```
##########################################

Student Name = Jitong Ding

##########################################



CSE017 Grading sheet for Jitong Ding

Homework Assignment TextDecoder

Total points                          maximum: 100

Completeness: All class/methods included (40)   [  40  ]

Compilation: Program compiles (20)               [  20  ]

Execution: Program executes properly (30)        [  28  ]

Style: Program obeys style rules (10)            [   9  ]

Subtotal                                            97
                                                 -------

Late Penalty                                     [      ]

Total Points                                        97
                                                 -------


##########################################
##########################################
```

```
 /*
CSE 17
Jitong Ding
jid221
Homework #3 DEADLINE: October 1, 2018
Program: Text Messaging
*/

import java.io.File;
import java.io.BufferedReader;
import java.io.FileReader;


/** A class that can read a series of text messages
  * from a file and then print out the decoded versions of them. */
public class TextDecoder{

  /** Date field*/
  /** A private data field named messages for the TextMessage[] of the TextDecoder and If there are fewer
messages than the size of the array, nulls appear at the end of the array.. */
  private TextMessage[] messages;
  /** A private data field named msgCount for the number of messages of the TextDecoder. */
  private int msgCount;

  /** Construct a new TextDecoder messages[10] and msgCount */
  public TextDecoder(){
    messages = new TextMessage[10];
    msgCount = 0;
  }
  /** A method for Reading messages from the file specified by the
msgFile object, updating messages and msgCount appropriately */
  public void readMessagesFromFile(File msgFile) throws Exception{
    /** Using BufferedReader and FileReader to read the message from the File*/
    BufferedReader message = new BufferedReader(new FileReader(msgFile));
    String text;
    String[] textMessage = new String[messages.length];
    /** Using while loop to copy the messages from the file to the textMessage array*/
    while((text = message.readLine()) != null){
      textMessage[msgCount] = text;
      msgCount +=1;
      if(msgCount==9){
        break;
      }
    }
    /** For loop to split the String into two parts */
    for (int i = 0; i< msgCount; ++i){
      String[] parts  = textMessage[i].split(" ",2);
      messages[i] = new TextMessage(parts[0],parts[1]);
    }
    if (msgCount<messages.length){
      for (int m = msgCount; m<messages.length; ++m){
        messages[m] = new TextMessage(null,null);
      }
    }
    message.close();
  }

  /** A method for printing all of the messages */
  public void printMessages(){
    for(int i =0; i<msgCount;++i){
      System.out.printf("%s %s\n",messages[i].getRecipient()+":",messages[i].getDecodedMessage());
    }
  }

  /** Checks if there is one command-line argument and exits with a helpful error message if
    * not. Otherwise, reads a set of messages from the specified file and then prints out
    * the decoded messages in the form above */
```



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

      if(args.length != 1){
        System.out.println("Usage: java TextDecoder filename");
        System.exit(0);
      }

      File messageFile = new File(args[0]);
      TextDecoder decoder = new TextDecoder();
      decoder.readMessagesFromFile(messageFile);
      decoder.printMessages();
    }
}
```

```
/*
CSE 17
Jitong Ding
jid221
Homework #3 DEADLINE: October 1, 2018
Program: Text Messaging
*/

/** A class for a single text message */
public class TextMessage{

  /** Date field*/
  /** A private data field named recipient for the recipient of the TaxTable. */
  private String recipient;
  /** A private data field named keyPresses for the keyPresses of the TaxTable.
*/
  private String keyPresses;

  /** Construct a new TextMessage recipient and keyPresses */
  public TextMessage(String aRecipient, String aKeyPresses){
    recipient = aRecipient;
    keyPresses =aKeyPresses;
  }

  /** An instance method return the recipient */
  public String getRecipient(){
    return recipient;
  }

 /** An instance method return the DecodedMessage */
  public String getDecodedMessage() {
    String decodeMsg = "";
    /** A two-dimension array for the decoded letters */
    String[][] decodeArray = new String[][]{{" "},{"."},{"A","B","C"},{"D","E","
F"},{"G","H","I"},
      {"J","K","L"},{"M","N","O"},{"P","Q","R","S"},{"T","U","V"},{"W","X","Y
","Z"}};
    int i =0;
   /** A while loop to transfer the message from the File to the decoded message
s */
    loop: while(i<keyPresses.length()){
      int numCount = 0;
     /** A if-statement to find the char'2' in keyPresses */
      if(keyPresses.charAt(i)=='2'){
        for(int k =1; k<4; ++k){
         /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
          if(i==keyPresses.length()-1){
            decodeMsg += decodeArray[2][0];
            break loop;
          }
          else{
          /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
 */
            if(i+k>keyPresses.length()-1){
              decodeMsg += decodeArray[2][k-1];
              numCount += k;
              break loop;
            }
            else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
              decodeMsg += decodeArray[2][k-1];
              numCount +=k;
              break;
            }
          }
        }
      }
    /** else A if-statement to find the char'3' in keyPresses */
      else if(keyPresses.charAt(i)=='3'){
        for(int k =1; k<4; ++k){
          /** A if-statement to find the i = keyPresses.length-1 in keyPresses *
/
          if(i==keyPresses.length()-1){
            decodeMsg += decodeArray[3][0];
```

-1 did not use 'this'


