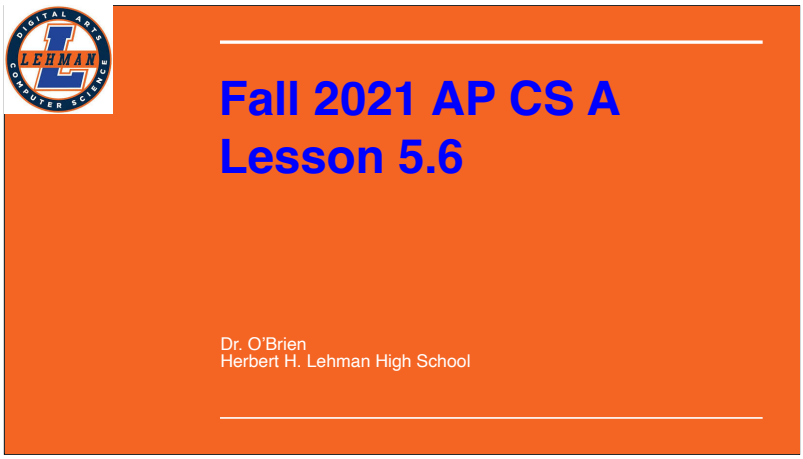
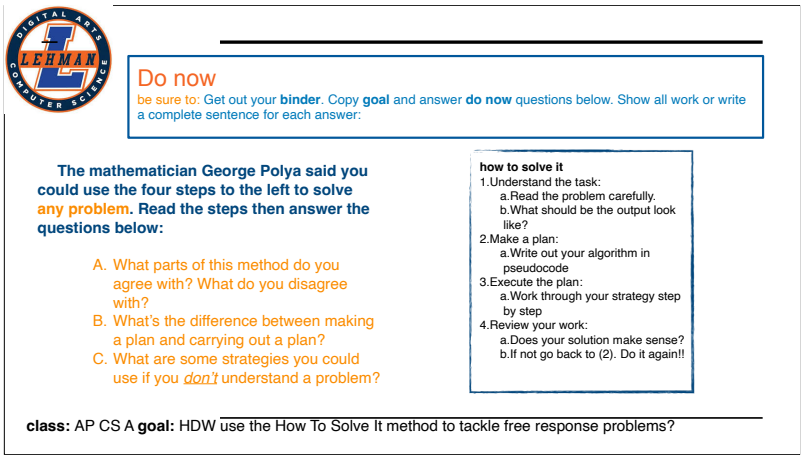


STANDARDS REFERENCED:

CSTA 11-12th grade standards: 3B-AP-12: Compare and contrast fundamental data structures and their uses.

NY State: 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.6
Demonstrate how at least two classic algorithms work and analyze the trade-offs related to two or more algorithms for completing the same task.



- A. Answers will vary. Students might complain that some problems are impossible to understand or impossible to solve.
- B. Making a plan is coming up with the series of steps you will need to perform to solve a problem. Carrying it out is actually doing it. This is like the difference between writing a recipe for cake and actually baking.
- C. Reread the problem, highlight key words. Look up any words you don't understand. Figure out whaty question is being asked

framing

- **what:** use the How To Solve It method to tackle free response problems
- **why:** This method helps organize how we solve free response problems
- **where to:** Comparing Java arrays and Python lists

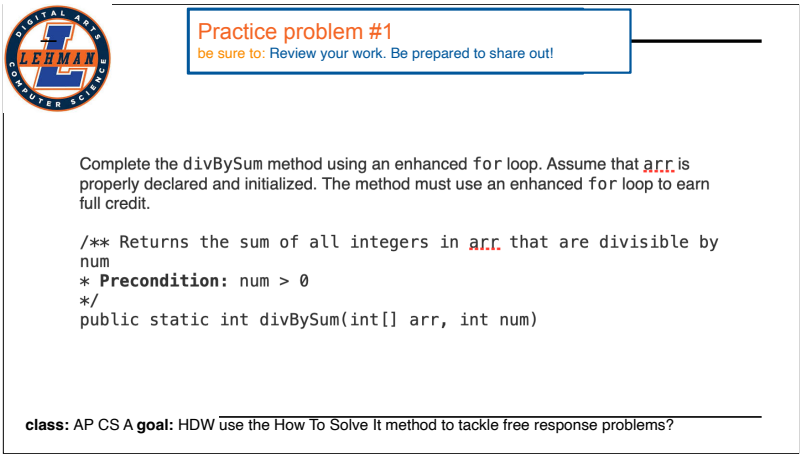
class: AP CS A **goal:** HDW use the How To Solve It method to tackle free response problems?

