SJA STEM PROGRAM

Rev 0.0.1

Contents

[Program Requirements 2](#_Toc475698964)

[Program Overview 2](#_Toc475698965)

[Week One – Introduction to Raspberry Pi and Kano (~1hr 30min – 2hrs) 3](#_Toc475698966)

[Learning Review 3](#_Toc475698967)

[Week Two – Introduction to Raspberry Pi and Kano (~1hr 30min – 2hrs) 4](#_Toc475698968)

[Learning Review 4](#_Toc475698969)

[Week Three – Introduction to Programing for Artists (~1hr 30min – 2hrs) 5](#_Toc475698970)

[Lesson Summary 5](#_Toc475698971)

[Learning Review 5](#_Toc475698972)

[Week Four – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 1 6](#_Toc475698973)

[Lesson Summary 6](#_Toc475698974)

[Learning Review 6](#_Toc475698975)

[Week Five – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 2 7](#_Toc475698976)

[Lesson Summary 7](#_Toc475698977)

[Learning Review 7](#_Toc475698978)

[Week Six – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 3 8](#_Toc475698979)

[Lesson Summary 8](#_Toc475698980)

[Learning Review 8](#_Toc475698981)

[Week Seven – Linux Operating Systems, Taking Control of the Terminal (~1hr 30min) 9](#_Toc475698982)

[Lesson Summary 9](#_Toc475698983)

[Learning Review 9](#_Toc475698984)

[Week Eight – Robotics Programming with Blockly and Dash (~1hr 30min) 10](#_Toc475698985)

[Learning Review 10](#_Toc475698986)

[Week Nine – Navigate the Robot Maze (~1hr 30min+) 11](#_Toc475698987)

# Program Requirements

This program does not have a prior programing experience requirement. The program can be taken by students who have prior programming experience as well as those who have never been exposed to programming.

* Students should be able to read.
* Students should be able to add, subtract, multiply, and divide.
* Students should be familiar with inequalities or at least understand them.
* Students should be able to use a keyboard and a mouse.
* Access to Raspberry Pi 3 computers loaded with KanoOS.

# Program Overview

This document proposes a general course outline for the St. John the Apostille (after school) Robotics Program. This program was designed to support the greater Science Technology Engineering and Mathematics initiatives at the school. The program introduces learners to core computer science concepts as well as an introduction to computing.

The following concepts will be explored:

* Hardware as it relates to the Raspberry Pi 3
* Operating Systems (Windows, Linux, MacOS)
* Data Representation (What is data?)
* Number Systems (Base2, Base10, Base16)
* Embedded Systems vs General Purpose Computers
* Input and Output devices
* Procedural Programming (What are Functions?)
* Cartesian Coordinate System (Make Art Canvas)
* Programming Language Syntax (CoffeeScript vs. C++)
* Boolean Math (AND, OR, NOT)
* Truth Tables
* Venn Diagrams
* Programming Logic - Conditional Expressions (== , < , > , >= , <= , !=)
* Conditional Constructs
* Looping
* Strongly Typed vs Weakly Typed languages
* Command Line Interface (CLI) vs Graphical User Interface (GUI) (Terminal Quest)

Students are often excited with the prospect of a “Robotics Course” and it is important to convey the importance of programing in the field of robotics. Therefore, this course focuses more on programming concepts than physical robotics and mechanical engineering. Robotics is introduced by way of pre-built robots that can be programmed using the programming concepts learned throughout the course.

# Week One – Introduction to Raspberry Pi and Kano (~1hr 30min – 2hrs)

This lesson is as an introduction to the Raspberry Pi. Students should be able to answer the following questions after this session.

* What is the Raspberry Pi and why it was created?
	+ Input and Output Pins
	+ Processor
	+ USB Ports
	+ Ethernet Connector
	+ Sound IN/OUT
	+ HDMI
	+ Power Connector
* Baking a Pi [Configuration/Boot]
	+ Install SD Card / Connect USBs / HDMI Output / Power Up
	+ Initial Boot and Disk Resize
* Register a Kano Account

## Learning Review

Students should be able to identify the following:

* List 3 things they learned about the Raspberry Pi (during the discussion)
* What can be done with a Raspberry Pi?
* What is a Raspberry Pi?
* Why is it cool?
* How many USB ports are on a Raspberry Pi?
* How what connector is for Video Output?
* What can be done with the GPIO pins?
* Why does the Raspberry Pi need an SD card?

