# Project 1 Part B: AlphabetPanel class

The goal of this project is to become more familiar with creating and using classes and objects in Java, to further introduce you to the concepts of abstraction and modularization, and to introduce you to collections of items. In this project you will build on your Part A. You will add a Text class that will enable you to show text to the screen and create an AlphabetPanel class that uses the Text class.

# Instructions

* Create an AlphabetPanel class that extends JPanel which uses the Text class posted on the course web site. **DO NOT MODIFY THE TEXT CLASS IN ANY WAY.**
* Add an AlphabetPanel class and implement the following:
  1. Shows all of the letters of the alphabet (using a Text object for each letter)
  2. reset – a method that resets the alphabet panel letters all to black
  3. setLetterColor – a method that sets the color of a specified letter to a specified color
  4. getLetterColor – a method that gets the color of a specified letter. If the letter is not valid (i.e., not in the panel), return null.
  5. ERROR HANDLING: make sure the previous two methods work the same for upper and lower case letters; and that it correctly handles (ignores) characters that are not ‘A’ through ‘Z’
  6. hasLetterBeenSeen – a method that returns true if the letter has been pressed (i.e., changed color) since the last reset
  7. isVowel – returns whether or not the character is a vowel
  8. Add a main method that creates an AlphabetPanel object and puts it in a JFrame, it should react to typed input in the following manner: when a consonant is pressed that letter should turn red, a vowel should change to green, the spacebar should reset or clear the alphabet panel (all other keys should do nothing).

# Hints / tips

* Internally, characters are stored as numbers, specifically, as integers. This means we can work with AlphabetPanel letters as if they were numbers. For example, ‘A’ + 1 = ‘B’. Similarly, to determine that ‘Y’ is the 25th letter of the alphabet, we could calculate ‘Y’-‘A’+1 = 25.
* You can add Text objects directly to your AlphabetPanel, without explicitly creating an ArrayList by calling this.add. You can use JPanel methods like this.getComponent or this.getComponents to retrieve the text objects later. Google the JPanel javadoc for more information about these methods.
* In order to support key events in the AlphabetPanel (or later in Project 6 in the HangmanGame class) you will need to add the following code (which you need to complete) in the **constructor** of your AlphabetPanel class:

**this**.setFocusable(**true**); // enables panel to listen to key events

**this**.addKeyListener(**new** KeyAdapter(){

**public** **void** keyTyped(KeyEvent e)

{

// logic to handle key events

});

* Google the javadoc for KeyEvent for more information about how to access keys the user types.