This guide provides content highlights relevant to each objective within the slide deck. This guide also has icons noting where there are knowledge checks and learner activities /exercises. *This guide is not a script*, so you will need to first review the slides taking note of the highlights. Any additional notes that the instructor needs or wants to add can be added in the instructor notes section within each block.

Icons within the presentation:

|  |  |
| --- | --- |
|  | Guided exercise - follow the actions of the instructor. Practice together as instructor demonstrates. “Show it- Try it”- demonstrate before asking students to perform the task |
|  | Exercise; take note of Code |
|  | Exercise in workbook/terminal |
|  | In class activity (could be group or independent practice) |
|  | Knowledge check |

**Chatbots day 1: quick overview**Lesson 1 Introduction and Python exploration: Students will discuss the use cases and concepts of the chatbot architecture

Lesson 2 Python exploration: Students will explore the functionality of Rasa chatbot stack components and review Pandas and scikit-learn key features

Lesson 3 Key components of a simple bot: Students will explore the architectural details of a simple bot

Lesson 4 Building and modifying a simple bot: Students will build and modify a starter-pack bot

Materials:

* Ground Rules
* Index Cards
* Adaptors
* All downloaded software
* Computer/Internet
* Large post-it paper, sticky notes & markers
* Laptop, Screen and Projector
* Instructor Guide
* Slack
* Google Forms and Drive

Day 1 - Objectives & Agenda at a Glance

|  |  |
| --- | --- |
| Time | Topics |
|  | **FYI- training team arrives hour before start to set up room.** |
| 15 min | **Introductions & Welcome**  Staff & team introduction   * Organization * Who I am * Goals for today  *“So if you’re hoping to learn basket weaving, or the history of modern dance, you are in the wrong place….”*   *“Before we dive into content, let’s get some introductions to this cohort we will be working with for the next three days:”*  Student introduction Pair and Share -   * Name * Department/Job responsibilities * Fun fact or something the team would be surprised to know |
|  | **Optional Addition:**   * Discussion ( you can prepare questions you want to ask) |
| 15 min | **Activation- Case Studies**  Divide students into groups ( there are 8 case studies)   * Review and discuss in small groups * Be ready to report domain of business, challenge, what will be improved, what they found interesting, etc. * Each group will report out |
| 1 -2 hrs | **Lesson 1: Introduction**  By the end of lesson 1, students will be able to:   * Summarize chatbot case studies in the domain of banking or health * Identify the terms of the chatbot creation life cycle * Find references and specifications of chatbot life cycle within chatbot case studies * Discuss the architecture and stack choices of a chatbot |
| 20 min | Break ( insert as needed throughout the day) |
| 1-2 hrs | **Lesson 2: Python exploration**  By the end of lesson 2, students will be able to:   * Summarize use of Python in Rasa AI stack * Summarize Pandas package for data wrangling * Summarize scikit-learn algorithms used in Machine Learning * Summarize use of scikit-learn and neural network libraries in Rasa Stack |
| 60 min | Lunch Break |
| 1-2 hrs | **Lesson 3: Exploring the architecture of a simple bot**  By the end of lesson 3, students will be able to:   * Navigate to starter-pack and explore folder structure * Explore elements of Rasa Core architecture * Explore elements of Rasa NLU architecture |
| 20 min | Break ( insert breaks throughout as needed) |
| 1-2 hrs | **Lesson 4: *Building a simple bot***  By the end of lesson 4, students will be able to:   * Install starter-pack requirements * Train and test starter-pack bot * Visualize stories as a graph with graphviz * Customize, retrain, test and visualize starter-pack bot |
| Debrief | Debrief- Revisit the class learning objectives. **Visit the parking lot. If some question answers need to be researched and presented the next day, be sure to find the answers and have them ready for the start of day 2.**  Preview – Start preparing students for the next class by giving a preview and connecting what they learned during the present class with material to come.  **Closing Activity: 3, 2, 1 Activity** (note card or google form)**-**   * + 3 most valuable things you learned   + 2 -things you wish you had more time to review/most difficult   + 1- concern/suggestion/question you still have   Closing activity can be put in slack/google forms (chatbots group) |

## General Facilitation Guidance

Demonstrations- There should always be a demonstration of a task prior to asking the participants to perform the task.  
  
