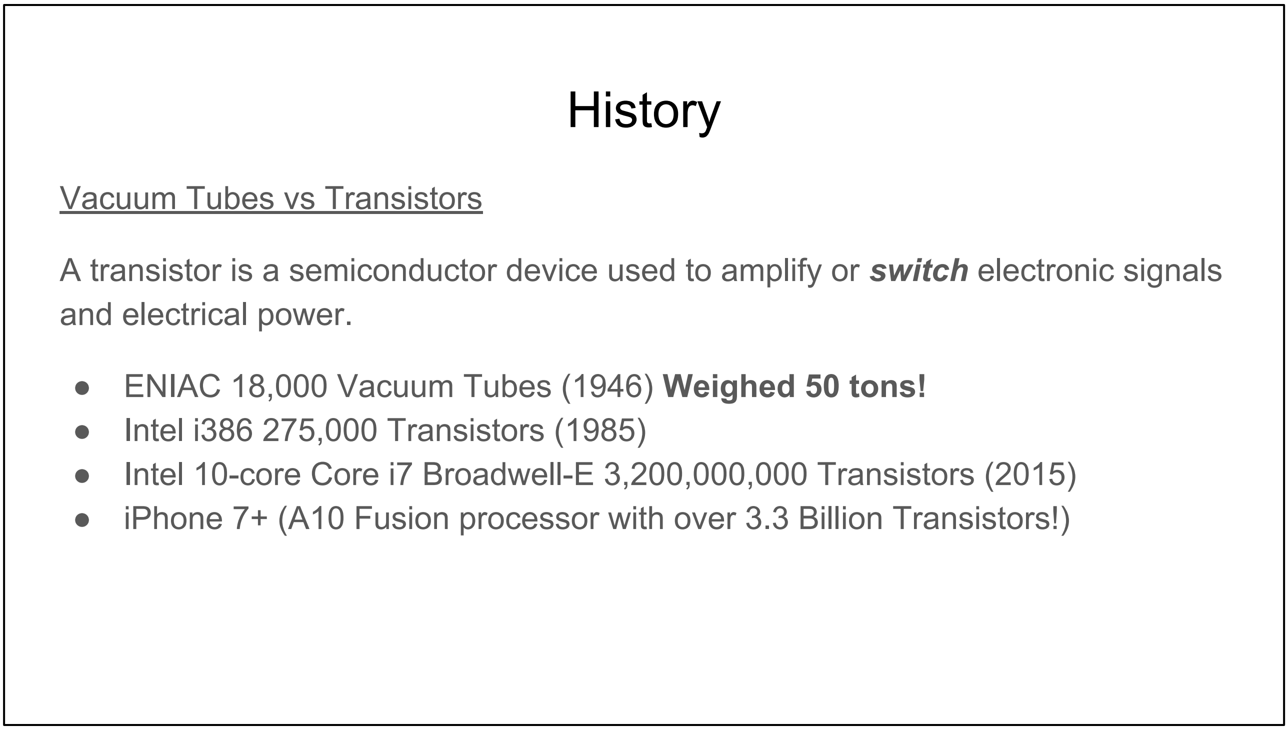
5. Explain a what a “digital computer” is.



History - Zero’s and One’s

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Binary Code - Machine Code - The First Programming Language
Explain to the Scouts about Binary numbers (base 2) and ASCII codes and how all data and instructions must ultimately be broken down into zero’s and one’s in order for computer microprocessors to understand them. Demonstrate using the Programming Merit Badge itself. It consists of 3 binary ASCII characters that represent the letters “B” “S” “A”. When placed on a sash the Scouts must make sure it’s in the upright position.



History

Vacuum Tubes vs Transistors

A transistor is a semiconductor device used to amplify or ***switch*** electronic signals and electrical power.

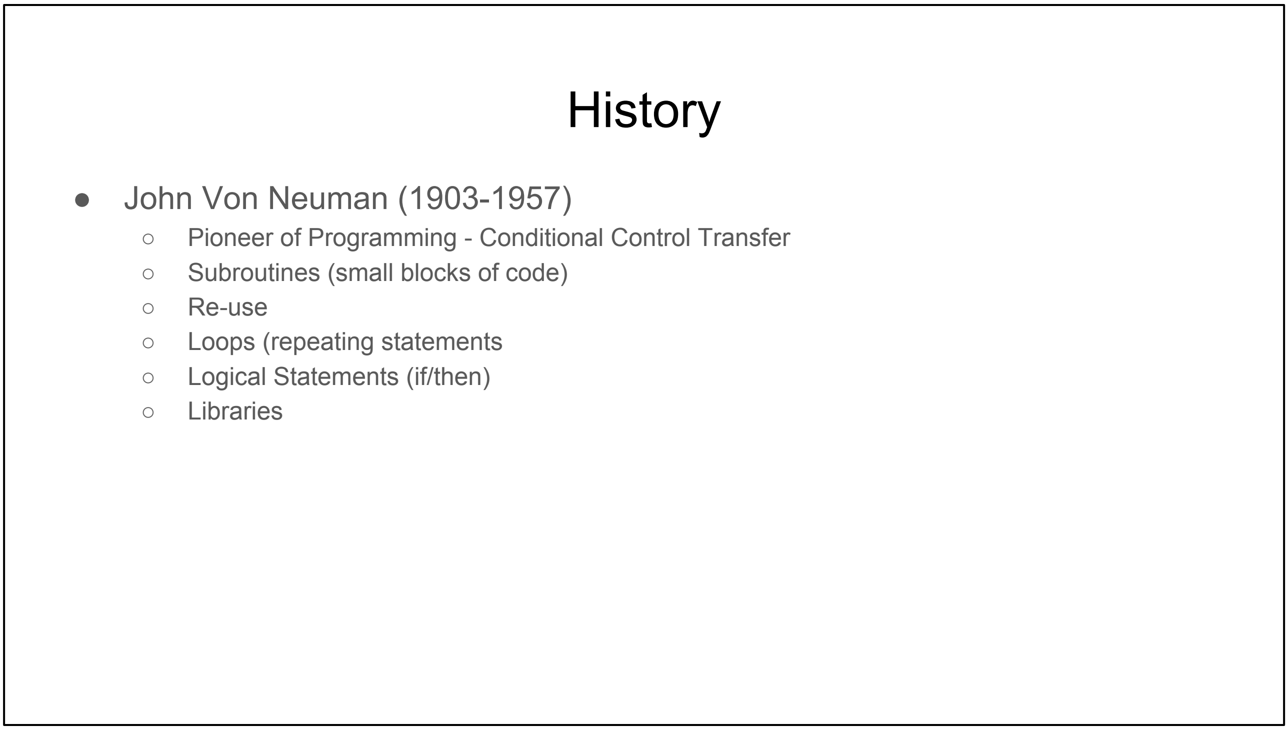
- ENIAC 18,000 Vacuum Tubes (1946) **Weighed 50 tons!**
- Intel i386 275,000 Transistors (1985)
- Intel 10-core Core i7 Broadwell-E 3,200,000,000 Transistors (2015)
- iPhone 7+ (A10 Fusion processor with over 3.3 Billion Transistors!)

1. Give the Scouts a brief overview and appreciation of how powerful today's microprocessors are compared to the first computers.

