

**Week 3: Software**

For a class of 12 students, grades 6-9

**Materials Needed**

Check In Sheet

Student Nametags

Week 3 Slides displayed on projector

For classroom portion:

Printouts of:

ASCII Alphabet in Binary charts

Letter Encoding Sheet (for “Encoding Your Initials in Binary” activity)

Sentence Encoding Sheets (for “Coded Messages” activity)

4 or 5 unique messages for Coded Messages Activity, sentences of 60 characters or less

**(Expanded ASCII Characters chart?)**

5-6 Triage stations set up around the room (monitor, VGA, power, keyboard, mouse)

12 pens or pencils

The 12 computers triaged last week (for OS Load Activity)

12 Triage CDs

12 OS Load Instructions

12 OS Load USB Keys with Ubermix 1.5.1 or newer

For workshop portion:

For Triage plus OS Load: 12 identical computers, pre-tested, with separate optical drives, hard drives, and RAM

Additional triaged computers ready for OS Load (from OS Load queue)

* Avoid black HP 6000 computers. They have a different OS Load process

**BIG CHANGES**

**3 bios example stations in class, to rotate through in 3 groups, in the middle of the room**

**os and program info first and foremost**

**recommend scavenger hunt along the walls on computer stations, paired up with worksheet**

**install/uninstall will be on the slides?**

**different types of OS, programs for linux**

**see about reinforcing triage in week 4 instead of week 3, and have students load ubermix on**

**pre-prepped computers in warehouse after classroom portion/break**

**Note: Students will be *paired up* to work on the in-class and in-workshop activities**

**2:45 - Volunteer Assistants Arrive**

* Volunteers should sign in to timeclock and wear name badges

**3:15 - 3:30 - Student Check In**

* Students sign in at front desk and pick up nametag, then go to classroom

**3:30 - Classroom Portion Begins**

Start Week 3 Slides presentation

Introductions - Instructors and Volunteers

Review Classroom rules

Follow along with the instructors, don’t skip ahead

Questions are encouraged

Any questions about Week 2: Hardware?

**How Computers Think**

Computers think with only two numbers - ONE or ZERO. This is called **binary code**. You can also think of it like a light switch being ON or OFF.

Each one or zero is called a **Binary Digit**, or **Bit.**

**Data** is all the bits you send or receive from a computer. For example, a file you save on the hard drive is turned into a collection of bits by the computer, so it’s data.

But humans don’t think in binary code. We use more than 2 numbers and letters. So all the information we input to a computer has to be converted to binary. This is called **encoding**. Every **input** a computer receives - like typing a letter on the keyboard -- is **encoded** into bits.

The encoded data is processed by the CPU. The CPU keeps that letter in binary until it needs to be **output** back to a human. If you’re typing, the output would be the letter displayed on the monitor. Before it shows the letter to you, the computer has to convert it from binary back to an alphabet letter. That’s called **decoding**.

We’re going to try some encoding and decoding so you can see how a computer thinks. Keep in mind that it may take you a minute to encode or decode a letter, but a computer can do it billions of times a second.

