AP Computer Science A: Unit 2, Program 5: Twitter

**Background:** Since its launch in 2006, Twitter has grown tremendously as a news and social media platform used by more than 319 million active users. As determined by its Alexa rating, Twitter is among the top websites visited both globally and within the United States.

Twitter works on a simple principle: users send out to one or more of their followers an intentionally-short message called a `Tweet.` The beauty of sending such messages lies in their brevity, making them much more easily consumable to the average user than other forms of news and social media. A `Tweet` historically has consisted of a `String` of a maximum of 140 characters (including URLs and uploaded multimedia) to accommodate an older protocol in text messaging. All Twitter users have a "handle", which is a pseudonym for the user that always begins with the "`@`" character. Messages may include recipients (another user's handle followed by a space) and may also be "tagged" for searchability among all Twitter. Tagging occurs via a "hashtag", which is a series of characters always beginning with the "`#`" character (and followed by a space).

In November of 2017, Twitter announced it was doubling the character limit of all messages to 280 characters, but for the purposes of this project we will stay with 140 characters maximum. All messages in Twitter are made up of characters in Unicode, allowing for people around the world to send messages to one another in a multitude of languages (which is why Java also uses Unicode).

**Directions:** You are to write a class that represents a `Tweet`, and build into that class the following characteristics:

- **Instance variables:**
  - An instance variable consisting of a `String` of characters to be the body of a `Tweet` to be sent. (Note that the body might contain recipient handles and/or hashtags in addition to the actual message – although the total must still be less than 140 characters)
  - An instance variable consisting of a `String` of characters representing a sender's "handle".
- **Constructor:**
  - A single parameterized constructor that receives two `Strings` of characters:
    - The first `String` consists of the body of a message a user wishes to send.
      - If this `String` exceeds the character limit, only the first 140 characters of the parameter become part of the actual Tweet. The remaining characters of the message are automatically cut off.
      - Otherwise, this `String` becomes the body of the message.
      - Note: The user's "handle" is **not** part of the 140-character limit.
    - The second `String` is a sequence of characters passed in as a parameter. It should be ***concatenated*** to an "`@`" character within the constructor. The resulting `String` should be stored as the sender's "handle".

- **Accessors/Getters:**
  - Two accessors, described as follows:
    - An accessor that returns just the body of a `Tweet`.
      - Signature for this method should read as:

```
public String getBody()
```

    - A second accessor that returns just the user's "handle".
      - Signature for this method should read as:

```
public String getHandle()
```

- **Two additional methods, described as follows:**
    - A `getAllRecipients` method that returns the “handle” of every directed recipient of the message, each separated by a new line:
        - Signature for this method should read as:

          ```
          public String getAllRecipients()
          ```
        - The format for what the returned `String` should look like follows:

          ```
          @PersonOne
          @PersonTwo
          @CorporationThree
          @CorporationFour
          ```
    - A `getAllHashtags` method that returns the “hashtags” found within a message, each separated by a new line, in the following format.
        - Signature for this method should read as:

          ```
          public String getAllHashtags()
          ```
        - The format for what the returned `String` should look like follows:

          ```
          #GOCUBSGO
          #CSForAll
          #140characters
          #h@$ht@g
          ```
- **`toString()`**
    - Finally, a `toString()` method that returns a `String` representation of a `Tweet` in the following format (note that the body of this particular message example contains EXACTLY 140 characters, NOT counting the sender’s handle):
        - `@UnnamedCSTeacherOne: @UnnamedCSTeacherTwo Star Wars is WAY better than Star Trek @UnnamedCSTeacherThree @UnnamedCSTeacherFour #PLC #SEL #noideawhatimtalkingabout`
    - Given the above example for what is returned in calling `toString()`, a call to `getAllHashtags()` would return the following:

        -
          ```
          #PLC
          #SEL
          #noideawhatimtalkingabout
          ```
    - Also given the above example, a call to `getAllRecipients()` would return the following:

        -
          ```
          @UnnamedCSTeacherTwo
          @UnnamedCSTeacherThree
          @UnnamedCSTeacherFour
          ```

- **Use the PROVIDED tester to identify whether your class works as intended.**