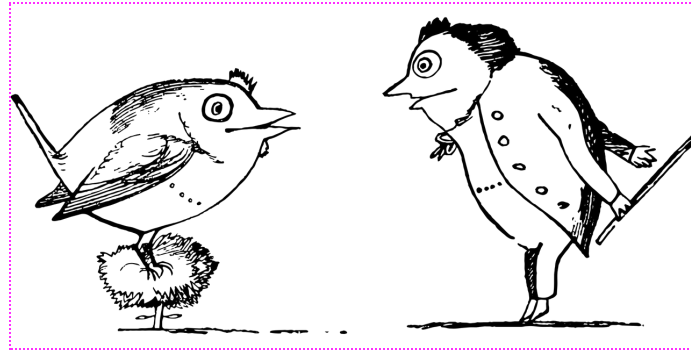


OPTION D
CLASS DESIGN and PROGRAM DEVELOPMENT



TwitterBirds are nearly extinct! They perform a ritual before building a nest together. It goes like this:

At the start of the ritual, there is a distance of 5 paces between each bird. They both tweet a random word. If both words are the same, the distance decreases by 1 pace. If the words are different, the distance increases by 1 pace. Both birds go nest-building if the distance becomes 0. If the distance becomes 10 or more, the TwitterBirds accept that they are not meant to be together and fly away to find another potential mate.

Today we will code 2 classes:

`TwitterBird.java`

to encapsulate TwitterBird properties and behaviour in one class/object

`Main.java`

to instantiate 2 TwitterBirds and code the ritual!

You are free to design any STATIC methods which can be called from your main program. For example, you could design a static method:

```
public static void displayArray(String[ ] x){//your code}
```

which would loop through x and display all Strings in it.

Then, in your main program you could do this:

```
TwitterBird t1 = new TwitterBird();  
String[ ] tweetsT1 = t1.getTweets();  
displayArray(tweetsT1);
```

This may make more sense after you have completed the exercise...

```

class TwitterBird
    //declare a property called rejections. This is a whole number.
    //declare a property called tweets that is a String array.
    //Here is the constructor method. As you are typing, hover over
    any red lines and import any necessary library files.
    Download the text file and save it where you can find it!

    public TwitterBird(){
        //assign a random number to the rejections property
        rejections = (int)(Math.random()*6);
        //initialise tweets to a length of 20
        tweets = new String[20];

        //import a text file into the project
        File file = new File("/Users/yourusername/Documents/tweetList.txt");

        //use a Scanner to read the file
        Scanner scan = new Scanner(file);

        //use a loop to read 20 lines into the tweets array

        int i = 0; //loop control variable

        while(scan.hasNext() && i<20) {
            tweets[i]= scan.nextLine();
            i++;
        }

        scan.close(); //close the Scanner
    }

```

Review of reading files in Java

//create a public accessor method, *getTweets()*, which **returns a reference to the tweets property**.

//create a public accessor method, *getTweet(int index)*, which **returns the String at the parameter index**. Make sure the code checks that the parameter is in the range 0 to *tweets.length-1*. If *index* is invalid, the method returns **null**.

//create a public accessor method, *getRejections()*, which returns the **rejections** property

//create a public mutator method, *setRejections(int num)*, which allows the **rejections** property to be updated.

```

} //end of class

```

Now let's code the ritual in our **Main.java** file! Here is a mix of pseudocode and comments - convert to Java!

```
public static void main(String args[]){

    //declare a variable called distance and initialise it to 5
    distance = 5

    //instantiate the first TwitterBird, t1
    //instantiate the second TwitterBird, t2

    //allow them to greet each other and share how many times they
    have been rejected!
    output "I have been rejected ", t1.getRejections(), "times!"
    output "I have been rejected ", t2.getRejections(), "times!"

    //start the ritual loop.
    loop while distance >0 and distance < 10
        //get a random tweet from one TwitterBird
        rand1 = random(0,19)
        tweet1 = t1.getTweet(rand1)
        //get a random tweet from the other TwitterBird
        rand2 = random(0,19)
        tweet2 = t2.getTweet(rand2)

        //if the tweets are the same, update distance
        if tweet1 = tweet2 then
            distance = distance - 1
        else
            distance = distance + 1
        end if

        output "Tweet1:", tweet1, ";Tweet2:", tweet2, ";Distance:", distance

    end loop //end of ritual

    //write code to determine why the ritual ended? Did they build
    a nest together?!
```

Adapt

Take a look at your program. Can you see why TwitterBirds are nearly extinct? Can you adapt your code to save them whilst allowing the romance of the ritual to continue. We all need rejection sometimes!

Review and Reflect

Write down any thoughts, observations, difficulties, comments etc that you have on this assignment.