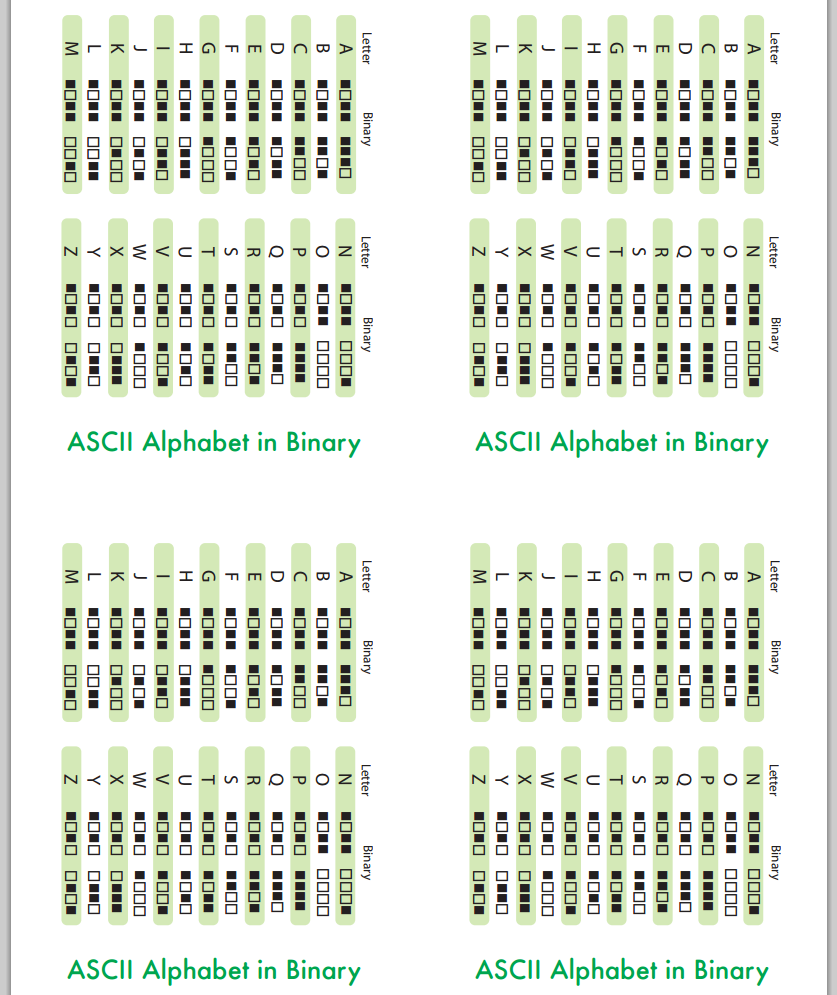
**>> Binary Encoding Activities**

*Adapted from Code.org’s Tinkersmith Binary Baubles module, https://code.org/files/CSEDbinary.pdf*

> Distribute ASCII Alphabet in Binary Charts, Letter Encoding Sheet, and Sentence Encoding Sheet to each student

When a computer encodes a letter into binary, it follows a standard pattern called **ASCII** that gives each letter a series of ones or zeroes. Each letter is 8 bits long and has a unique pattern so the computer can tell the letters apart. A series of 8 bits is also called a **byte**.

The ASCII Alphabet in Binary chart shows you the bits that make up each letter.



On the chart, a black box is a 1 and a empty box is a 0.

Look up K on the chart. Its binary encoding is 1-0-1-1--0-1-0-0

**> Initials Encoding Demo**

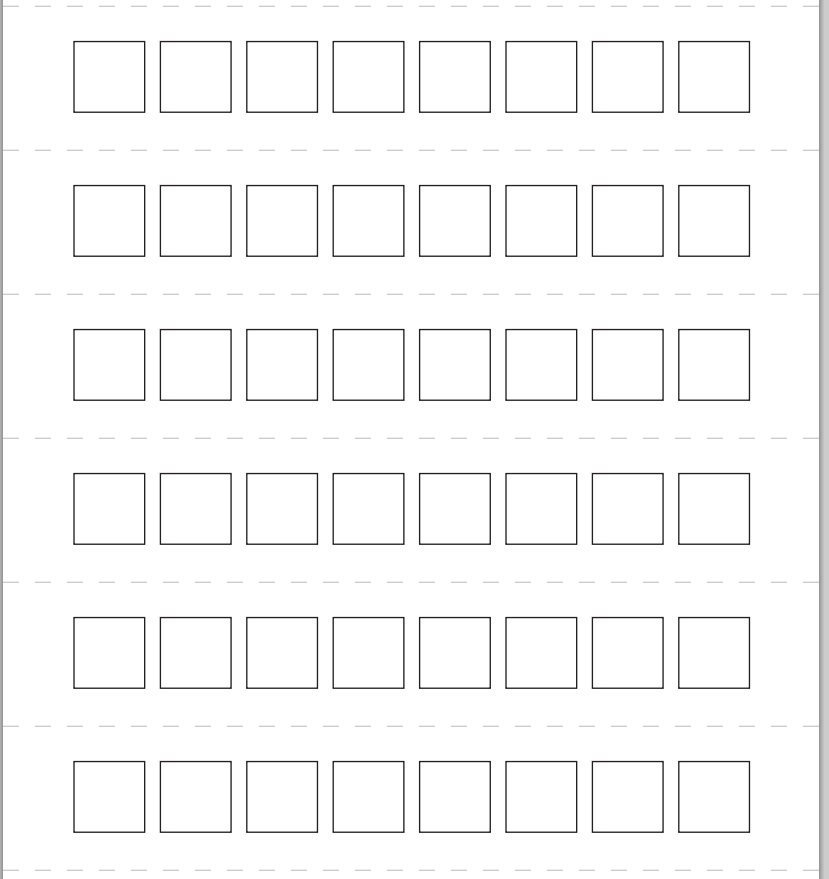
The instructor will use the chart to encode their own initials. Write down first initial next to one line on a **Letter Encoding Sheet**. Lookup the binary for that letter on the chart and fill in the same boxes on the first line. Do the same for the last initial.

This is an example of what a computer does when you type letters on the keyboard.

Now each student will converts their initials to binary the same way.

**>> Encoding Your Initials in Binary Activity**

Write each of your initials next to a line on the Letter Encoding Sheet. Using the ASCII chart to look up your first initial. Fill in the boxes to match the chart.



