Description of User Stories
Sprint 3

1. View Higher Levels
   a. We made it so that all lessons are viewable once clicked on from the homepage.
      i. Once the user clicks on a button for a particular lesson, they are presented with the lesson introduction and resource link for more information.
2. Locked Levels
   a. We made it so that all lesson can be viewed but questions cannot be answered until the requirements are met.
      i. User's view the lesson same as before by clicking the button for that lesson but will not be able to access the questions for it until the previous lesson is completed with 80% or higher accuracy.
      ii. The first lesson, Intro under Elementary Programming, is already unlocked and is the default starting point.
      iii. The next button to access lessons is already set as disabled by default but is enabled once requirements are met. All progress information is saved under sharedpreferences objects.
3. Short Tutorial
   a. We made it so that before a set of questions for a lesson there is a brief introductory paragraph explaining to users what the lesson will be about.
      i. Under the library class there is an array that stores all text for the lessons including the title, intro and body.
      ii. The lesson activity pulls from this array and calls the setText() method to set the appropriate text for title, into and body.
         1. nameOfLesson.setText(lessonSet[0]);
         2. intro.setText(lessonSet[1]);
         3. body.setText(lessonSet[2]);
4. App Tutorial
   a. We made it so that there is an app tutorial that is available for users to be able to see how to navigate the app and also see the various features of the app.
      i. Once navigated to the option tab there is a button for App tutorial.
      ii. Once clicked, the UI is changed so that the screenshots of the app can be viewed and scrolled through, each explaining portions of the app.
      iii. This was done using the methods:
         1. hideMainOptions()
            a. Hides the original display that contains the four button on the options page.
         2. showTutorialScreen()
            a. Shows the new displaying containing the images.
5. Pretest

   a. We implemented a pretest that shows after sign up to gauge the user’s current skill level with java programming.
      i. Pretest is automatically shown after user completes sign up
      ii. All information for pretest is in the library class
      iii. Pretest questions are stored in an array and choices are stored in another array. Both are accessed from PreTestActivity.java
      iv. At the end of the pretest a results screen is shown that shows the percentage correct from each section.
      v. This is calculated by taking number of correct selections and dividing by number of questions for the section(10) to get the percent.

6. Resource Link
   a. We put resource links at the bottom of every lesson that take the user to a youtube video to further explain the information in the upcoming section.
      i. There is an array under lesson activity that contains the links.
      ii. The links are set based on which section they’re under and the lesson number.
7. Feedback
   a. We put a feedback button in the options page that allows users to send feedback responses to the developers.
      i. Once feedback is clicked users see a screen that ask for a subject line (optional) and message. Once inputted they hit send feedback and can send an email to the developers through gmail.
      ii. Emails are sent using intents and the developers emails have been hard coded in.
8. Bug Reporting
   a. We put a bug reporting button in the options page that allows users to report bugs to the developers.
      i. Once bug reporting is clicked users see a screen that ask for a subject line (optional) and message. Once inputted they hit send bug report and can send an email to the developers through gmail.
      ii. Again, emails are sent using intents and the developers emails have been hard coded in.
9. Frequently Asked Questions
   a. There is a button in the options page titled “Frequently Asked Questions” that allows user to view common questions and their answers.
      i. Once a user clicks on the frequently asked questions button it will display the frequently asked questions.
      ii. A user will be able to read through the frequently asked questions.
         1. Questions are held in an array in the Options activity java file
         2. The answers are there also in a separate array.
         3. Both arrays are accessed through showFAQScreen(); method.
10. Completion Certificate

a. We put a completion certificate in the profile page that is unlocked once the user completes every lesson in the app.
   i. If the user clicks on the profile button it will check to see if every lesson has been completed with 80%or higher.
      1. This was done using an if statement inside the finishedAllLessons() method
   ii. If the user has completed every lesson with an 80% or above then the the image of certification will appear along with a TextView saying to click the image to save.
   iii. If the user clicks on the image it will ask if the user wants to save the image of certification using alert dialog builder.
      1. This was done using the AlertDialog.Builder class
      2. using a Builder.setPositiveButton() method if they click save
      3. Also using a Builder.setNegativeButton() method if they click don't save
   iv. If the user clicks save then it will save a png to the downloads directory
      1. This was done by setting the file path to Environment.DIRECTORY_DOWNLOADS
      2. Then creating a file with that path
      3. Then using an outputstream to compress the file and put it in the downloads
   v. The png will have each section and the users score listed in order
   vi. If the user clicks don't save then it will not save the image

11. Progress Snapshot
   a. Under progress in the profiles tab the user's current completion percentage is displayed for each lesson.
      i. When the user clicks on the progress button it will display each lessons score that was 80% or above in a TextView
      ii. This is done using the progressSnapshot() method

12. Skill Mastery
   a. There are medals for each section under profiles that display once a user has completed a section.
   b. Each section has its own medal.
   c. Medals are set as transparent by default but if users have completed all the lessons in a section fully with 80% or higher accuracy, the medal for it changes visibility so that it shows fully.
      i. To see if requirements are met medalProgress() is called in the profile activity class.
      ii. The medals transparency is affected by changing its alpha value

13. Perfectionist Medal
   a. This is also under the profiles tab and is awarded once users complete all lesson at 100% accuracy.

    b. This medal is also set to transparent by default and changes visibility if all lessons have been completed with 100% percent accuracy.
        i. Again to see if requirements are met medalProgress() is called in the profile activity class.

14. Erase Data
    a. Allows users to reset all progress and start again from the beginning.
    b. Once clicked, prompts a confirmation message.
    c. If confirmed all shared preferences are cleared.
        i. myPref.edit().clear().apply();