2. Explain a little about what a Transistor is and its significance to computers and computer programming (binary code).

3. Explain to the Scouts that there are other Merit badges that explore these concepts deeper (Electronics, Robotics)

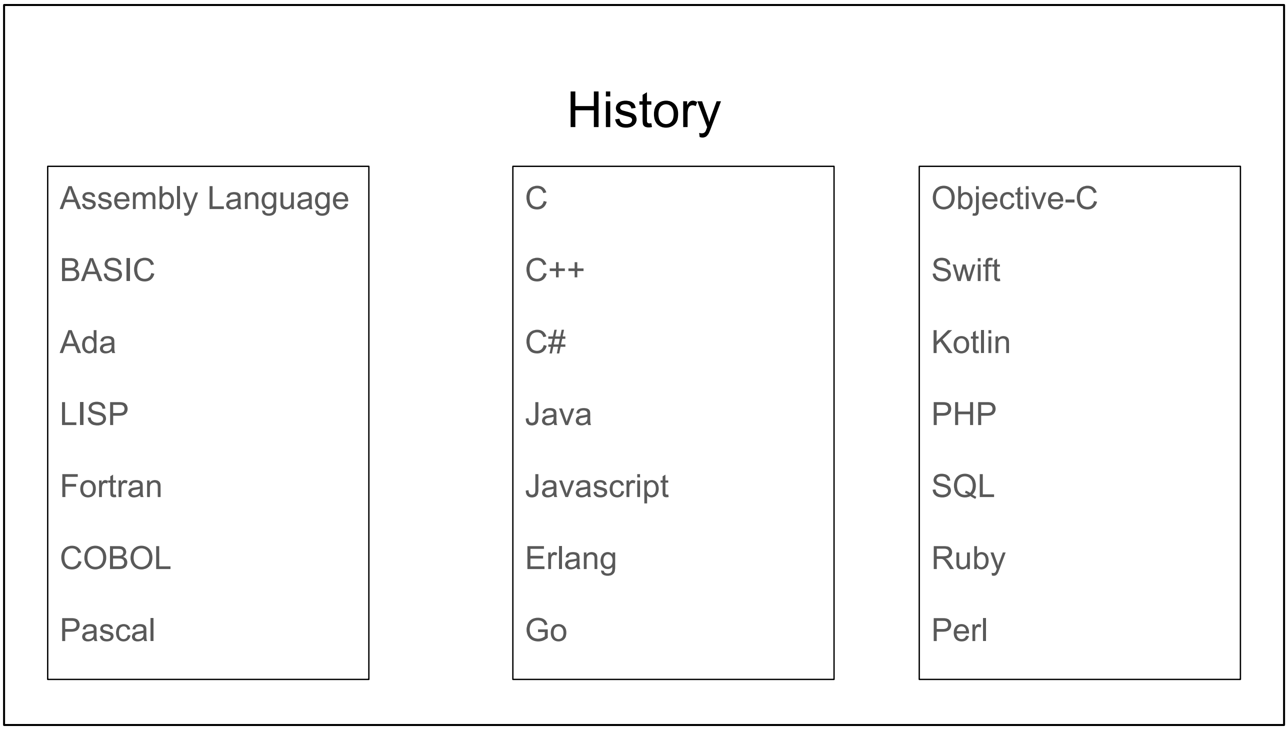
4. Show the Scouts my i386 cpu and a 3906 Transistor. Let them examine them.



History

- John Von Neuman (1903-1957)
  - Pioneer of Programming - Conditional Control Transfer
  - Subroutines (small blocks of code)
  - Re-use
  - Loops (repeating statements
  - Logical Statements (if/then)
  - Libraries

Talk a little about John Von Neuman and explain “Conditional control transfer” and why these concepts helped the evolution of computer programming languages.



History

Assembly Language

BASIC

Ada

LISP

Fortran

COBOL

Pascal

C

C++

C#

Java

Javascript

Erlang

Go

Objective-C

Swift

Kotlin

PHP

SQL

Ruby

Perl

1. Next generation programming languages took programming beyond binary code and assembly. These languages allow developers to write programs that ran on different processors where before each program was specific to only a single processor.

2. For instance, if, in 1990, i had an apple computer and an intel windows computer and i wrote my program in assembly language, I would have to write 2 very different programs because they had different processors with different instruction sets. If, however, I used a higher level next generation such as BASIC or C, I would only have to write it once and compile it on each platform.

3. Explain a little bit to the Scouts what a compiler is and why it is needed.

John Von Neuman
Conditional control transfer - subroutines

Programming in Objects

Programming for the World Wide Web

Hello World!

Step-by-Step Communications

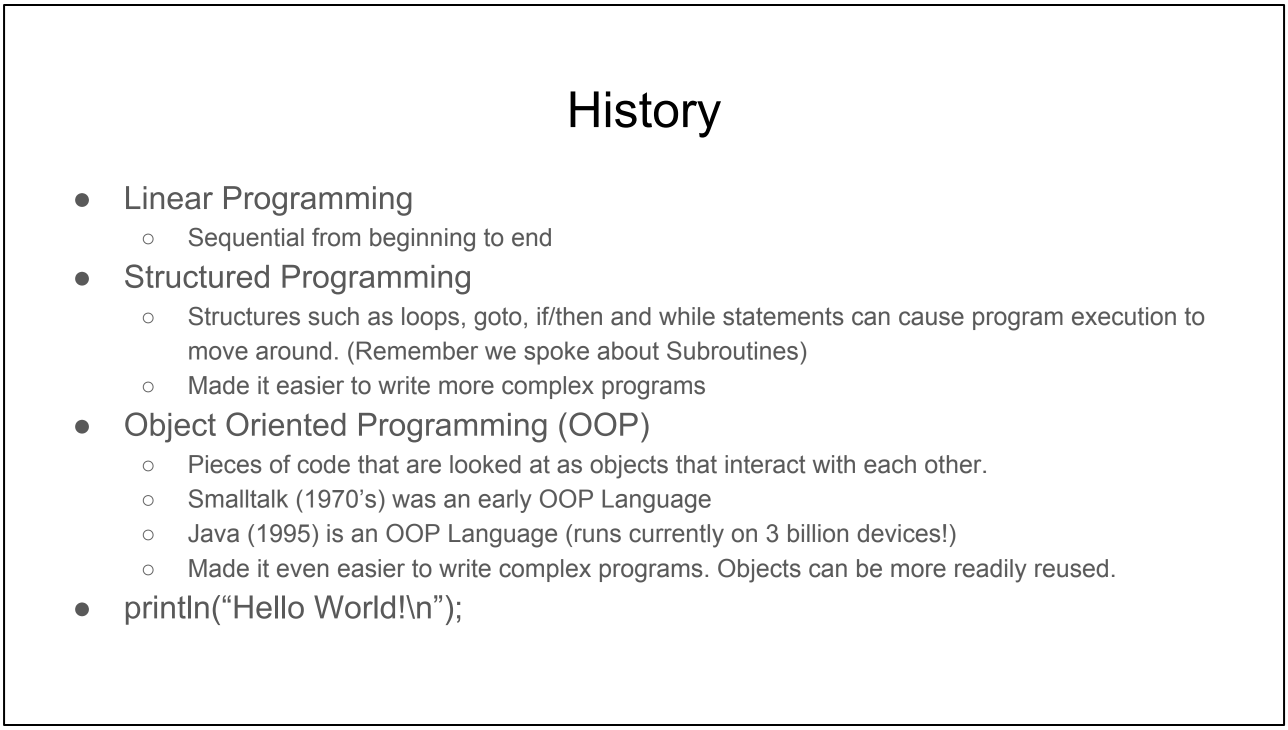
“Fill the canteen with water”