**> Decode Initials Activity**

Each student should pass their Letter Encoding Sheet to their neighbor. Use the chart to match the patterns and find the correct letter. Write the letter on the Letter Encoding Sheet. Now you’ve decoded their initials.

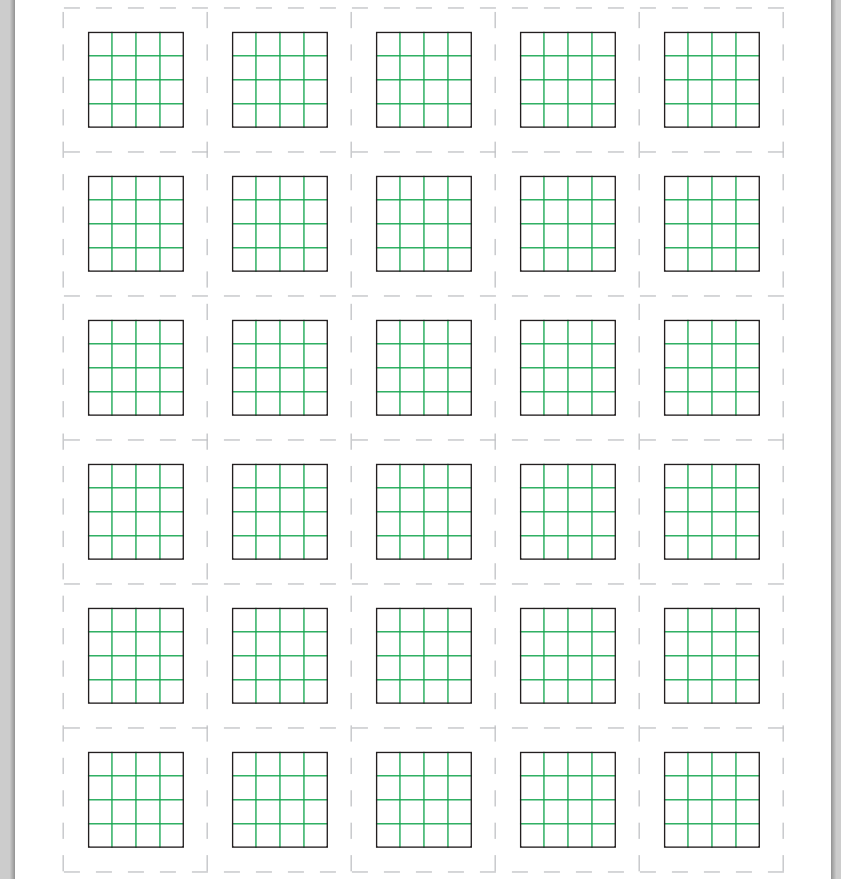
This is similar to what a computer does when it displays a letter on the monitor.

**>> Coded Messages Team Activity**

Divide the class into teams of 2 (5 or 6 teams total).

Teams should sit at each computer station in the classroom.

Give each team a typed unique Message to encode. They will encode each letter using the Sentence Encoding Sheet. Two letters go in each square.



**(Use more complex message with capitals and punctuation? Will need to hand out Expanded ASCII Characters chart instead.)**

**(Update this: Actual ASCII binary for Space is 00100000)**

Use 8 blank boxes, or 0-0-0-0--0-0-0-0, for a SPACE character. Make sure the teams don’t write their message on their sheet. They should write their names on the sheet.

After the team has encoded their message, they pass the sheet to another team. That team then decodes their sentence and writes down the result on the sheet.

Compare the resulting decoded message to the original message (this is why the encoding team writes their names on the sheet). Were there any differences? If so, were the errors in the encoding or the decoding?

When computer data is bad, it’s called corrupted. This can happen if your RAM or hard drive is bad, or if a program doesn’t encode or decode data right. It can cause your computer to crash or act strange.

**(Break here after the encoding activities instead of after in-classroom OS Load activity?)**

**Software**

Software is binary code that gives the computer instructions. It uses the CPU, internal components, and input and output peripherals to accomplish a task.

There are two main types of software: operating systems (OS) and programs.

**Operating Systems**

Operating systems are complex pieces of software that supports a computer's basic functions, like drawing graphics on the monitor, saving files, running programs, and controlling peripherals.

OSes are big and complex and take a lot of work to create. So there are only a few OSes that are widely used. Desktop computers use different OSes than tablets and smartphones.

**> Question: What are the operating systems you have heard of?**

**> Answers: Windows, Mac OS, Linux, Android, iOS, ChromeOS on Chromebooks**

**Windows**

Windows is a very common and popular OS. It was first released in 1985. Most desktop computers use Windows. It runs on computers made by many different companies, like Dell and HP. It costs about $100, but it’s included in the price of most new computers.