Lesson Summaries- Always do a descriptive summary to recap content before moving to the next mini lesson and to close out current the mini lesson. If possible ask learners first to summarize ( great way to assess), then follow with your own summary. Summaries repeat the main points/takeaways and also provide opportunity to discuss the use cases or workplace applications throughout the training.

Explain the stop light index cards- before the training, prepare green and red index cards. Make a set of three index cards, one of each color, for each group member. These are used mainly after activities, installing items or after exercises.

**Red**- Not ready to continue, I need help, I have a question, I’m lost, **you are going too fast**.

**Green**- Finished, ready to continue, I got it!

Explain Parking Lot- There should be a pdf given at the beginning of class that has links for the KCs and parking lot. This is where students can post questions or comments at anytime OR when time is short and the instructor cannot pause for questions. Instructor will eventually need to stop at points in the lesson to check the parking lot and address the comments. For example, this can be done before a break or at the end when debriefing.

Exercises and Activities- Should be in a folder. Confirm in advance that you are able to access them all, and that code works properly

**Effective Questioning Job Aid**

What happened?  
 Why do you think it happened that way?  
 Is that what you expected?  
 What could have been done differently?  
 What would you do differently next time?

Can you see any themes emerging here?

What principles are at work here?

What connects all these experiences?

Where and when do you think you might use these ideas?

How do you think you could put them into effect?

What criteria would need to be in place for this idea to work?

What possible problems do you need to be aware of?

Where do you go from here?

**How to ask questions**

Here are some strategies for ensuring your questioning gets good results:

•Review your slides and plan your questions

•Choose the right words

•Draw on people's existing experience to help them find solutions to a problem

•Use open questions when you want to open up and explore issues

## Lesson 1 Introduction and Python exploration

Students will discuss the use cases and concepts of the chatbot architecture

|  |  |  |
| --- | --- | --- |
| Introduction and Case Studies  Slides: 1-7  Time: 30 min  Demonstrate    Use Cases | **Objective (s):**   * Summarize chatbot case studies in the domain of banking or health   **Instructor/Participant Introductions ( reference agenda above for how to start training )**  **Facilitation Guidance**   * Chatbot usage stat- The global chatbot market is expected to reach $1.23 billion by 2025 ( source: business insider) and approximately 45% of end users prefer chatbots as the primary mode of communication for customer service inquiries * Interactivity/Optional Discussion- Room check familiarity with chatbots. Ask open ended questions to spark conversation about experiences. Then show the definition of a chatbot * Inform learner of Lesson 1 objectives * State importance and benefits of training- Example, “You will be able to build and fine tune a chatbot, and you will be able to help your organization provide an improved customer experience which is very valuable. Save time and resources!”   **Activity -** Use cases- Jigsaw activity with up to 9 use cases. Divide students into groups. Discuss for 15 minutes in groups.  **TASK**   * + Students will receive weblinks to a case study  |  | | --- | | - Case A: [Cornell Bot](https://microsoft.github.io/techcasestudies/bot%20framework/2017/06/15/WeillCornell.html)  - Case B: [Erica Part 1](https://www.forbes.com/sites/quora/2016/10/28/meet-erica-bank-of-americas-new-voice-ai-banking-system/#28ab9b6d50db) [Erica Part 2](https://www.bizjournals.com/charlotte/news/2018/05/21/bank-of-america-rolls-out-ai-assistant-erica-to.html)  - Case C: [Tia part 1](https://rasa.com/solutions/healthcare/) [Tia part 2](https://www.asktia.com/app/)  - Case D: [Cora](https://www.ibm.com/industries/banking-financial-markets/front-office/chatbots-banking)  - Case E: [Raiffeisen](https://rasa.com/solutions/banking/)  - Case F: [Eno](https://www.forbes.com/sites/capitalone/2017/10/19/becoming-a-bot-ai-design-and-the-incomparable-eno/#681ccfffebc2)  - Case G: [Helvetia](https://rasa.com/customers/insurance/)  - Case H: [SmartCom](https://rasa.com/customers/telecom/)  - Case I: [Travel Bot](https://rasa.com/customers/travel-and-transport/) |  * + Review and discuss in small groups   + Be ready to report domain of business, challenge, what will be improved, what they found interesting, etc.   + Each group will report out   **Note:** To remain in the world of the practical, rather than abstract, we have nine chatbot case studies we will refer back to throughout the day. Also studies show adults learners retain knowledge more deeply when they learn collaboratively. This training is not just about you imparting knowledge to them, but developing a cohort that learns together.  **Note:** All exercises and activities should be explained/ demonstrated before beginning. It’s a good idea to ask 2-3 people to repeat back the instructions to make sure everyone understands the instructions.  Be sure to ask open ended questions that lead to more discussion versus yes or no answers. ***Questions that may lead to in depth conversation outside of the time allotment (or off topic) can be added to the parking lot. “Parked”***  **Instructor Notes:** |