Planning/Requirements

State Diagrams

Linear or Structured

Requirements 2a and 2b

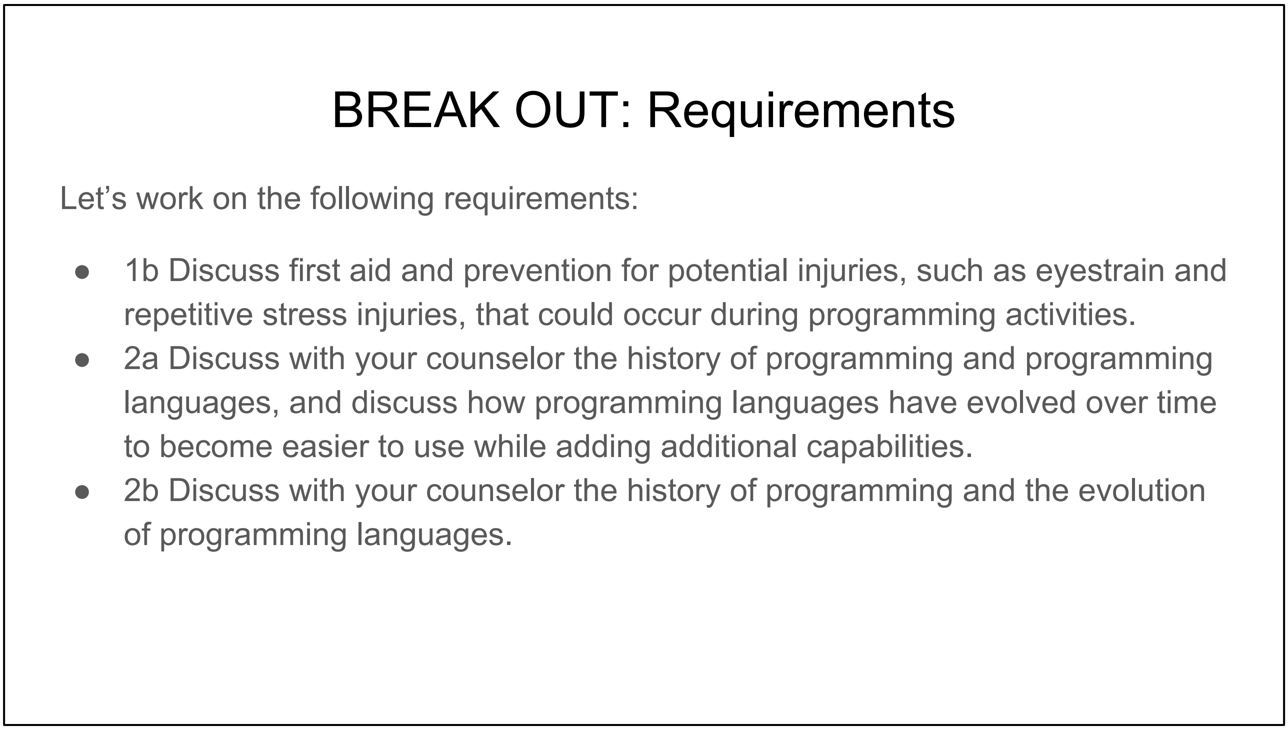


History

- Linear Programming
  - Sequential from beginning to end
- Structured Programming
  - Structures such as loops, goto, if/then and while statements can cause program execution to move around. (Remember we spoke about Subroutines)
  - Made it easier to write more complex programs
- Object Oriented Programming (OOP)
  - Pieces of code that are looked at as objects that interact with each other.
  - Smalltalk (1970's) was an early OOP Language
  - Java (1995) is an OOP Language (runs currently on 3 billion devices!)
  - Made it even easier to write complex programs. Objects can be more readily reused.
- println("Hello World!\n");

1. Go over Linear, Structured and Object Oriented Programming concepts and how this evolution has made it easier to write programs.

2. Explain to the Scouts what the "Hello World!" Program is.



BREAK OUT: Requirements

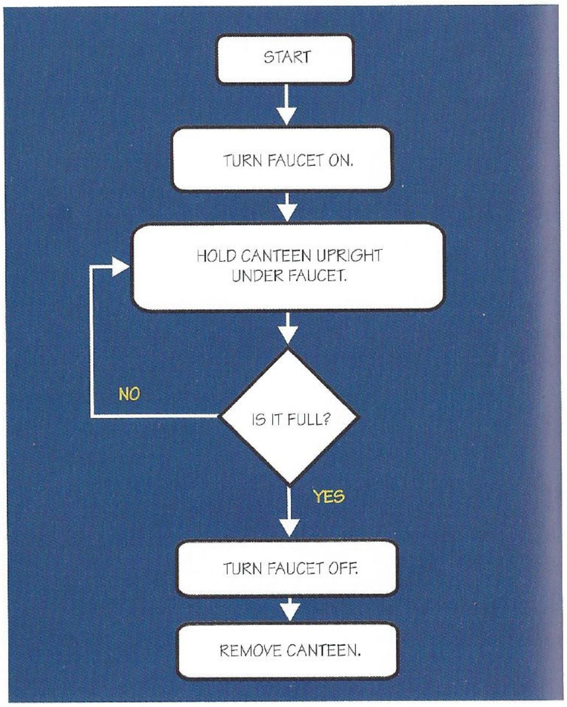
Let’s work on the following requirements:

- 1b Discuss first aid and prevention for potential injuries, such as eyestrain and repetitive stress injuries, that could occur during programming activities.
- 2a Discuss with your counselor the history of programming and programming languages, and discuss how programming languages have evolved over time to become easier to use while adding additional capabilities.
- 2b Discuss with your counselor the history of programming and the evolution of programming languages.

What is Programming?

“Fill your canteen with water”

- Flow Chart
- Rectangles (actions)
- Diamonds (decisions)



START

TURN FAUCET ON.

HOLD CANTEEN UPRIGHT UNDER FAUCET.

IS IT FULL?

NO

YES

TURN FAUCET OFF.

REMOVE CANTEEN.

1. Most of this section in the pamphlet was covered in my introduction. Concentrate only on flow charts and state diagrams.

2. Explain to the Scouts the challenge of “Step-by-Step Communication” and how a computer needs to be told exactly how to do something when writing a program hence the importance of planning and writing down steps before you actually begin writing code.

3. Talk a little about the “Fill your canteen with water” flow chart and how it would be helpful to have this if you needed to program a computer to complete this task.

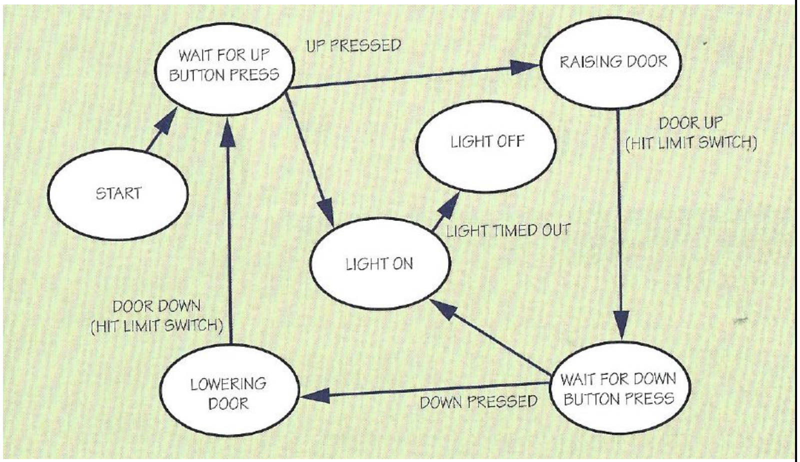
What is Programming?

State Diagrams

- Describes programs that do tasks sequentially
- State Machines

Plan the work, Work the Plan

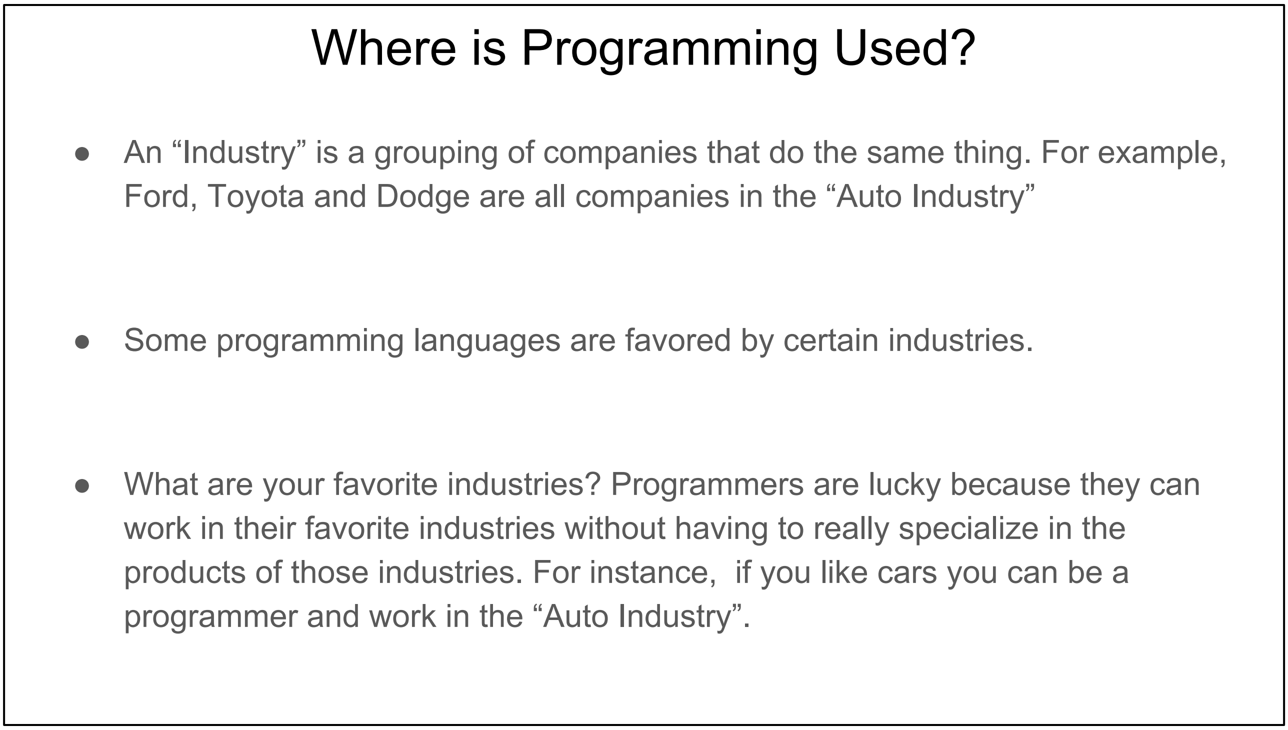
- Pseudocode
- Formal Requirements
- Algorithms



