Voice Programming

Data Collection

Imagine you are a programmer with Repetative Stress injury

Typing with keyboard and mouse is difficult for you

How would you speak code if you have an intelligent voice programming IDE ?

Instructions

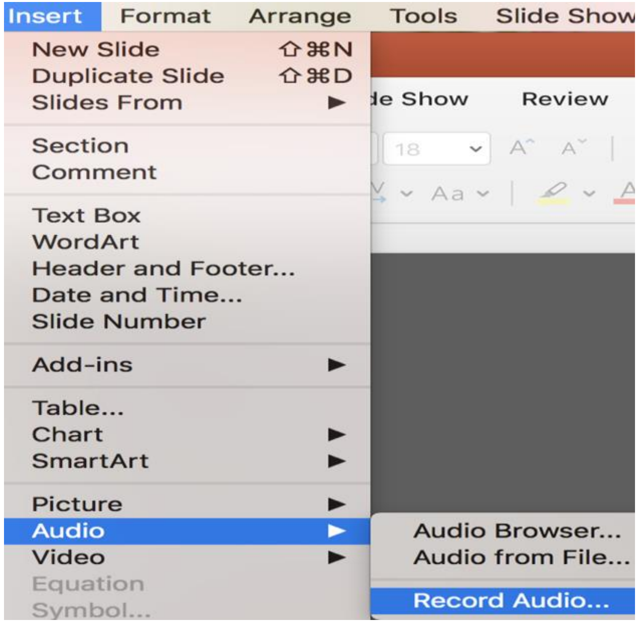
1. Do **not** open this presentation in SlideShow mode.
2. Fill out the **pre-experiment questionnaire** in slides 4 and 5.
3. After the questionnaire, each slide contains a complete Java program or a fragment of a program.
4. You can go through the programs **in any order** you want.
5. In each program, you have to speak either a missing line or a highlighted line and record it.
6. There is **no rule** on how to speak a line of code.
7. You do not need to type the missing line in the program.
8. After all the recordings, fill out the **post-experiment questionnaire** in slide 27.
9. **Save** the file and **upload** to https://keithv.com/codestudy/done

Recording a Sample Audio

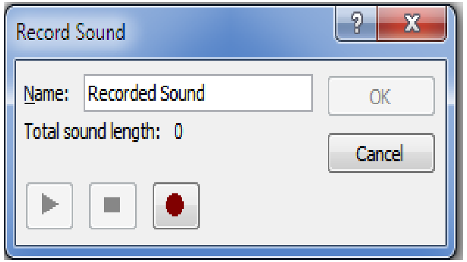
1. Go to **Insert** tab from the toolbar above.
2. Open the drop-down list on the **Audio** button.
3. Choose **Record Audio**.
4. A small pop-up box will be shown.
5. You do not need to change the **Name**.
6. Click the **red disc** to begin recording.
7. Click the **rectangle** to stop recording.
8. Click the **triangle** to playback the recording.
   a. You should hear your voice. If not, check the sound settings in your operating system and try again.

6. If you want to re-record, follow steps 4 and 5 above.
7. When finished, click **ok**.
8. A sound icon should appear on your slide.
   a. Its location and size isn't important.

1-3



4 -7



Name: Recorded Sound

Total sound length: 0

8

Practice Recording

- Practice recording your audio reading the line below:

The quick brown fox jumped over the lazy dog

- If you do not hear your voice, check the sound settings

Pre-experiment Questionnaire

Age: □ **Gender:** □ **Programming experience:** □ **years**

How much do you agree or disagree with the following statements (**Put number** from **1 to 7** in the box)?

Strongly Disagree	Disagree	Slightly Disagree	Neutral	Slightly Agree	Agree	Strongly Agree
1	2	3	4	5	6	7

1. I consider myself a fluent speaker of English: □

2. When I speak English, I have a foreign accent: □

3. When I speak English, people sometimes have trouble understanding me: □

Pre-experiment Questionnaire

How much do you agree or disagree with the following statements (**Put number** from **1 to 7** in the box)?

