Lesson 4: The CPU

Time: 1 60 minute class

Learning Objectives:

- Students will learn about the basic functionality of one of the most important components of a computer system (the CPU)
- Students will learn how the CPU carries out instructions in a logical manner
- Students will simulate a computer by executing the tasks of the cpu, memory and display

Instructional Activities: Teacher will introduce the lesson by playing a short video from code.org titled "How Computers Work: CPU, Memory, Input and Ouput. Teacher will emphasize the importance for students to understand how these parts work together to execute computer programs. After watching video, teacher and students will discuss some of the most important components of the CPU and how the cpu is able to carry out the instructions of a computer program. Teacher will display a copy of The Computer System Block Diagram and will illustrate the role of the inputs, outputs and memory.

Setup:
Divide the students up into groups of three.
Assign their roles.
(Remaining pair can be working as one person acting as "CPU" and the other as "Memory" and "Display".)

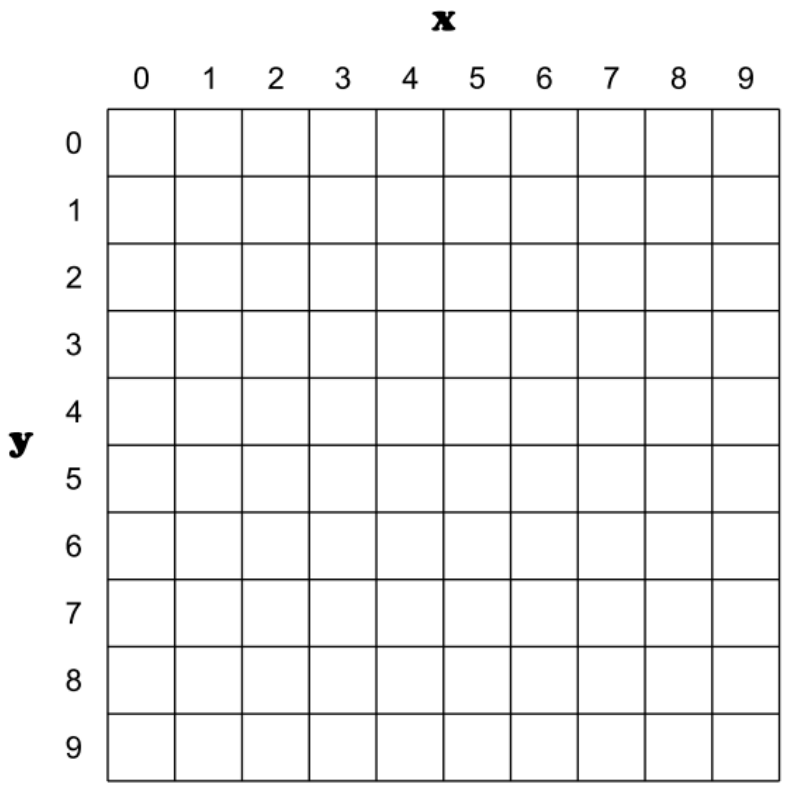
Name _____ Date _____

Display

How Computers Work

Greetings Display! Your job is to wait until the CPU tells you to plot an (**x**, **y**) point on the screen. When you are asked to plot a point, find the column that corresponds to the **x**-value and the row that corresponds to the **y**-value and plot the intersection by filling in the square.

Don't show the image to your group partner(s) until the CPU has finished running the entire program.



Student Roles:

CPU (working with the program counter sheet) will request instructions and data from memory, as required. These parts will also process any necessary calculations, writing results back to memory and instructing the display how to plot the data.
Memory (working with memory and program worksheet) keeps hold of the program and passes one instruction at a time onto the CPU. It also keeps a constant track of the current values of x and y.
Display (working with display grid) responds to the plot command from the CPU by marking the screen as instructed.

Student Activities:

Activity

Each student receives their relevant worksheets, the **Display** student making sure that nobody can see what he or she is plotting.

The **CPU** student begins, by asking for the first instruction from memory. Then it 'processes' it:
If the instruction starts with the work PLOT, the **CPU** needs to ask for the current value of x and the current value for y and then passes these values on to the **Display**, together with the plot command.

If the instruction begins with ADD or SUB, then the **CPU** needs to ask the **Memory** for the current value of the required variable(s). It then needs to calculate the result and tell the **Memory** the new value of the variable to be stored.
When the **Display** receives an instruction it needs to plot the value of x and the value of y on the display grid.

Example

For this example, assume that the program begins with the following instructions:

Add 4 to x

Add 6 to y

Plot (x,y)

The group should handle these instructions as follows:

- **CPU** asks for (fetches) the instruction (add 4 to x) from the **Memory**
- **CPU** asks the **Memory** for the value of x.
- **CPU** adds 4 to the value of x and tells the **Memory** to store the new value for x.

- **CPU** is now finished with this instruction and places a ‘tick’ next to the number 1, to keep track of which instruction to ask for next.
- **CPU** asks for (fetches) the instruction (add 6 to y) from the **Memory**
- **CPU** asks the value of y from the **Memory**.
- **CPU** adds 6 to the value of y and tells the Memory to store the new value for y.
- **CPU** is now finished with this instruction and places a ‘tick’ next to the number 2, to keep track of which instruction to ask for next.
- **CPU** asks for (fetches) the instruction (plot x, y) from the **Memory**
- **CPU** asks the value of x from the **Memory**.
- **CPU** asks the value of y from the **Memory**.
- **CPU** tells the **Display** to plot the value of x and y.
- etc.

Formal Assessment:

Group Activity will be collected and graded based on this rubric

Learning Adaptations: Visuals, Handouts, Pair grouping,

Follow-up Discussion:

The purpose of the simulation exercise is to give the students a small taste of what it is that computers do.

The single-most important “take-away” from this exercise is the following:

The **CPU** had no idea what was being drawn on the **Display** - it was just mindlessly following the instructions in the program.

Computers do not “understand” what they are doing. Drawing a picture of a cat is the same (from the computer's perspective) as drawing a picture of a dog - it's just a series of instructions to execute.

Since computers are simply executing the instructions in the program, it is the programmer's responsibility to write the program correctly. If there is a mistake in the program, the computer will still go ahead and try to execute the program as written.