Talk about how a state diagram can be used to define state transitions and how I have used them in the real world: network handshaking communications. This kind of state diagram is called a “State Machine” which is a *design pattern*.

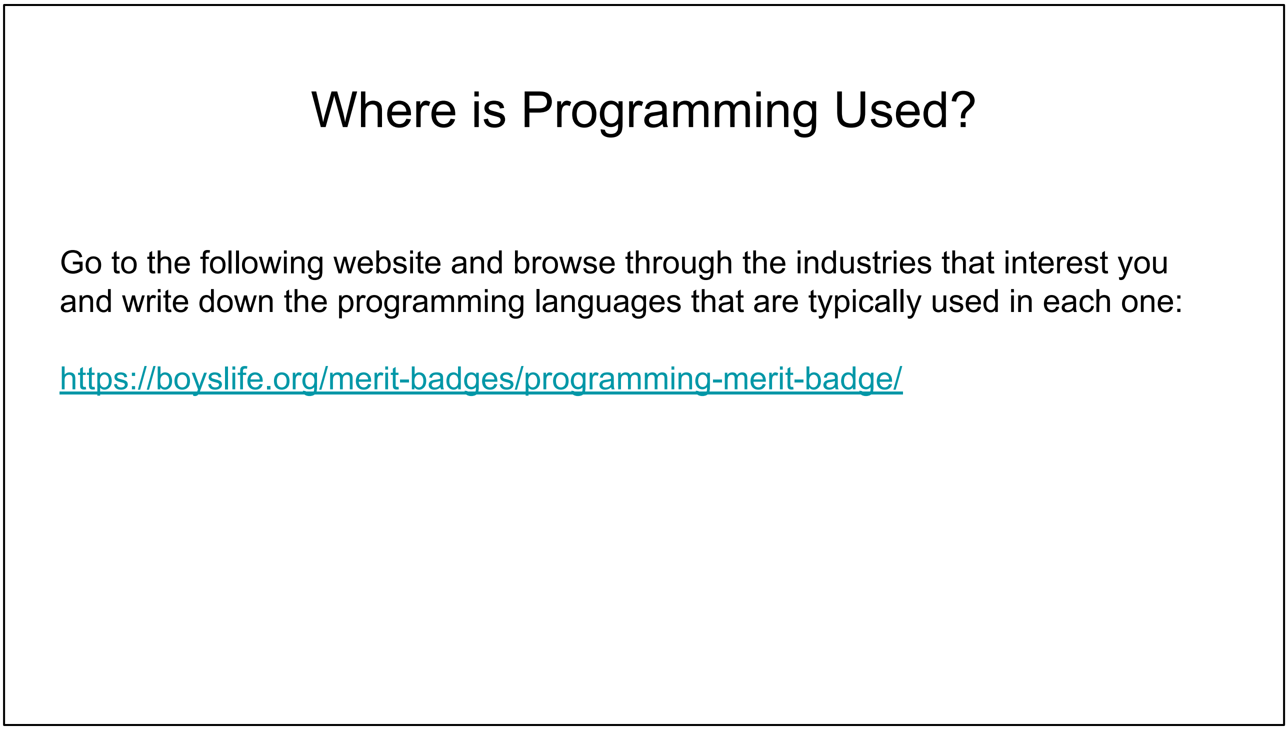
Explain a little to the Scouts about what design patterns are and how they promote reuse.

<<< Ask the Scouts to plan a very short program using one of the techniques described above. >>>



Where is Programming Used?

- An “Industry” is a grouping of companies that do the same thing. For example, Ford, Toyota and Dodge are all companies in the “Auto Industry”
- Some programming languages are favored by certain industries.
- What are your favorite industries? Programmers are lucky because they can work in their favorite industries without having to really specialize in the products of those industries. For instance, if you like cars you can be a programmer and work in the “Auto Industry”.



Where is Programming Used?

Go to the following website and browse through the industries that interest you and write down the programming languages that are typically used in each one:

https://boyslife.org/merit-badges/programming-merit-badge/

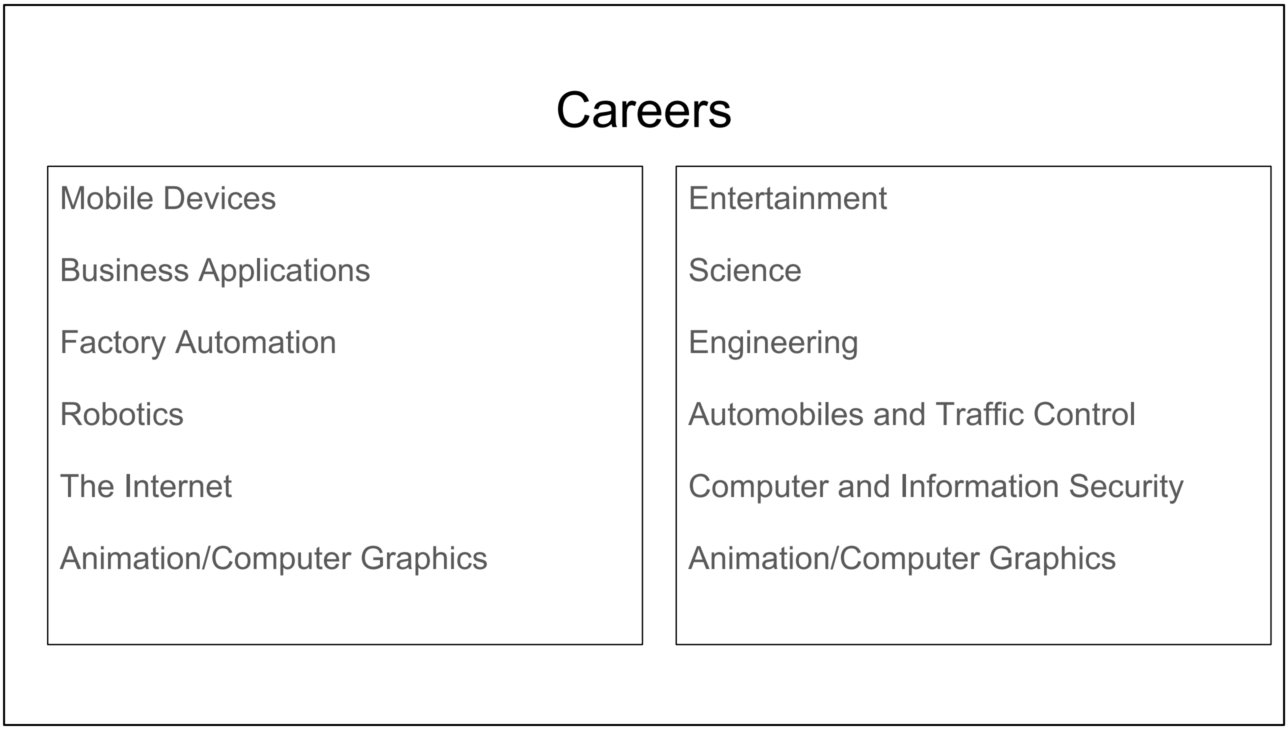
Show the Scouts the industry selection web page at boyslife.org that displays the programming languages for each industry. Have them choose 10 programming languages from the industries that interest them. If they brought a chromebook or computer now is the time for them to use it, otherwise, let them use one of the loaners.

https://boyslife.org/merit-badges/programming-merit-badge/

This slide is looking to help them complete requirements:

- *3a. Create a list of 10 popular programming languages in use today and describe which industry or industries they are primarily used in and why.*
- *3b. Describe 3 different programmed devices you rely on every day.*

Once the Scouts create their list talk about the programming languages they selected and why they are suited for the corresponding industries. Explore via google if needed.