Strongly Disagree	Disagree	Slightly Disagree	Neutral	Slightly Agree	Agree	Strongly Agree
1	2	3	4	5	6	7

4. I frequently use speech interfaces to control a computer: □

5. Speech interfaces (e.g. Siri, Alexa) sometimes have trouble understanding me: □

6. I consider myself an expert programmer: □

7. I frequently write programs: □

Program 1

```
1. // This class represents an entry in a contact list
2. public class ContactEntry {
3.     private String name;
4.     private String email;
5.     public ContactEntry(String name, String email) {
6.         this.name = name;
7.         this.email = email;
8.     }
9.     public String getName() {
10.        return this.name;
11.    }
12.    public String getEmail() {
13.        return this.email;
14.    }
15.}
```

Dictate highlighted line 4

Program 2

```
1. // Display the numbers from an arraylist that are divisible by 3
2. public static void main(String[] args) {
3.    ArrayList<Integer> list = new ArrayList<Integer>();
4.    list.add(84);
5.    list.add(3);
6.    list.add(21);
7.    for (int i = 0; i < list.size(); i++) {
8.
9.          System.out.println(numbers.get(i) + " is divisible by 3");
10.      }
11.   }
12.}
```

Dictate missing line 8

Program 3

```
1. // Count the number of 'e's in a string
2. public static void main(String[] args) {
3.    String str = "elephant";
4.    int count = 0;
5.    for(int i=0;i<str.length;i++){
6.       if (str.charAt(i) == 'e') {
7.          count++;
8.       }
9.    }
10.   System.out.println(count);
11.}
```

Dictate highlighted line 5

Program 4

```
1.  // Employee contact list
2.  public class Employee {
3.      private int age;
4.      private double salary;
5.  
6.          this.age = age;
7.          this.salary = salary;
8.      }
9.      public String getSalary() {
10.         return salary;
11.     }
12. }
```

Dictate missing line 5 that represents Employee class’s constructor taking age and salary as parameters

Program 5

```
1.  //reads a file words.txt and displays the words as a list
2.  public class test {
3.   public static void read(ArrayList<String> wordList, File wordFile)throws FileNotFoundException{
4.    Scanner input=new Scanner(wordFile);
5.    while(input.hasNext()){
6.      String currWord=input.next();
7.      wordList.add(currWord);
8.     }
9.   }
10. public static void main(String[] args) throws FileNotFoundException {
11.     ArrayList<String> wordList = new ArrayList<String>();
12.     File wordFile=new File("words.txt");
13.     read(wordList, wordFile);
14.     System.out.println(wordList);
15.   }
16.}
```

Dictate highlighted line 13

Program 6

```
1.  // Create an object of the Triangle class and display the area
2.  // Sample Output: Area=20.0
3.  public class Triangle {
4.      double base, height;
5.      public Triangle(double b, double h) {
6.          base = b;
7.          height = h;
8.      }
9.      public void area() {
10.         System.out.println("Area="+0.5*base*height);
11.     }
12.     public static void main(String args[]) {
13.         double base = 5.0, height = 8.0;
14.
15.         obj.area();
16.     }
17. }
```

Dictate missing line 14

Program 7

```
1. // The Cow class that inherits from Animal class and implements method sound()
2. abstract class Animal {
3.     public abstract String sound();
4. }
5. class Cow extends Animal {
6.     @Override
7.     public String sound(){
8.         return "noooooo";
9.     }
10.}
```

Dictate highlighted line 7

Program 8

```
1. // Read elements in an Array and display
2. public static void main(String[] args) {
3.     int [] myArray = new int[5];
4.     Scanner scan = new Scanner(System.in);
5.     for (int i = 0; i < 5; i++) {
6.
7.     }
8.     System.out.println(Arrays.toString(myArray));
9. }
```

Dictate missing line 6 that reads integers from standard input into myArray

Program 9

```
1.  /* Displays the elements according to their type
2.   * Example output: Found Integer value: 22
3.   *                Not found integer or Double: Hello
4.   *                Found Double value: 2123324.58 */
5.  public static void main(String args[]) {
6.     Scanner scan = new Scanner("22 Hello 2123324.58");
7.     while (scan.hasNext()) {
8.        if (scan.hasNextInt())
9.           System.out.println("Found Integer value: " + scan.next());
10.       else if(scan.hasNextDouble())
11.          System.out.println("Found Double value: " + scan.next());
12.       else
13.          System.out.println("Not found integer or Double : " + scan.next());
14.    }
15. }
```

Dictate highlighted line 10

Program 10

```
1. // Iterating a string array that contains the strings "bird","fish" and "cow"
2. public static void main(String[] args) {
3.    String[] animals;
4.
5.    for (String item: animals) {
6.       System.out.println(item);
7.   }
8. }
```