# Week Two – Introduction to Raspberry Pi and Kano (~1hr 30min – 2hrs)

This lesson is as an introduction to the Raspberry Pi. Students should be able to answer the following questions after this session.

* Introduction to Computing
	+ General Computer vs Embedded Systems
	+ Input and Output Devices
* Taking a Byte our of Data Representation
	+ Bit, Nybble, Byte, Word, DWord, QWord
	+ Counting Like a Computer
	+ Data Representation
	+ American Standard Code for Information Interchange
	+ Data Decoder Challenge

## Learning Review

Students should be able to identify the following:

* What is an input device? Give examples.
* What is an output device? Give examples.
* What is data representation and why is it important?
* What is base 10?
* What is base 2?
* What is MSB and LSB?
* Who is Allen Turing?

# Week Three – Introduction to Programing for Artists (~1hr 30min – 2hrs)

Find your artistic side by writing code to create art using Kano Make Art. Students will learn the basics of a coordinate based drawing system, navigating the drawing area, and using code as well as math to draw. Work at your own pace and explore the capabilities of Make Art.

Have the students open Make Art from the desktop and watch the intro video. After watching

the video, let the students play around with the game while you watch and direct those that ask for

help.

The game has instructions that they should be able to follow without much support, however; be

watchful for those that are slow and guide them.

## Lesson Summary

* Introduction to MakeArt
	+ CoffeeeScript and Javascript
	+ Define what a function is and it’s purpose
		- Discuss CoffeeScript function definitions
		- Discuss C++ function definitions and how it looks different
	+ Discuss functions used in MakeArt (line, move, circle, etc)
	+ Explain the coordinate plane used in MakeArt vs. typical one used in math
	+ Have students work through the basic MakeArt problems
	+ Have students work through medium MakeArt problems
* Flags of the world project.
* Review

## Learning Review

Students should be able to identify the following:

* 3 things they learned in the session
* Demonstrate navigation of the coordinate plane.
* Draw basic shapes square, circles, and ellipses.
* What are functions?
* What are inputs?
* Identify functions and parameters (inputs) eg. move, moveTo, circle, line
* Identify [0,0] , [250,250] and [500,500] on the canvas
* How do we set a background color?
* How do we set the foreground color?
* How do we draw a circle?
* What arguments does it take?
* What dose the stroke function do?
* How do you set the thickness?
* How do you set the color?
* How do you draw a line?
* What are the arguments?
* How do you move to a specific point?
* What are the arguments?
* How do you draw part of a circle?
* What are the arguments?

# Week Four – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 1

Explore more advanced programming concepts such as boolean logic, conditionals.

## Lesson Summary

* Introduction to Boolean Logic
	+ George Boole (Father of Boolean Algebra)
	+ Thinking in 1’s and 0’s … true or false
	+ Discuss Finite vs. Infinite (Discrete vs Calc)
* Logical Operations
	+ And
	+ Or
	+ Not
	+ Truth Tables for all discussed
* Logical Statements are True/False nothing in between.
* Discuss programming symbols: == , < , > , >= , <= , !=
* Why is logic needed?
	+ To control our program flow.
	+ What happens if your logic is bad?
* No More Training Wheels
	+ Review Functions
		- Discuss CoffeeScript function definitions (loosely typed)
		- Discuss C++ function definitions and how it looks different (strict typing)
	+ What is a variable?
		- Discuss CoffeeScript variable definitions (loosely typed)
		- Discuss C++ variable definitions and how it looks different (strict typing)
	+ My First Program with Variables Assignment – Let them change the variable names
	+ Call me if you need that done! – Function Assignment
		- Have them copy and modify the function example.
		- Have them change it to multiply or add a function to do that.
	+ Are you the same? – Equals Logic (==)
		- Have them copy and modify the code example.
		- Why is your output different?
	+ To Be or Not To Be – Not Equals Logic (!=)
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Are you smaller? – Less Than Logic (<)
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Are you larger? – Greater Than Logic (>)
		- Have them copy and modify the code example.
		- Why is your output different?

## Learning Review

* Who is the father of Boolean logic?
* What is a conditional? (give an example)
* What is the result of a condition?
* Why do we need logic?
* Give an example of AND logic.
* Give an example of OR logic.
* Give an example of NOT logic.

