Lesson Plan: Abstractions

Objective:

- Students will define and demonstrate what is an abstraction is and how to use one through coding
- Students will demonstrate how to use variables as abstractions
- Start the Create Performance Task for the AP

Standard:

- 9-12.CT.10 Collaboratively design and develop a program or computational artifact for a specific audience and create documentation outlining implementation features to inform collaborators and users.

Materials:

- Slides

Time Breakdown:

1. Do Now (3 min)
2. Abstraction Lecture/Slides(7 min)
    a. Examples
    b. Variables as Abstractions
    c. Dynamic vs. Static projects
3. Taco Truck Project and how it is an abstraction (5 min)
4. Purpose vs. Functionality (3 min)
    a. Define
    b. Example Vending Machine
5. Start the Create Performance Task using already made Taco Truck Project (Rest of class)

Lesson Plan: Lists

Objective:

- Students will understand what a list is in computer science
- Students will use methods to print, alter and find the length of a list
- Students will create and use a list through code

Standard:

- 9-12.CT.7 Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.

Materials:

- Slides

Time Breakdown:
Student/Group Paced Activities

6. Do Now (3 min)
7. Abstraction Lecture/Slides(7 min)
    a. Examples
    b. Variables as Abstractions
    c. Dynamic vs. Static projects
8. Taco Truck Project and how it is an abstraction
9. Purpose vs. Functionality
    a. Define
    b. Example Vending Machine
10. Start the Create Performance Task using already made Taco Truck Project

Lesson Plan: List Practice Programming Day 1

Objective:
- Students will practice creating and altering lists through coding challenges

Standard:
- 9-12.CT.7 Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.

Materials:
- iPad/Chromebook

Time Breakdown:
Students will work in pairs and the teacher will circulate to answer any questions.

Hatch - List Practice

1. **Create a List:**
    - **Named color with 6 colors in it**
2. **Add to a List:**
    - Ask the user to enter another color name
    - Ask the user to enter an index to insert the color at
    - Insert the new color at that index
3. **Remove from a List:**
    - Ask the user to enter an index number to remove from the list
    - Remove that color from the list.
4. **Accessing Value in a list:**
    - Have the sprite say the first color in the list.
    - Have the sprite say the last color in the list.
    - Have the sprite day the 4th color in the list.

Hatch - Wish List

1. Add 2 items to your list
        i. This can be anything you would like to buy such as food, games, etc.
    - Ask the user what item they would like to add to the wish list. Add this to the list.
        i. Repeat 3 times
    - Tell the user that the first item in your list is out of stock.
    - Ask the user what item they would like to replace the first item with.
    - Then replace the first item with the item the user input.
    - Have your sprite tell the user how long the list it.
        i. Use a complete sentence.
        ii. Hint: Use the join block and the length of block.

Hatch - Number List (Students edit already made code)

1. You will only need to change the set blocks here to hold the proper values.

2. Find the difference between the 3rd and 4th elements/item of numberList and assign it to the variable named difference

3. Find the quotient of the last and 1st elements of numberList and assign it to the variable named quotient

4. Find the product of the 2nd and 3rd elements of numberList and assign it to the variable named

product

5. Perform MOD on the 2nd and 4th element and assign it to the variable named mod

Lesson Plan: Algorithms

Objective:
- Students will define and demonstrate what an algorithm is
- Students will create their own algorithms
- Students will understand how algorithms are used in everyday life

Standard:
- Demonstrate how at least two classic algorithms work and analyze the trade-offs related to two or more algorithms for completing the same task.

Materials:
- Slides
- Paper
- Writing Utensils

Time Breakdown:
Student/Group Paced Activities
- Do Now (3 min)
- Introduce Lists (7 min)
    - What methods you can use on them
    - What they look like in code
- Practice AP Questions (5 min)
- List Activity (5 min)
    - Have student construct and alter a list as a class using methods from the slides
- Coding Challenge rest of class to practice

Lesson Plan: Day 2 of programming

Objective:
- Student will be able to:
    - Practice and show their skills in creating list
    - Enhance their knowledge of list while using the different procedures that are encased within Hatch

Standard:
- 9-12.CT.4 Implement a program using a combination of student-defined and third-party functions to organize the computation.
- Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.

Materials:

- Chrome books or ipad
- Mineolahs.oyoclass.com
- Slide deck

Time Breakdown:

- Do Now (3 min)
  - As students come in, there is an entry ticket recapping what different functions/procedures list already have
- Coding (35min)
  - Students are working on completing different list coding assignments
  - They are able to use past slides
  - We use the 3 before me process, where students have to ask 3 other classmates prior to asking me. In addition they need to recap and apply all the suggestions.
- Closure(5)
  - Bring the class together to discuss the challenges they faced and how they were able to overcome it.

Lesson Plan: List Reference Sheet

Objective:

- Students will be able to:
  - Explain the reference sheet
  - Compare and contrast between Hatch & Pseudocode
  - Apply their comprehension through practice AP questions.

Standard:

- 9-12.CT.9 Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.
- 9-12.DL.2 Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.

Materials:

- Slides
- List Reference Worksheet
- Writing Utensils

Time Breakdown:
Student/Group Paced Activities

- Introduction/instructions (3minutes)
- Main activity (30min)
  - This activity is a self paced activity where students sign into a peardeck where there are videos created to help explain each section of the AP reference sheet.

    - They then have to answer questions based on that section of the reference sheet
- Closure(5min)
    - Ask the students to reflect on the reference sheet. This allows them to make connections and see where they struggled. It gives a better gage of what we should focus on.

Lesson Plan: Review

Objective:
- Students will be able to:
    - Explain the reference sheet
    - Compare and contrast between Hatch & Pseudocode
    - Apply their comprehension through practice AP questions.

Standard:
- 9-12.CT.9 Systematically test and refine programs using a range of test cases, based on anticipating common errors and user behavior.
- 9-12.DL.2 Communicate and work collaboratively with others using digital tools to support individual learning and contribute to the learning of others.

Materials:
- Slides
- Quizziz
- gimkit
- Review Worksheet

Time Breakdown:
Student/Group Paced Activities
- We are reviewing for the multiple choice portion of the unit wrap up.
- We first review the worksheet, where we start with the ones the students found the most difficult
- We then proceed to move onto the vocabulary quizziz
- Then to gimkit for the concept and more practice multiple choice questions.
-

Lesson Plan: Practice CPT (2 days)

Objective:
- Students will be able to:
    - Develop a program using list and variables while including user input

- Explain their program and answer questions

Standard:

- 9-12.CT.7 Design or remix a program that utilizes a data structure to maintain changes to related pieces of data.
- 9-12.CT.10 Collaboratively design and develop a program or computational artifact for a specific audience and create documentation outlining implementation features to inform collaborators and users.

Materials:

- Google doc (record their responses)
- PowerPoint (references)
- Chrome book/ ipad (to code)

Time Breakdown:
Student/Group Paced Activities

- Introduction/instructions (5 minutes)
  - Go over the requirements for the create task.
  - Give them the choice between 4 programs to create:
    i. Random number generator
    ii. Fortune teller
    iii. Sports roster
    iv. Music playlist
- Create Task (37 minutes)
  - Students are to choose a program and make sure the requirements are met.
  - They are able to reference all prior materials and to ask their classmates for help.
  - They use the ask 3 before me rule prior to jumping in.
- Exit Tickets:
  - On the exit ticket we ask:
    i. What was challenging
    ii. What was a breeze
    iii. What they need more practice with.