Dictate missing line 4

Program 11

```
1. // This program creates an ArrayList list1 and adds three strings into the list
2. //Sample output: [pen, box, bottle]
3. public static void main(String args[]){
4.     ArrayList<String> list1=new ArrayList<String>();
5.     list1.add("pen");
6.     list1.add("box");
7.     list1.add("bottle");
8.     System.out.println(list1);
9. }
```

Dictate highlighted line 4

Program 12

```
1.  /* The Printer class prints a string and a float rounded to 3 decimal places
2.  *  Sample output: Phrase:Hello Number:12.500
3.  */
4.  public class Printer {
5.     private String phrase;
6.     private double number;
7.     public Printer(String phrase, double number) {
8.        this.phrase = phrase;
9.        this.number = number;
10.    }
11.    public void printPhrase() {
12.
13.    }
14.}
```

Dictate missing line 12

Program 13

```
1. /* A recursive Fibonacci method where nth fibonacci number is the sum of
2. * (n-1)th and (n-2)th fibonacci numbers
3. */
4. public static int fib(int n) {
5.     If (n <= 1) {
6.         return n;
7.     }
8.     return fib(n-1)+fib(n-2);
9. }
```

Dictate highlighted line 8

Program 14

```
1.  // The countVowel method takes the string "Hello" and return no. of vowels
2.  class Vowel{
3.     String myWord;
4.     int countVowel(String myWord) {
5.         this.myWord=myWord;
6.         int count =0;
7.         for(int i=0;i<myWord.length();i++){
8.             char ch = myWord.charAt(i);
9.             if(ch=='a' || ch=='e'||ch=='i' || ch=='o' || ch=='u')
10.                count++;
11.        }
12.        return count;
13.    }
14.    public static void main(String args[]){
15.        Vowel obj = new Vowel();
16.        int count;
17.
18.        System.out.println(count);
19.    }
20. }
```

Dictate missing line 17 where variable **count** receives no. of vowels returned from the method **countVowel.**

Program 15

```
1.  public class Entry {
2.     public String name;
3.     Public String age;
4.  /**
5.   * @param newAge A variable of type int
6.   */
7.     public String setAge(int newAge) {
8.        age = newAge;
9.     }
10. /**
11. * @param newName A variable of type String
12. */
13.    public void setName(String newName) {
14.       name = newName;
15.    }
16. }
```

Dictate the highlighted lines 10-12

Program 16

```
1.
2. public double getMin(double[] array) {
3.  double min = Double.MAX_VALUE;
4.   for(double num : array) {
5.    if (num < min) {
6.     min = num;
7.    }
8.   }
9. }
```

Dictate the missing comment in line 1 that describes the behavior of the method getMin

Program 17

```
1. // The Equation class calculates result as
```

$$result = \frac{\sqrt{y} + x}{z^2}$$

```
2. public class Equation {
3.    public static void main(String args[]) {
4.        double x=8.0, y=16.0, z=3.0;
5.        double result;
6.        result=(Math.sqrt(y)+x)/(Math.pow(z,2));
7.        System.out.println(result);
8.    }
9. }
```

Dictate highlighted line 6

Program 18

```
1. /* The class Complex initialize an integer variable largeNumCounter to 0 and
2. * increments it.
3. */
4. public class Complex {
5.     private double largeNum;
6.
7.     public double increment() {
8.         largeNumCounter++;
9.       }
10.  }
```

Dictate missing line 6

Program 19

```
1. /* The Test class creates a public constant variable CAPACITY that is static
2. * and final and initialized to 15
3. */
4. public class Test {
5.   public static final int CAPACITY=15;
6.   public Test() {
7.       System.out.println("Capacity = " + CAPACITY);
8.   }
9. }
```

Dictate highlighted line 5

Program 20

```
1.  // This program converts a word to its corresponding number
2.  public static void main(String[] args) {
3.     String str = "three";
4.     switch(str) {
5.        case "one":
6.           System.out.println("1");
7.           break;
8.
9.           System.out.println("2");
10.          break;
11.       case "three":
12.          System.out.println("3");
13.          break;
14.     }
15. }
```

Dictate missing line 8

Post-experiment Questionnaire

How much do you agree or disagree with the following statements (**Put number** from **1 to 7** in the box)?

Strongly Disagree	Disagree	Slightly Disagree	Neutral	Slightly Agree	Agree	Strongly Agree
1	2	3	4	5	6	7

1. I found speaking a missing line of code easy:
2. I found speaking a highlighted line of code easy:
3. I found the given programs were easy to understand:

Please answer the following by **writing** in the box:

4. What parts of code (e.g. variables, loop, method) were you most uncertain about what to speak?