Careers

- Mobile Devices
- Business Applications
- Factory Automation
- Robotics
- The Internet
- Animation/Computer Graphics

- Entertainment
- Science
- Engineering
- Automobiles and Traffic Control
- Computer and Information Security
- Animation/Computer Graphics

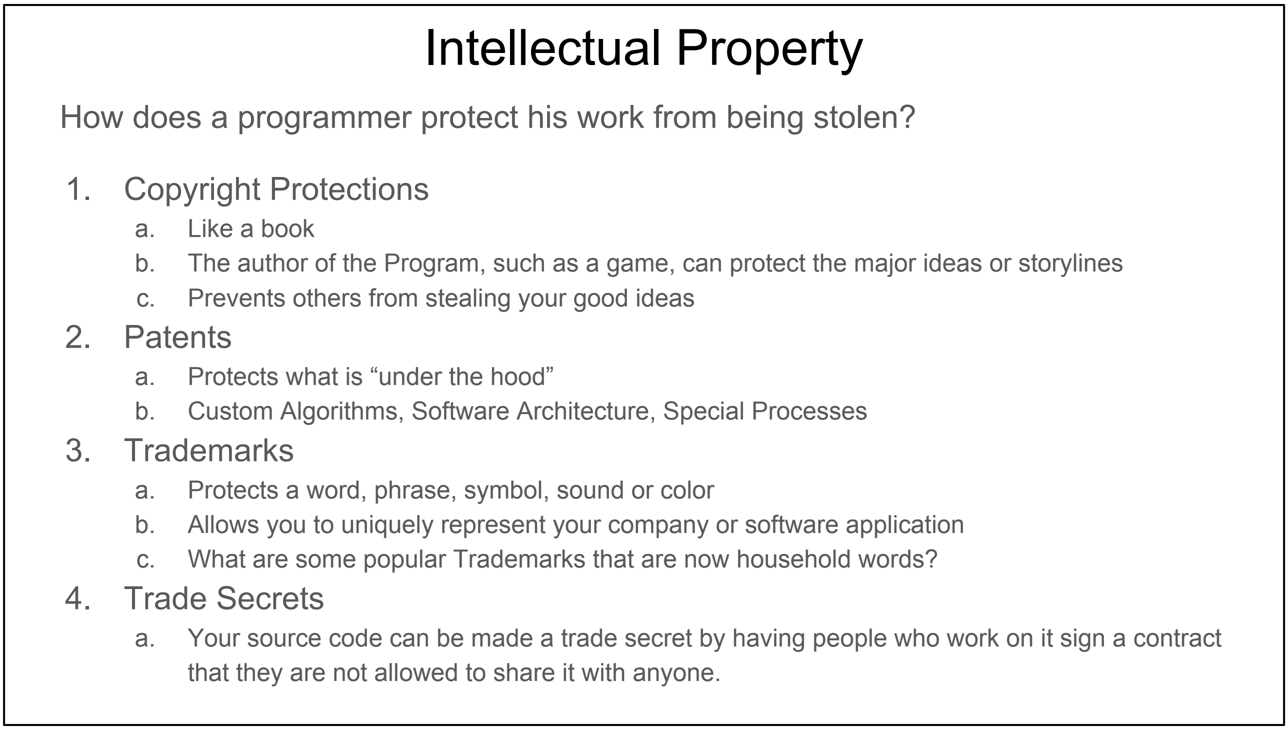
Explain to the Scouts that the Programming Merit Badge pamphlet has a really good chapter devoted to “Where is Programming Used?” that explains the above industries and how they rely on computer programmers.

Computer programming is a great career because it can be applied to many different industries. This has the following advantages:

- There are lots of jobs available
- You can work anywhere in the world
- You can pick the industry you like and get job there

Summarize this section in the pamphlet as follows:

- Mobile Devices
- Business Applications
- Factory Automation
- Robotics
- The Internet
- Animation/Computer Graphics
- Entertainment
- Science
- Engineering
- Automobiles and Traffic Control
- Computer and Information Security
- Animation/Computer Graphics

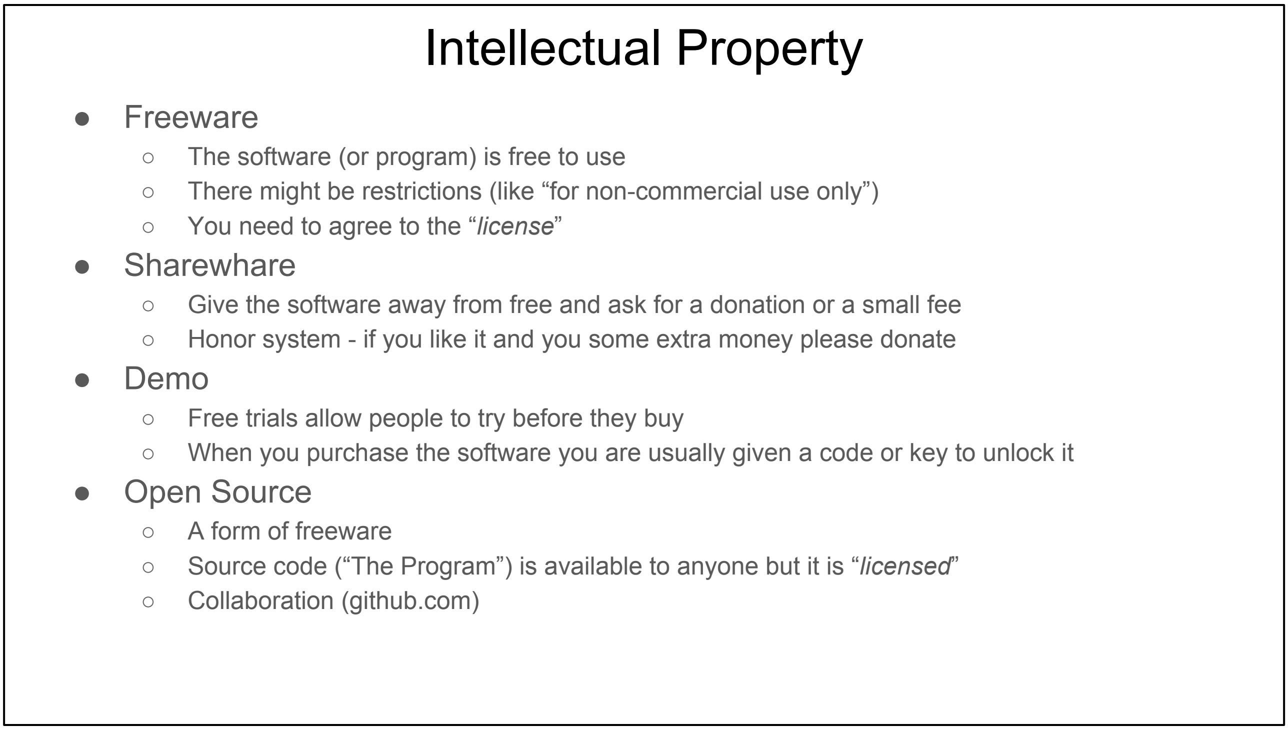


Intellectual Property

How does a programmer protect his work from being stolen?