Activity: Practice with free response problems

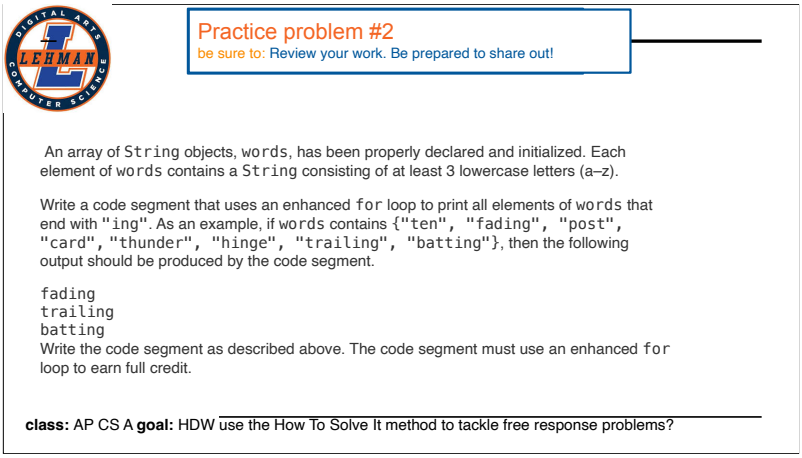
Work through each problem in the worksheet. For each problem…

1. Understand the problem: Read carefully. Describe the task your solving in one sentence.
2. Make a plan: Write out a code in pseudocode (again in your notes).
3. Execute your plan: Translate your pseudocode into Java. Write it in your worksheet.
4. Review your work: Does your code make sense? How do you know?

class: AP CS A **goal:** HDW use the How To Solve It method to tackle free response problems?




```
public static int divBySum(int[] arr, int num)
{
  int counter = 0;
  for (int item : arr){
    if ( item % num == 0){
        counter += item;
    }
  }
 return counter
}
```




```
for (String word : words){
   int length = word.length();
   String ending = word.substring(length - 3, length);
   if (ending.equals(“ing”){
     System.out.println(word)
   }
}
```

Practice problem #3a

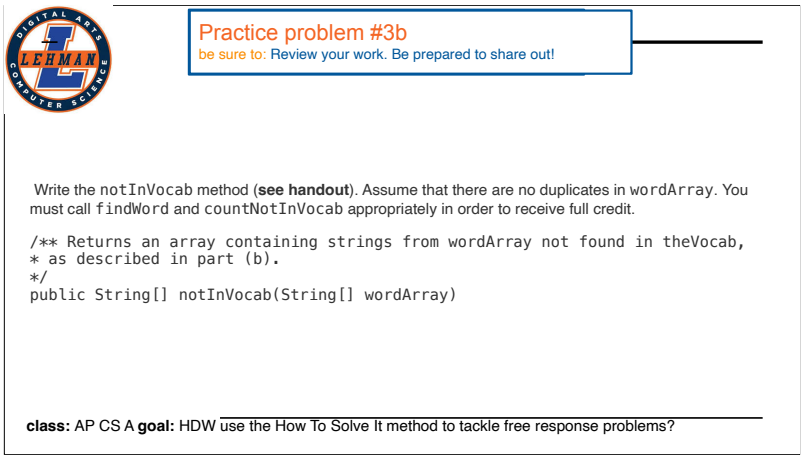
be sure to: Review your work. Be prepared to share out!

(a) Write the `countNotInVocab` method. Assume that there are no duplicates in `wordArray`. You must use `findWord` appropriately to receive full credit.

```
/** Counts how many strings in wordArray are not found in theVocab, as described in
* part (a).
*/
public int countNotInVocab(String[] wordArray)
```

class: AP CS A **goal:** HDW use the How To Solve It method to tackle free response problems?

```
public static int divBySum(int[] arr, int num)
{
  int counter = 0;
  for (int word : wordArray){
    if ( !(word.findWord()){
        counter ++;
    }
  }
 return counter
}
```



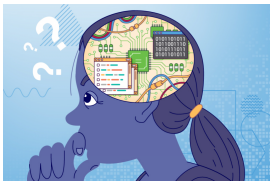

```
public String[] notInVocab(String[] wordArray) {
  int count = CountNotInVocab(wordArray);
  String[] newArray = new String[count];
  counter = 0;
  while (counter < count){


}
```

Reflection: Thinking about thinking

be sure to: Answer each question below with a complete sentence.

1. How useful was the how to solve it method for solving these problems?
2. What would you do differently?



class: AP CS A **goal:** HDW use the How To Solve It method to tackle free response problems?