**Mac OS**

Mac OS is only for computers made by Apple. Mac OS was first released in 1984. It’s used by a lot of people for creative jobs. It’s used on about 10% of computers. Mac OS comes with every Apple computer, so it’s always built in to the computer’s price.

**Linux**

The computers that you are refurbishing have a Linux operating system. Linux is a free and open-source OS. Linux is free to download and use and it runs on computers from all different companies. It’s developed by programmers around the world who add to and fix bugs in Linux and then share with everyone else. Anyone can see the software source code that makes up Linux. Linux programmers don’t all work for one company, but there are different versions of Linux, like Ubuntu and Red Hat. These versions look different and are for different purposes, like a server or a desktop PC.

**Programs**

Programs (or application or app) are software that do a specific job. A calculator program’s job is just to do math. You can have dozens or hundreds of different programs on any computer. Which programs you use depends on what you use your computer for: typing reports, games, graphics, browsing the Internet.

Programs have to be run on the right OS. They won’t work on a different OS unless the programmer has written a different version. It’s like Xbox and Playstation games. Each game is a program, and a Xbox game won’t run on a Playstation. If the game developer wrote a different version of the game for each console, you still can’t use the Xbox disc in the Playstation. So if you’re looking for a program for your computer’s OS, you need to get a version that is written specifically for it. For the computers you’re going to build, that OS is called Ubuntu Linux.

**Installing an Operating System**

You install, or load, an OS onto a computer by using a simple program on a USB drive or CD/DVD.

This will replace any OS already on the computer’s hard drive!

At Kramden we have an OS Load process that uses a USB drive with the Ubermix OS.

**>> OS Load Activity**

Each team should now follow the OS Load Procedures for their first computer.

* Check that the computer is clean inside and out and doesn’t have any loose or broken parts.
* Plug in all the peripherals.
* Plug in the Ubermix USB drive
* Boot up the computer.
* Wait until the Ubermix installation menu appears.
* Type “1”, then type “yes”.
* Wait until the installation program says it’s done.
* Remove the USB drive and restart the computer.
* Ubermix should start up. Type in the password (“password”).
* STOP.

When the OS Load is complete and the computers have booted into Ubermix, the teams should stop and listen to a short description of what Ubermix is:

**About Ubermix**

Ubermix is a version of Linux. It’s a great OS because it’s easy to use and runs fast on refurbished computers. It doesn’t get viruses or other bad programs. It comes with 60 programs already installed, including games, programs for typing, slideshows, and other schoolwork, Web browsers, and more.

Kramden Institute uses UberMix on almost all of its computers because it’s great for kids -- and adults, too. It lets you get all your school work done, browse the internet, and even play games made for Linux.

**>> OS Load Activity 2**

Teams should now complete the OS Load procedures.

**(Check these steps - what’s missing?)**

* Set time and date
* Restart once
* Shutdown

Then they should do OS Load on the second computer.

**(Save the career section for week 2?)**

**Careers in Tech Part 1 (What Tech Jobs are available?)**

**Careers in Tech Part 2 (How to prepare for a Tech Job)**

**4:30 - Break time**

Snacks and Bathroom break

**4:45 - Workshop Portion begins**

Lead class to Warehouse Final Test area - Far left benches

**(5:00 - Wednesday Work Night volunteers will enter work area)**

In the same teams, students will triage 2 additional computers in the workshop and complete OS Load on them. If time permits, they can do the OS Load process on more computers from the OS Load queue. Reinforce the Troubleshooting steps if there are problems.

**Triage computer 3 and then do OS Load on it**

**Triage computer 4 and then do OS Load on it**

**Optional - OS Load additional computers**

If time permits, the students can install Ubermix on more computers. Avoid the black HP 6000s, since they don’t follow the normal OS Load process.

At the end of class, remind students that the computers they loaded Ubermix onto today will be donated to other students.

**EXTRAS / NOTES**

These encoded bits get “written” to and “read” from the computer’s storage

Hard Drive - permanent storage of files

RAM - temporary storage used by the CPU

On a hard drive, the bits are tiny atoms of platinum that are either positively or negatively magnetically charged

On RAM and USB Drives, bits are whether a tiny circuit has electricity or not

TYPE IN A NEW FILE -> SAVE FILE COMMAND -> ENCODE FILE TO BITS -> WRITE BITS

OPEN FILE COMMAND -> READ BITS -> DECODE FILE FROM BITS -> DISPLAY FILE

Visual:

Program Program Program

\ | /

OS

|

BIOS

Where’s the OS?

Files on the hard drive, usually

But it doesn’t have to come from the