|  |  |
| --- | --- |
| Chat Bot- Life Cycle  Demonstrate    Use Cases | **Objective (s):**   * Identify the terms of the chatbot creation life cycle * Find references and specifications of chatbot life cycle within chatbot case studies   **Facilitation Guidance**   * When developing a bot, there is a life cycle that you will go through * Chatbots are a product in continuous development, improvement and testing cycle * Requirements- you will need to research and gather market information before making your bot. For example, who is the target market/customer, what solutions do you want to provide and problems you want to solve * Specifications- identify features and functionality of the bot * Conversational script- you will need to build conversational scripts that are similar to a conversation a customer would have with a person ( NLP/AI ) * Architecture- engineering of the design will need to include front and back end components * Development- the bot is developed during this stage it includes a lot of coding and testing * Testing- you will need to see if the bot says the right thing when it is supposed to. The testing will need to be done in the actual platform as well * Deployment- deploy to the hosted environment * Development, Testing, Deploying- these step are intertwined and are usually repeated several times before moving to the next phase * Publishing- your bot will need to be approved (can be submitted to several apps stores) * Monitoring- the conversational scripts will need to be monitored * Promoting- having the bot in an app store is a great way to promote. You can also advertise * Analysis- the performance, metrics, and conversation will need to be analyzed   **Activity 2 -** Students should go back in the same groups to identify the components of the chatbot lifecycle within their pretend bot and report out to group.  **Notes-** The classroom needs to be organized in a way that makes it easy for students to collaborate and get in and out of groups when necessary.  ***Questions that may lead to in depth conversation outside of the time allotment can be added to the parking lot. “Parked”***  **Instructor Notes:** |

|  |  |
| --- | --- |
| Chatbot - Architecture  Demonstrate | **Objective(s):**   * Discuss the architecture and stack choices of a chatbot   **Facilitation Guidance**   * When engineering your chatbot, it will need to be able to determine what the person is asking and what action is appropriate * Within the technology stack there are different technologies you can use. Whichever stack you choose, it will need to have three components. User interface, NLU, and a core dialogue system * You can choose to create a bot custom or through a platform. Throughout the training we will use the RASA platform to create the bot. Participants will eventually make a bot from scratch * Within the RASA platform the components are RASA NLU and Core. Rasa NLU’s job is to interpret messages, and Rasa Core’s job is to decide what should happen next. Each component has a lot of documentation which can be found at <http://rasa.com/docs/getting-started/overview/> * Rasa NLU performs Natural Language Understanding, which means taking free-form text and turns it into structured data * The Core tracks the conversations and decides how to proceed ( Dialog management) * FYI- You do not have to so use RASA NLU with Core. They are independent * Why Conda and Pip? RASA was coded on the backend in Python. NOTE- other programming languages can be used. Python is a programming language, and Anaconda will provide the virtual environment. Anaconda also simplifies package management and deployment. Pip is a package management system used to install and manage software packages written in Python   Source: Rasa.com  **Notes-** It may be helpful to show the website and where the documentations is. The training will go more in depth about each component, so limit questions during this portion. For the knowledge check, be sure to review the answers and to address misconceptions and review why the answer choice is correct.  ***Questions that may lead to in depth conversation outside of the time allotment can be added to the parking lot. “Parked”***  **Instructor Notes:** |

## 

## Lesson 2: Python exploration Students will explore the functionality of Rasa chatbot stack components

|  |  |
| --- | --- |
| Python and RASA AI stack  scikit-learn | **Objective(s)**   * Summarize use of Python in Rasa AI stack   **Facilitation Guidance**   * For the slide that asks again why Python in the beginning. Reiterate why we are using the technology we are using ( because Rasa was built using Python) * Pandas, is an open source library that provides developers with convenient data structures analytic tools; is an important tool for Python. * Scikit-learn: Machine Learning in Python;Simple and efficient tools for data mining and data analysis * Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN   **Instructor Notes:** |