```
            break loop;
          }
          else{
           /** A if-statement to find the i+k > keyPresses.length-1 in keyPresse
s */
            if(i+k>keyPresses.length()-1){
              decodeMsg += decodeArray[3][k-1];
              numCount += k;
              break loop;
            }
            else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
              decodeMsg += decodeArray[3][k-1];
              numCount +=k;
              break;
            }
          }
        }
      }
    /** A else if-statement to find the char'4' in keyPresses */
      else if(keyPresses.charAt(i)=='4'){
        for(int k =1; k<4; ++k){
         /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
          if(i==keyPresses.length()-1){
            decodeMsg += decodeArray[4][0];
            break loop;
          }
          else{
           /** A if-statement to find the i+k > keyPresses.length-1 in keyPresse
s */
            if(i+k>keyPresses.length()-1){
              decodeMsg += decodeArray[4][k-1];
              numCount += k;
              break loop;
            }
            else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
              decodeMsg += decodeArray[4][k-1];
              numCount +=k;
              break;
            }
          }
        }
      }
    /** A else if-statement to find the char'5' in keyPresses */
      else if(keyPresses.charAt(i)=='5'){
        /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
        for(int k =1; k<4; ++k){
          if(i==keyPresses.length()-1){
            decodeMsg += decodeArray[5][0];
            break loop;
          }
          else{
          /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
 */
            if(i+k>keyPresses.length()-1){
              decodeMsg += decodeArray[5][k-1];
              numCount += k;
              break loop;
            }
            else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
              decodeMsg += decodeArray[5][k-1];
              numCount +=k;
              break;
            }
          }
        }
      }
     /** A else if-statement to find the char'6' in keyPresses */
      else if(keyPresses.charAt(i)=='6'){
       /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
        for(int k =1; k<4; ++k){
          if(i==keyPresses.length()-1){
            decodeMsg += decodeArray[6][0];
            break loop;
```

```
            }
            else{
            /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
    */
              if(i+k>keyPresses.length()-1){
                decodeMsg += decodeArray[6][k-1];
                numCount += k;
                break loop;
              }
              else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
                decodeMsg += decodeArray[6][k-1];
                numCount +=k;
                break;
              }
            }
          }
        }
       /** A else if-statement to find the char'9' in keyPresses */
        else if(keyPresses.charAt(i)=='9'){
        /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
          for(int k =1; k<5; ++k){
            if(i==keyPresses.length()-1){
              decodeMsg += decodeArray[9][0];
              break loop;
            }
            else{
            /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
    */
              if(i+k>keyPresses.length()-1){
                decodeMsg += decodeArray[9][k-1];
                numCount += k;
                break loop;
              }
              else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
                decodeMsg += decodeArray[9][k-1];
                numCount += k;
                break;
              }
            }
          }
        }
      /** A else if-statement to find the char'7' in keyPresses */
        else if(keyPresses.charAt(i)=='7'){
        /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
          for(int k =1; k<5; ++k){
            if(i==keyPresses.length()-1){
              decodeMsg += decodeArray[7][0];
              break loop;
            }
            else{
            /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
    */
              if(i+k>keyPresses.length()-1){
                decodeMsg += decodeArray[7][k-1];
                numCount += k;
                break loop;
              }
              else if(keyPresses.charAt(i+k)!=keyPresses.charAt(i)){
                decodeMsg += decodeArray[7][k-1];
                numCount += k;
                break;
              }
            }
          }
        }
      /** A else if-statement to find the char'8' in keyPresses */
        else if(keyPresses.charAt(i)=='8'){
       /** A if-statement to find the i = keyPresses.length-1 in keyPresses */
          for(int k =1; k<4; ++k){
            if(i==keyPresses.length()-1){
              decodeMsg += decodeArray[8][0];
              break loop;
            }
```



```
            else{
            /** A if-statement to find the i+k > keyPresses.length-1 in keyPresses
    */
              if(i+k>keyPresses.length()-1){
                decodeMsg += decodeArray[8][k-1];
                numCount += k;
                break loop;
              }
              else if(keyPresses.charAt(i+k)!= keyPresses.charAt(i)){
                decodeMsg += decodeArray[8][k-1];
                numCount += k;
                break;
              }
            }
          }
        }
      /** A else if-statement to find the char'0' in keyPresses */
        else if (keyPresses.substring(i,i+1).equals("0")){
          decodeMsg += decodeArray[0][0];
          numCount +=1;
        }
      /** A else if-statement to find the char'1' in keyPresses */
        else if (keyPresses.substring(i,i+1).equals("1")){
          decodeMsg += decodeArray[1][0];
          numCount +=1;
        }
       /** A else statement*/
        else {
          numCount += 1;
        }

        i += numCount;

}

      return decodeMsg;
    }

}
```

```
##########################################
##########################################

############ Compiled Result ###############

Source Code Compilation:

##########################################

##########################################
############ Execution Result ##############
##########################################

##########################################
Test1 output - testOutput1.txt

6107586533:   HELP PLS
4848675309:   JENNY I GOT IT
6107584096:   WHO ARE YOUR BEST STUDENTS

##########################################
Test2 output - testOutput2.txt

6107584096:   . ABC
6107584096:   DEFGHI
6107584096:   JKLMNO
6107584096:   PQRSTUVWX
6107584096:   YZ
6107584096:   .ADGJMPTW

##########################################
Test3 output - testOutput3.txt

123:   . ABC
12345:   DEFGHI
444445:   JKLMNO
123456677890:   PQRSTUVWX
8192364132:   YZ
2135123:   .ADGJMPTW

##########################################
Test4 output - testOutput4.txt

6107584096:   ..ADJLGDAC
6107584096:   DADEFADAMPR
6107584096:   P.ABG. NM....
6107584096:   A.JL.AEFDE
6107584096:   . ABC
6107584096:   DEFGHI
6107584096:   JKLMNO
6107584096:   PQRSTUVWX
6107584096:   YZ
```

-1 poor formatting

-1 slightly wrong output


```
##########################################
Test5 output - testOutput5.txt

Usage: java TextDecoder filename
```