1. Copyright Protections
   a. Like a book
   b. The author of the Program, such as a game, can protect the major ideas or storylines
   c. Prevents others from stealing your good ideas
2. Patents
   a. Protects what is “under the hood”
   b. Custom Algorithms, Software Architecture, Special Processes
3. Trademarks
   a. Protects a word, phrase, symbol, sound or color
   b. Allows you to uniquely represent your company or software application
   c. What are some popular Trademarks that are now household words?
4. Trade Secrets
   a. Your source code can be made a trade secret by having people who work on it sign a contract that they are not allowed to share it with anyone.

- Explain to the Scouts about the fine line of software patents
- Talk to the Scouts about the Paltalk vs Microsoft patent lawsuit that I was involved in.
- Talk about each type of intellectual property and have them think of examples.

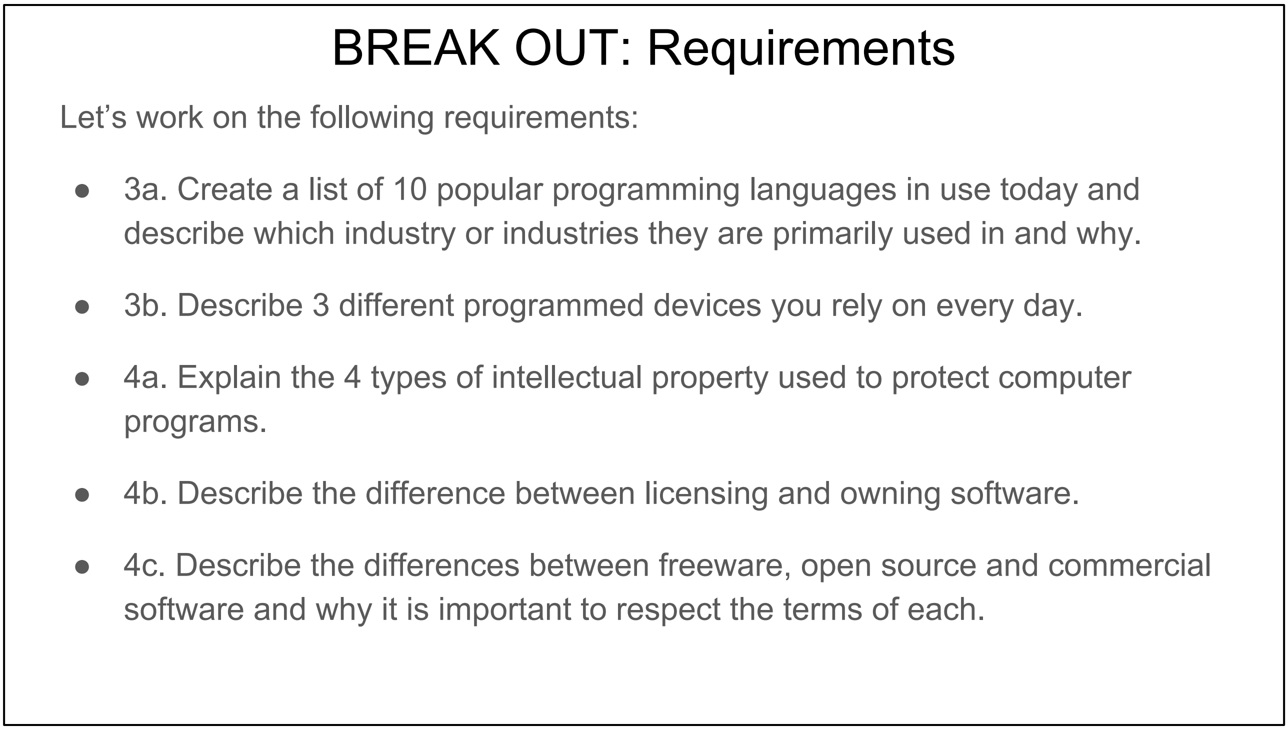


Intellectual Property

- Freeware
  - The software (or program) is free to use
  - There might be restrictions (like “for non-commercial use only”)
  - You need to agree to the “*license*”
- Sharewhare
  - Give the software away from free and ask for a donation or a small fee
  - Honor system - if you like it and you some extra money please donate
- Demo
  - Free trials allow people to try before they buy
  - When you purchase the software you are usually given a code or key to unlock it
- Open Source
  - A form of freeware
  - Source code (“The Program”) is available to anyone but it is “*licensed*”
  - Collaboration (github.com)

- Talk to the Scouts about the 4 types of Intellectual property and give them examples

- Explain to them the advantages of “Open Source” licensing and how the community is responsible for great software such as:
  - Firefox web browser
  - Open Office
  - Notepad++
  - 7-Zip
  - Linux (open source operating system)
  - Ubuntu (a *flavor* of linux)

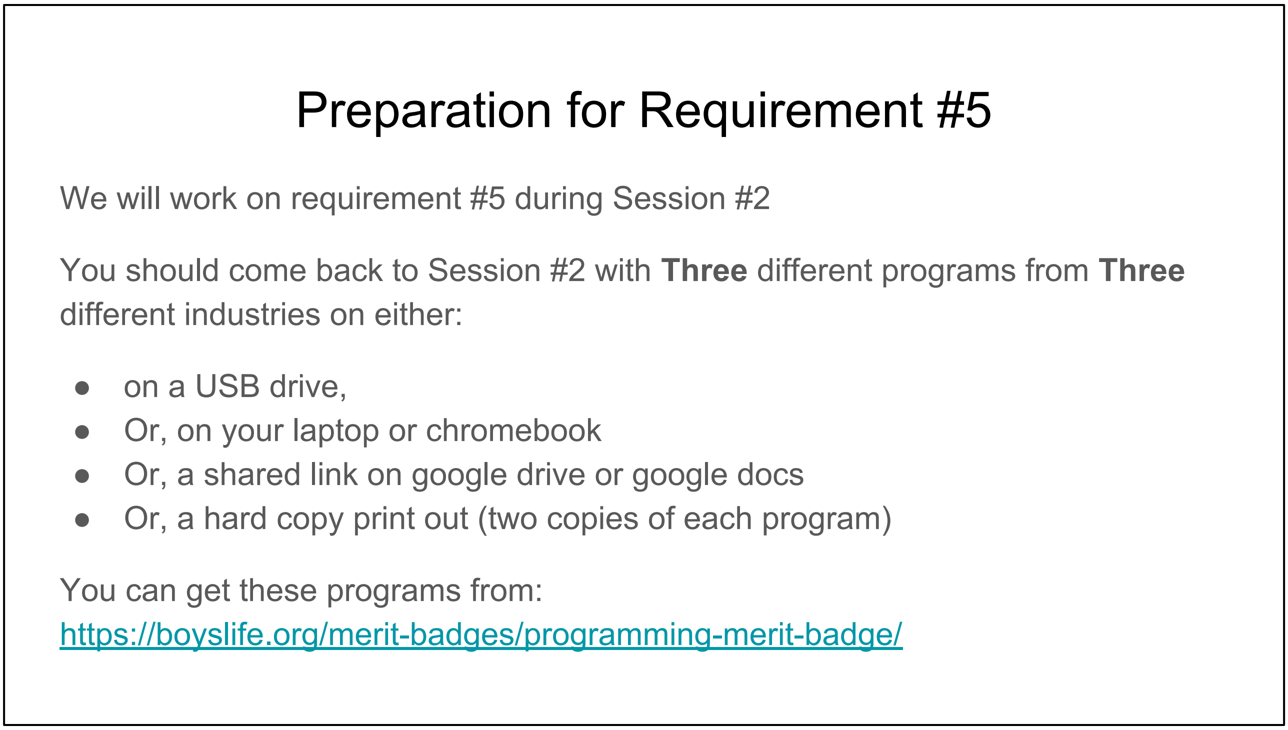


BREAK OUT: Requirements

Let’s work on the following requirements:

- 3a. Create a list of 10 popular programming languages in use today and describe which industry or industries they are primarily used in and why.
- 3b. Describe 3 different programmed devices you rely on every day.
- 4a. Explain the 4 types of intellectual property used to protect computer programs.
- 4b. Describe the difference between licensing and owning software.
- 4c. Describe the differences between freeware, open source and commercial software and why it is important to respect the terms of each.