## 

## Lesson 3: Key components of a simple bot Students will explore the architectural details of a simple bot

|  |  |
| --- | --- |
| Simple bot architecture  Demonstrate    Exercise- Review, Match & Assess | **Objective(s)**   * Navigate to starter-pack and explore folder structure * Explore elements of Rasa Core architecture * Explore elements of Rasa NLU architecture   **Facilitation Guidance**   * 3 components, NLU, User Interface and CDS * These 3 components can be custom built or using a platform * The Rasa community ( which comes with Rasa stack) is a resource to interact and converse with other builders * Rasa Core and NLU have different components that need to be need to be in place, and should be a part of the Rasa Starter pack * The starter pack will be included in the class folder * NLU configuration file should have pipeline and language specified * Core components and their format- Domain, Stories, scripts that can contain custom actions * Rasa NLU components - training data in markdown or JSON, Configuration file * Optional files- google drive service, makefile, readme   Core vs NLU  As you can see, Rasa Core has more components in Rasa Stack in the overall bot architecture    **Exercise**- Review, Match & Assess  Within the chatbot directory you should see the following folders:   * moodbot * concertbot   Each of those folders is a separate bot.  Please split into **two** groups for this exercise and pick one of the bots.  Each group is assigned the following tasks:   * Examine the contents of the bot folders   + What is the contents of the folders?   + Which NLU pipeline components does the bot use?   + Which files are related to which purpose within the bot architecture? NLU or Core?   + Is there a README file with anything interesting? * Reporting out their findings to the rest of the group   **Note:** All students should have the Rasa.ai components already installed on their machine and all respective folders in the main class folder  **Technical Notes:**  **Instructor Notes:** |

## 

## Lesson 4: Building and modifying a simple bot

## Students will build and modify a starter-pack bot

|  |  |
| --- | --- |
| Building a simple bot  Demonstrate    Discussion    Exercise- train, test & visualize your bot | **Objective(s)**   * Install starter-pack requirements * Train and test starter-pack bot * Visualize stories as a graph with graphviz * Customize, retrain, test and visualize starter-pack bot   **Facilitation Guidance**   * The NLU model will be based on the sample data provided with the bot, it can be trained using **make train-nlu** command in the terminal, which is a shortcut for a longer command explained in the slides * The Core will be based on the sample data provided with the bot; use the command in the terminal **make train-core** * Within the **starter-pack** folder   + **dialogue**: where Rasa Core training data and models are saved   + **nlu**: where Rasa NLU models are saved * For testing you will run your sample bot in the terminal using **make cmdline** command, which again is a substitute for a longer command explained in slides * Once the bot is ready, during testing try entering simple responses * You can enter ctrl + c to exit out of the running chat process with your bot * **graphviz** allows you to visualize chatbot stories as a graph. It is software that needs to be installed outside of Rasa Stack, but students should have it installed & ready for them to use in class. You will need to run Rasa visualization function to create a graph * In the development phase you will need to customize and retrain your bot * Instructor should demonstrate and students should follow along customizing bot’s domain, stories, and NLU training data * Additionally, instructor needs to demonstrate and students will need to re-produce an updated graph after enhancing as a part of testing   **Exercise**- train, test & visualize your bot  Each group, from within your respective bot folder, train your respective bots.  Then try to chat with your bot.   * Take notes on your conversation * How smart was your bot?   Then adapt the code below from the in-class activity to produce the graph for your bot. **Remember** you will need to specify the correct name of the domain file and the stories file.  python -m rasa\_core.visualize -d domain.yml -s data/stories.md -o graph.png  Be prepared to share your graph and notes with the group.  **Note**: Be sure to check that all students are able to locate all components. Ask questions regarding the bot they built. Example: Did everything work as expected? Why or why not.  **Assignment**- weatherbot development  Instructor needs to prompt students to start working on assignment if they are finished early or the class as a whole is done early, if not, the assignment is a take-home exercise.  The assignment has samples and sample output with detailed notes within it. The students need to follow along the script, replicate steps, examine existing files and code, add more data to NLU using the tool demonstrated and also use Rasa’s Interactive Learning module to generate more stories to Core model as shown in the notes. They need to make sure their bot runs, they can visualize the stories, they can add more NLU and Core training data, retrain their bot and test it.  **Technical Notes:**  **Instructor Notes:** |

