

Activities in of Support of Computer Programming

Teacher: Mr. Jacob Johnson

Course: Computer Programming

Semester: 2

Dates: 3/17/20-3/30/20

Email: johnsonjl@bpcschools.org

Background: The following activities are to be completed as support material for Mr. Johnson's Computer Programming Course. We are nearing the last part of the 3rd Quarter and are about to begin the 4th Quarter, so these materials will largely reflect information and coursework that we have been discussing in class from that timeframe. While this work will not be submitted to Mr. Johnson, the activities are expected to be completed in order to help prepare for our eventual return to normal classwork once this event has cleared up enough to return to our regular schedules. You should attempt to do one activity a day, ideally. Should you have any questions or concerns, do not hesitate to contact me via email.

Activities:

Modules 1 & 2 Review: Look back over the contents and exercises found in these modules on Codecademy. If you see any mistakes you made originally, try to fix them now. If you see any coding practices that we now observe, try to include those now too.

Data Types - Strings - Part 1: Review "Strings" as a data type. What sorts of data do we normally keep in Strings? Think up an example or two. Look around your room and identify something you could describe using a String. What would this look like? What are the properties of Strings? If you wanted to add a String together with another String, how would you go about this? What do we call adding one String to another? If you're not sure, please look this up. I have provided notes both on Google Classroom and there's information on Codecademy as well as the internet at large.

Data Types - Integers - Part 2: Review "Integers" as a data type. What sorts of data do we normally keep in Integers? Think up an example or two. Look around your room and identify something you could describe using an Integer. What would this look like? How would it look different from any other data type? How would you know you had an Integer and not a Float, for example? What are the properties of Integers? If you wanted to add an Integer together with another Integer, how would you go about this? What do we call adding one Integer to another? (Note: this is not the same name we use when dealing with Strings! Think more about general

mathematical concepts for a hint). If you're not sure, please look this up. I have provided notes both on Google Classroom and there's information on Codecademy as well as the internet at large.

Modules 3 & 4 Review: Look back over the contents and exercises found in these modules on Codecademy. If you see any mistakes you made originally, try to fix them now. If you see any coding practices that we now observe, try to include those now too.

Data Types- Floats - Part 3: Review "Floats" as a data type. What sorts of data do we normally keep in Floats? Think up an example or two. Look around your room and identify something you could describe using an Integer. What would this look like? How would it look different from any other data type? How would you know you had a Float and not an Integer, for example? What are the properties of Floats? If you wanted to add a Float together with another Float, how would you go about this? What do we call adding one Float to another? (Note: this is not the same name we use when dealing with Strings! Think more about general mathematical concepts for a hint). If you're not sure, please look this up. I have provided notes both on Google Classroom and there's information on Codecademy as well as the internet at large.

Modules 5 & 6 Review: Look back over the contents and exercises found in these modules on Codecademy. If you see any mistakes you made originally, try to fix them now. If you see any coding practices that we now observe, try to include those now too.

Data Types - Lists - Part 4: Review "Lists" as a data type. What sorts of data do we normally keep in Lists? Think up an example of a list or two containing at least 5 objects or pieces of information each. Look around your room and identify items you could add to a list. What would this look like? How would it look different from any other data type? How would you know you had a List and not a Dictionary, for example? What are the properties of Lists? If you wanted to add an list together with another List, how would you go about this? What do we call adding one List to another? If you're not sure, please look this up. I have provided notes both on Google Classroom and there's information on Codecademy as well as the internet at large.

Data Types - Dictionaries - Part 5: Review "Dictionaries" as a data type. What sorts of data do we normally keep in Dictionaries? Think up an example of a Dictionary or two containing at least 5 objects or pieces of information each that could serve as a key and at least 5 corresponding values. Look around your room and identify items you could add to a Dictionary. What values would you associate with each of their keys? What data types would you use for those values? Integers? Strings? When would each be appropriate? What would this look like? How would it look different from any other data type? How would you know you had a Dictionary and not a List, for example? What are the properties of Dictionaries? If you wanted to add a Dictionary together with another Dictionary, how would you go about this? What do we call adding one

Dictionary to another? If you're not sure, please look this up. I have provided notes both on Google Classroom and there's information on Codecademy as well as the internet at large.

Modules 7 & 8 Review: Look back over the contents and exercises found in these modules on Codecademy. If you see any mistakes you made originally, try to fix them now. If you see any coding practices that we now observe, try to include those now too.

Variable Review - Describe Objects as Variables: What are the general naming conventions behind each variable in the Python Programming Language? Pick ten different objects in the room you are currently in and give them names as variables that would actually make sense if you were going to be referencing them in a program. Try to pick a pattern and stick with it. Remember to not use any of the words that are Reserved, that Python will not allow you to use. Can you think of any terms that are reserved? Don't overthink it. I'm sure you know a few. Think about what we normally write in every program. Certain terms that show up cannot be used as variable names.

Modules 9 Review: Look back over the contents and exercises found in this module on Codecademy. If you see any mistakes you made originally, try to fix them now. If you see any coding practices that we now observe, try to include those now too.

Open Source Python Projects: Look up and research at least two different Open Source Python projects that you can find on the internet. SourceForge.net or SF.Net is a particularly good resource towards this end, but there are many others. You may also want to explore Github. What do these projects do? What does their mission statement or main page say about why they went with Python as their main programming language? Do they describe why they have provided their source code freely to the public? If so, why?

Complete The Middle section of Module 10: We had been working through Module 10 in class together during our last week of in-person classes. We would have finished the last 5 or so sections of Module 19 together on a returning week. Please try to finish the middle few parts of this Module on your own. Feel free to email me if you get stuck at all. Just try to be as detailed as possible in where you last left off. Screenshots would really be the ideal way to deal with this. If you need help taking a screenshot, let me know what kind of browser, device, and operating system you're working with, and I will do my best to get back to you.

Complete The last 3 parts of Module 10: Module 10 got particularly dicy toward the end, so we have chunked it up quite a bit. We had been working through Module 10 in class together during our last week of in-person classes. We would have finished the last 5 or so sections of Module 19 together on a returning week. Please try to finish this last bit of the Module on your own. Feel free to email me if you get stuck at all. Just try to be as detailed as possible in where you last left off. Screenshots would really be the ideal way to deal with this. If you need help

taking a screenshot, let me know what kind of browser, device, and operating system you're working with, and I will do my best to get back to you.

Reviewing List Slices: Look inside of your refrigerator and identify ten different items that are inside. Create a List of these 10 items. Format this list as if it was a Python data-type called "fridge_list." Create a new, empty list called "fridge2." If we are going to make the contents of fridge2 represent a slice of fridge_list, create a new list using only the items described by the following slice: fridge_list[1,4]. What items were not included in this new list? Which item did you inevitably have to skip over? Why did you have to skip over at least one item in this list? Look carefully at what has been asked of you for the answer. Email me if you want some help figuring it out.