# Week Five – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 2

Explore more advanced programming concepts such as boolean logic, conditionals, and looping.

## Lesson Summary

* No More Training Wheels
	+ Review Functions
		- Discuss CoffeeScript function definitions (weakly typed)
		- Discuss C++ function definitions and how it looks different (strongly typed)
	+ What is a variable?
		- Discuss CoffeeScript variable definitions (weakly typed)
		- Discuss C++ variable definitions and how it looks different (strongly typed)
	+ My First Program with Variables Assignment – Let them change the variable names
	+ You must be at least 42 inches to ride! – Greater Than or Equal To Logic (>=)
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Your bag must weigh 52lbs or less! – Less Than or Equal To Logic (<=)
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Making Decisions – If, Else If, Else
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Rinse and Repeat – The for loop
		- Have them copy and modify the code example.
		- Why is your output different?
	+ While the water was running… While Loop
		- Have them copy and modify the code example.
		- Why is your output different?
	+ Custom Program – What Am I?
		- Students are given the assignment to write a program to meet the specs.
	+ Drawing a Cartesian Plane
		- Axis Lines
		- X and Y label
		- Draw a Grid – Advanced
		- Graph a function - Advanced

## Learning Review

* Give an example of AND logic.
* Give an example of OR logic.
* Give an example of NOT logic.
* What is a loop?
* Write an example of a “For” loop
* Write an example of a “While” loop
* What is the difference between .. and … in a for loop.
* What does “by” do in a CoffeeScript for loop?
* Find the mistake in the loop.

# Week Six – Advanced Programing in MakeArt (~1hr 30min – 2hrs) – Part 3

Introducing object oriented programming.

## Lesson Summary

* Introduction to Objects and Classes
	+ Thinking in Objects
	+ Class Structure
	+ Member Variables
	+ Methods
* Putting it all together, your first class
* Convert Advanced Graphing Program to an Object

## Learning Review

* What is a class?
* What is a method?
* What is an object?
* What is a constructor?
* How do you define member variables?
* How do you define a method?

# Week Seven – Linux Operating Systems, Taking Control of the Terminal (~1hr 30min)

Introduction to operating systems: Windows, Mac OS, and Linux. Free Software Foundation and Operating Systems (GNU). Exploring the power of the command line terminal. It’s easy and fun with terminal quest.

## Lesson Summary

* Operating Systems
	+ Microsoft (DOS / Windows)
	+ Apple Computers (MacOS)
	+ Unix (Before Linux)
	+ Linux
* Who uses Linux?
* Why is Linux Cool?
* Becoming a Linux Ninja
	+ Introduction to Terminal Quest
	+ Live Raspberry Pi – Terminal Session

Students should be able to complete all of terminal quest in the given time. Once they are finished they can free play in MakeArt, StoryMode, or MakeMinecraft.

## Learning Review

* How do you look around? (list contents of a directory)
* How do you move an object? (move a file)
* How do you change rooms? (change directories)
* How do you print text to the screen?
* How do you delete an object? (delete a file)
* How do you list the contents of an object?
* How do you edit a file in the shell?
* How do you make a directory?
* How do you read permissions of a file or directory?

# Week Eight – Robotics Programming with Blockly and Dash (~1hr 30min)

Using programming logic students have learned throughout the course. Students will use a drag and drop programming interface to build software for a robot. We will revisit the following concepts:

* Functions
* Parameters
* Conditions
* Looping
* Variables

## Learning Review

* How did our programming experience help us program the robots?
	+ Did your understanding or conditions, loops, and logic help and how?
* Students should be able to explain functions of basic navigation
	+ Forward
	+ Reverse
	+ Left and Right turning
	+ Combine these combinations to perform a task
		- Example move around but not bump into things using the front and rear sensors.

# Week Nine – Navigate the Robot Maze (~1hr 30min+)

Using programming logic students have learned throughout the course and their understanding of the Blockly interface. Students will be divided into three teams. Each team will work on a program to allow the robot to navigate it’s way through a maze and obstacles. The goal is to make it out on the other side and additional points will be given for clever designs using the sensors.

This final assignment should probably be done on a weekend to allow for additional time to complete.

* Have students track what works, what didn’t work, and most importantly why?

Good Luck!