**Day 2-** Chatbots day 2: quick overview  
  
Lesson 1: Rasa UI Students will build & enhance two simple bots using Rasa UI

Lesson 2: Build a Custom drugbot students will follow the lifecycle steps for creating a custom bot in the pharmaceutical domain  
Day 2 - Objectives & Agenda at a Glance

|  |  |
| --- | --- |
| Time | Topics |
|  | **FYI- training team arrives hour before start to set up room.** |
| 15 min | **Introductions & Welcome**  **Review earlier sessions**- Revisit “muddiest” points from the previous session, parking lot and 3,2,1. All previous questions need an answer prepared. |
| 3 hrs | **Lesson 1: Rasa UI**   * Summarize and explore the capabilities and functionalities of Rasa UI * Demonstrate the use of Rasa NLU, Rasa Core & Rasa UI as HTTP servers * Use Rasa UI to create a simple bot * Use Rasa UI to train and test the Rasa NLU model * Practice replicating moodbot with the Rasa UI framework |
| 20 min  60 min | Break ( insert as needed throughout the day)  Lunch Break |
| 3 hrs | **Lesson 2: Build a custom drugbot**   * Discuss the data, its format, and features that will power the bot * Define bot requirements, specifications and conversational script * Scope out drugbot's architecture and its components * Explain the process of transforming of the tabular data from RxNorm to data/nlu\_data.json file * Explain the contents of bot\_domain.yml file * Explain the contents of bot.py file * Train NLU, Core and visualize initial bot stories * Interactively train dialogue of drugbot, re-train and test the bot |
| Debrief | Debrief- Revisit the class learning objectives. **Visit the parking lot**  **Hold an evaluation session where students complete the link to the survey** |

## 

## Lesson 1: Rasa UI Students will build and enhance two simple bots using RASA UI

|  |  |
| --- | --- |
| Rasa UI  Demonstrate    Discussion    Exercise 1- implement case study chatbots using Rasa UI | **Objective(s)**   * Summarize and explore the capabilities and functionalities of Rasa UI * Demonstrate the use of Rasa NLU, Rasa Core & Rasa UI as HTTP servers * Use Rasa UI to create a simple bot * Use Rasa UI to train and test the Rasa NLU model * Practice replicating moodbot with the Rasa UI framework   **Facilitation Guidance**   * Rasa UI is not the same as the Rasa stack; it is a user interface for developers built on top of Rasa stack (it’s an Optum product!) * Within Rasa UI you can: -Develop -Edit -Train -Test -Monitor, and -Maintain bots * Rasa NLU & Rasa Core can be used as a HTTP servers * After Rasa NLU, Rasa Core & Rasa UI servers are started the application is running. Students should be able to see the login screen when they paste http://localhost:5001 into their browser * After logging in ( UN: admin and PW: admin) they should see the Rasa UI dashboard * On the dashboard you will see agent. Agent= bot. Agents can be added, imported, and deleted * Within the Rasa UI, Agents can be edited ( add- intent, entities, actions and stories). Within each area you can add expressions, templates, etc. * When intents are now created expressions will need to be associated with them since this is what the NLU engine will use to train our model * All students should have the Rasa UI components already installed on their machine (except for Rasa Core, Rasa NLU and Rasa UI, PostgreSQL server is also installed and the database for Rasa UI is already pre-populated for them) * Conversation tab-view the recorded parse requests in the form of conversations in a messenger window * Insight tab- usage statistics * Configuration tab- view loaded NLU configuration * The files for the NLU models students have trained for a given agent will be stored in ./models/nlu/name\_of\_agent directory ( ./ = directory ) * **moodbot** (already in the Rasa UI set up, students need to glance at it and then delete first and then follow along to replicate it) Create an agent ( bot ) and name it **moodbot**   Populate intents and all other components as demonstrated by the instructor and shown in slides   * To train the bot go to the training tab to start training * Go the agents tab when you are to test, and the conversations will appear in the conversations tab. Be sure to select the agent/bot from the drop down to view the correct conversations.   **Exercise 1**- implement case study chatbots using Rasa UI  So far we have had very specific tasks for exercises.  Today will be more self directed.  On day one, you manually added self-generated training data to Markdown files. Today we will use a different tool set.  **Task**  In small groups, create bots from the day 1 case studies.   * Start with the conversational scripts and makrdown file from day one. * Populate the Rasa UI with the desired data:   + Intents   + Actions   + Stories   + Entities, etc.   What NLU configuration will you need?  For this exercise there is no **right** answer.  We do provide two **SAMPLES** of how this process could be approached.   * Simple Banking Chatbot * Simple Healthcare Chatbot   Please use these examples as references to assist you with the thinking process.  **Note:** This section has a lot of hands on and guided instruction. Frequent checks and questioning would be helpful to make sure students understand the UI and steps being explained.  **Technical Notes:**   * Rasa UI does not require Rasa Core integration, but we do demonstrate how to enable it and run Rasa Core server   **Instructor Notes:** |