Have them ponder a little on 4b and 4c. Let them make the associations by discussing amongst themselves.



Preparation for Requirement #5

We will work on requirement #5 during Session #2

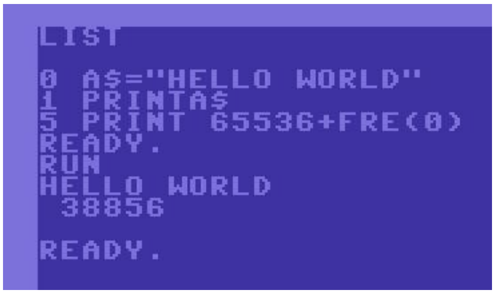
You should come back to Session #2 with **Three** different programs from **Three** different industries on either:

- on a USB drive,
- Or, on your laptop or chromebook
- Or, a shared link on google drive or google docs
- Or, a hard copy print out (two copies of each program)

You can get these programs from:
https://boyslife.org/merit-badges/programming-merit-badge/

- Talk to them about requirement #5. Refer to my Wood Badge journal entry that talks about the requirement.
- If there is still time in Session #1 you can have the Scouts start selecting their programs and even start working on the requirement.
- Let the Scouts know that if they can not make Session #2 they can fulfil the final requirement and, with a Buddy, schedule a time to meet with me.

This Concludes Session #1



```
LIST

0 A$="HELLO WORLD"
1 PRINTA$
5 PRINT 65536+FRE(0)
READY.
RUN
HELLO WORLD
 38856

READY.
```

Session 2 - Lab for Requirement #5

Can you guess which computer language uses this guy as a mascot?

- This entire session will be devoted to working on Requirement #5.

- Even though the Scouts will have been instructed to do this requirement on their own and have it reviewed at this session I also give them the option to complete it here from start to finish. All of them chose the latter.

- Worksheets can be downloaded here: https://drive.google.com/open?id=1Y032YQHHUVeCQZaAZVWKTLibAUNTjmG0uUV-mXU5XKw.

- Each Scout will need 3 copies of the above worksheet to complete this requirement. One for each program/industry.

- The idea here is to bring at least 3 computers (one for each of their worksheets) and set them up running different environments. I used 3 old laptops and found suitable programming environments for each. I also found that when speaking to others about this many offered old laptops of their own for us to use. So ask around if you don't have any.

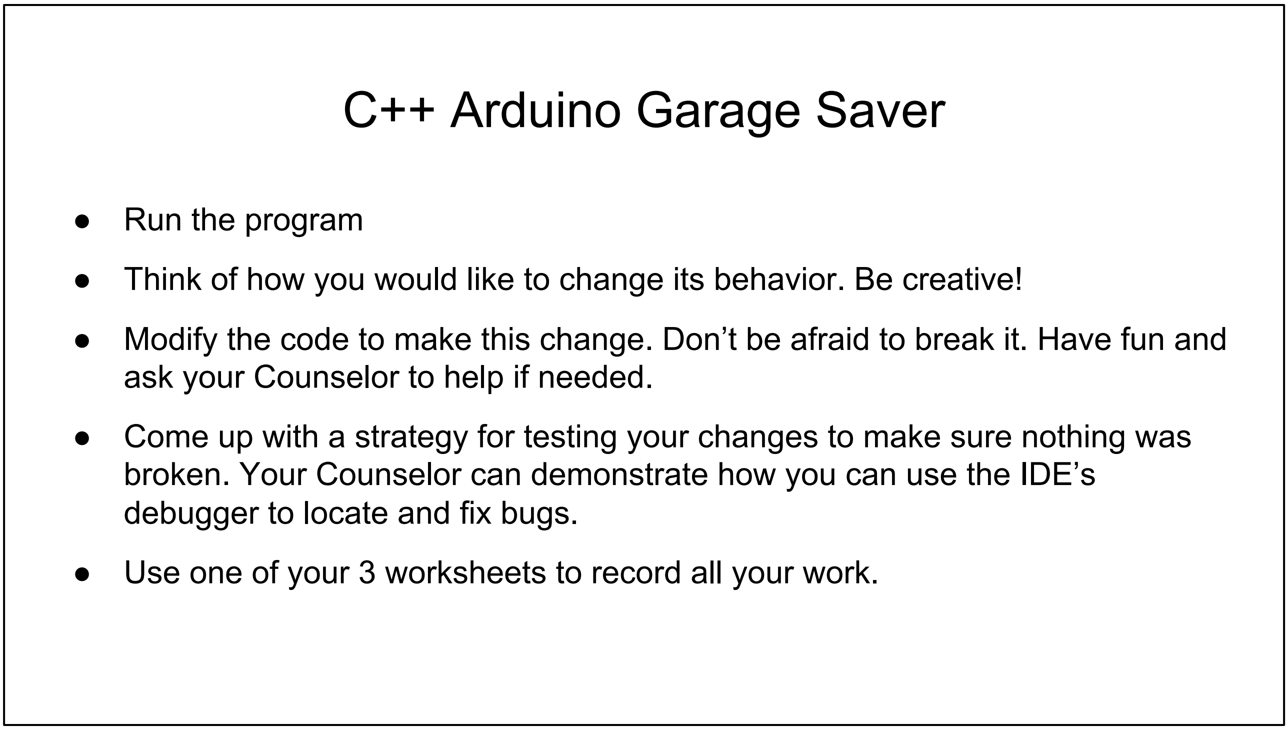
- Have the Scouts pair up if possible but still maintain their own worksheets. They'll each work at a workstation and then rotate once they have completed the tasks in their worksheet.

- Scouts should select an industry suitable for each workstation. The industries

- and programs can be researched here: https://boyslife.org/merit-badges/programming-merit-badge/. The programs and languages i have chosen for the workstations span many industries.

- The following slides explain the programs and environments running on each workstation. Go over them with the Scouts and explain what they have to do for each. Feel free to choose your own programs or use the ones given on the boyslife.org site above. Resources to support the languages and programs I have chosen for the slides below can be downloaded here: https://drive.google.com/open?id=1mEDn3br7TTy340uHLY9nkoUt9nwvLfRA

- Final note here - have 2 copies of the source code on each workstation. One for reference and the other for the Scouts to modify. Once a set of Scouts has moved on to the next workstation save their work to another folder and revert the changes so the next Scout can start fresh..



C++ Arduino Garage Saver

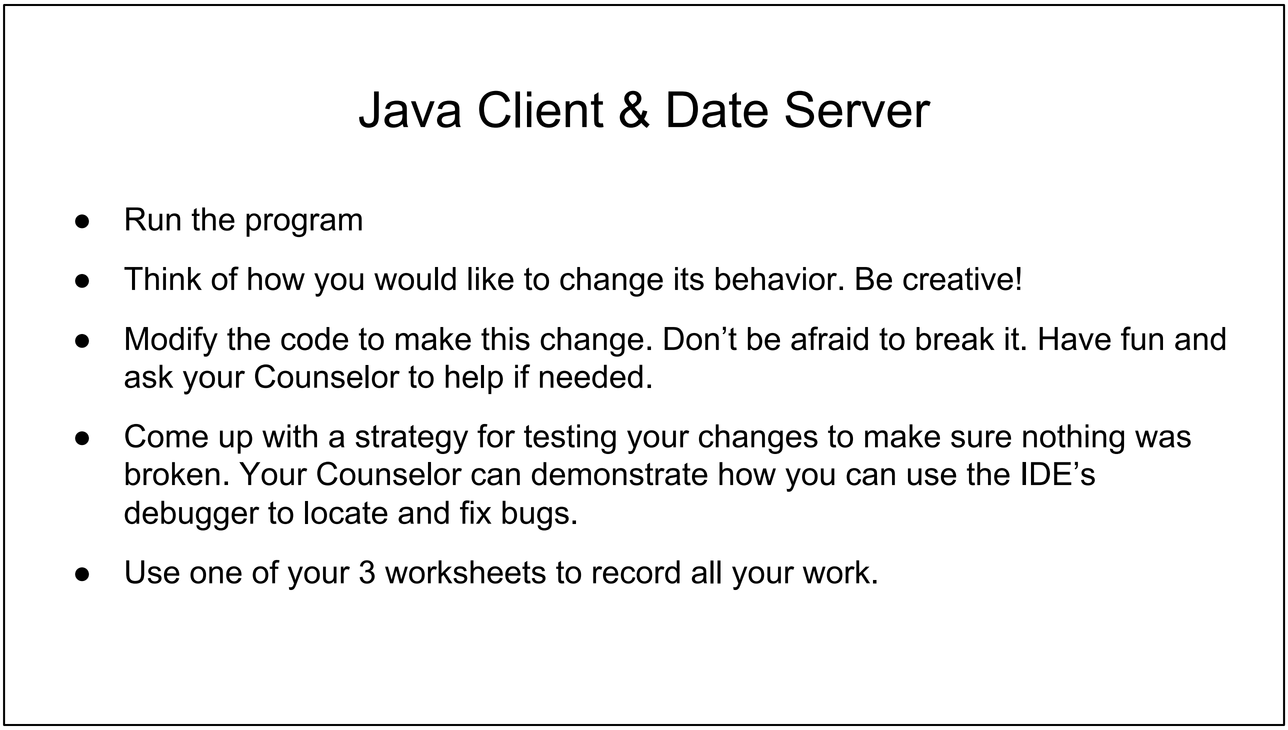
- Run the program
- Think of how you would like to change its behavior. Be creative!
- Modify the code to make this change. Don’t be afraid to break it. Have fun and ask your Counselor to help if needed.
- Come up with a strategy for testing your changes to make sure nothing was broken. Your Counselor can demonstrate how you can use the IDE’s debugger to locate and fix bugs.
- Use one of your 3 worksheets to record all your work.

- A lot of cars don’t have proximity sensors. This feature is very useful when parking your car in a garage because it will beep if you get too close to the wall. In this example we examine a program that instructs an arduino microcontroller to sound an alarm if a toy car comes too close to the back of a cardboard box. Hence simulating a car parking in a garage.

- Download the latest version of the Arduino IDE here: https://www.arduino.cc/en/Main/Software and set it up on a computer. You can use an old laptop running windows but if the version is too old you might want to try installing linux on it. I’ve found a lot of the light linux builds available today run great on these old laptops.

- As stated in the previous slide the garage saver program can be found here in the Arduino directory: https://drive.google.com/open?id=1mEDn3br7TTy340uHLY9nkoUt9nwvLfRA

- It’s best to copy the entire directory structure in the above link twice on each workstation. One for a reference and the other for the Scouts to modify. Revert the changes back to the reference copy after the Scout moves on to the next workstation.



Java Client & Date Server

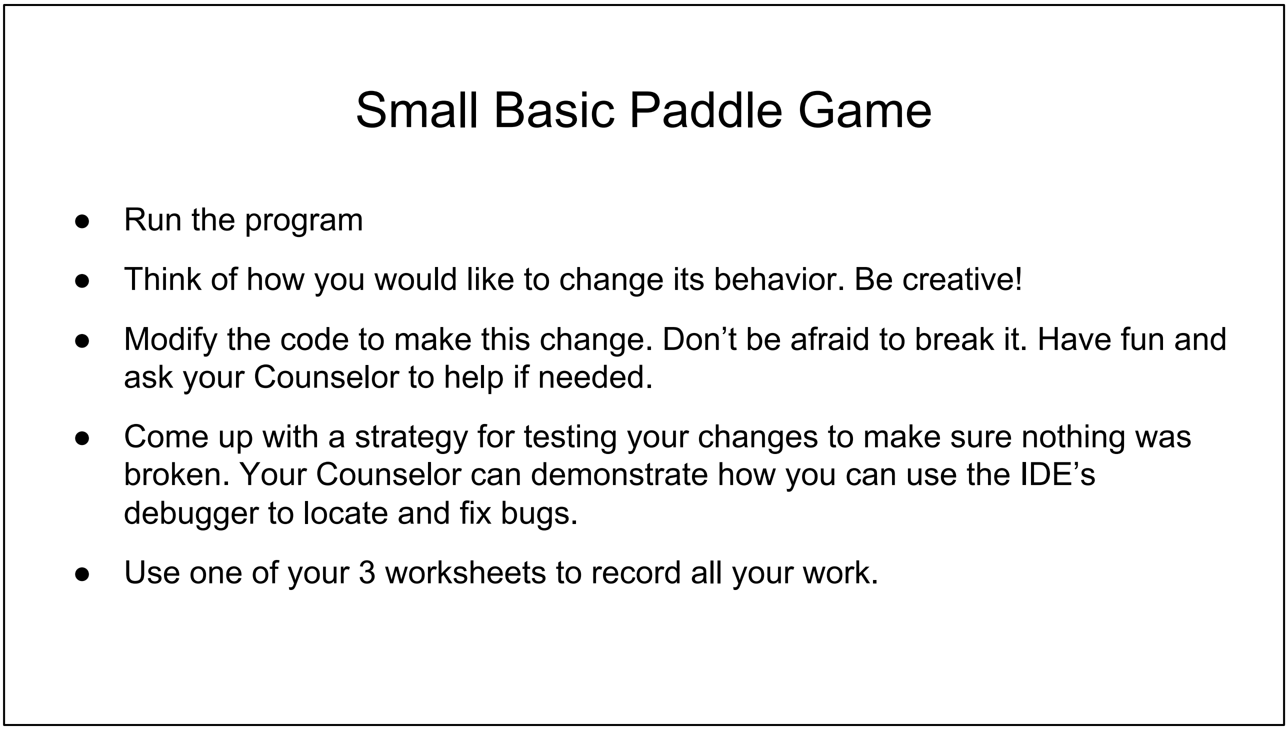
- Run the program
- Think of how you would like to change its behavior. Be creative!
- Modify the code to make this change. Don’t be afraid to break it. Have fun and ask your Counselor to help if needed.
- Come up with a strategy for testing your changes to make sure nothing was broken. Your Counselor can demonstrate how you can use the IDE’s debugger to locate and fix bugs.
- Use one of your 3 worksheets to record all your work.

- This is a simple Client and Server implemented in Java. The server runs on port 8080 and accepts tcp/ip connections. When a client connects, it sends it the current date/time and closes the connection.

- For an IDE I used the free community edition of intellij. You can download it here: https://www.jetbrains.com/idea

- As stated in the previous slide the source code can be found here in the Java directory: https://drive.google.com/open?id=1mEDn3br7TTy340uHLY9nkoUt9nwvLfRA

- Again, it’s best to copy the entire directory structure in the above link twice on each workstation. One for a reference and the other for the Scouts to modify. Revert the changes back to the reference copy after the Scout moves on to the next workstation.



Small Basic Paddle Game

- Run the program
- Think of how you would like to change its behavior. Be creative!
- Modify the code to make this change. Don't be afraid to break it. Have fun and ask your Counselor to help if needed.
- Come up with a strategy for testing your changes to make sure nothing was broken. Your Counselor can demonstrate how you can use the IDE's debugger to locate and fix bugs.
- Use one of your 3 worksheets to record all your work.

- This is a paddle game implemented in Microsoft Small Basic. There are lots of cool example programs on their site. The beauty of this is I was able to get it to run nicely on an old windows xp laptop.

- Download the IDE from https://smallbasic-publicwebsite.azurewebsites.net/

- As stated in the previous slide the source code can be found here in the Java directory: https://drive.google.com/open?id=1mEDn3br7TTy340uHLY9nkoUt9nwvLfRA

- More info about running it on xp is here: http://smallbasic.com/faq.aspx. To do that you'll need .net 3.5 and a few other files located in the resources link above in a folder named small-basic-for-xp.

- Again, it's best to copy the entire directory structure in the above link twice on each workstation. One for a reference and the other for the Scouts to modify. Revert the changes back to the reference copy after the Scout moves on to the next workstation.

Feedback

YOUR FEEDBACK MATTERS

At the end of session #2 hand out feedback forms here:

https://docs.google.com/document/d/1IyVvZobaMvzqZSjwRHaMY48Fk9T8jK6Ss6dqIgU_a5Q/edit?usp=sharing