## Lesson 2: Build a Custom drugbot

## Students will follow the lifecycle steps for creating a custom bot in the pharmaceutical domain

|  |  |
| --- | --- |
| Build a custom bot  Demonstrate    Discussion    Exercise 2: modifying drugbot | **Objective(s)**   * Discuss the data, its format, and features that will power the bot * Define bot requirements, specifications and conversational script * Scope out drugbot's architecture and its components * Explain the process of transforming of the tabular data from RxNorm to data/nlu\_data.json file * Explain the contents of bot\_domain.yml file * Explain the contents of bot.py file * Train NLU, Core and visualize initial bot stories * Interactively train dialogue of drugbot, re-train and test the bot   **Facilitation Guidance**   * RxNorm provides normalized names and unique identifiers for medicines and drugs; allows computer systems to communicate drug-related information efficiently and unambiguously * Students will not access the RxNorm dataset directly, instead the instructor will go over the steps taken previously to retrieve data as a reference in case they want to go to the data source and change their settings/queries, etc.   + Relevant data was downloaded via a SQL query   + Drugbot will use RxNorm data for querying and drug information pertaining to dosage form, brand, ingredients of the drug.   + The data is organized by : brand dose\_form ingredient suppress prescribable * The architecture for Drugbot will be similar to Rasa’s own chatbot examples and will have NLU data, stories, domain file, configuration file, and Python scripts * **Drugbot\_Data\_transform.ipynb** details the functions created to transform data to an appropriate format for drugbot to ingest. The data has already been prepared but take the time to examine it with students. This is the script the students would have to modify, if they are adding more intents, entities and slots to their **drugbot** * The NLU data produced by the above script is **chatbot/drugbot/data/nlu\_data.json** * We have pre-populated scripts and data, the students should follow along instructor’s directions to replicate the steps and then add more training data to the bot * Instructor trains the bot, students reproduce the steps * Instructor runs the bot to test it and visualize it, students reproduce the steps * The main focus of this part of the day is creating as many training data for stories as possible and making sure the students are comfortable with using interactive learning as well as understand how each of the parts of the bot structure is connected to each other (See Technical Notes) * Like day 1 Instructor re-trains the bot (NLU and Core), visualizes updated stories after an interactive learning session, and then runs the bot to test it; students should follow along to re-produce the steps   **Exercise 2**- modifying drugbot **Enhance drugbot** Individually or in pairs, we will think through adding functionality to Drugbot.  Below there is a series of questions to guide you through this process.  Again, for this exercise there is no right answer.  We expect you to use as many sticky notes and markers and you do lines of code in this exercise.  Please use these examples as references to assist you with the thinking process. **Question 1** What additional question could the bot answer if we used ALL of the data that we have in the drug\_data.csv? **Question 2** What parts of bot architecture would change if we were to implement our ideas? **Question 3** What additional elements would we need?  **Technical Notes:**   * There will be an additional file (**chatbot/drugbot/data/pre\_generated\_stories.md**) that students shouldn’t use for now: a file with generated stories that he/she can show the students after they have attempted to run interactive learning session on their own. If students have any trouble creating stories through interactive learning, help them out on individual basis * This bot requires at least 15 conversations or so in the stories to work as expected, students need to take time to do it after they have successfully added a couple of new ones * The end results will deviate from student to student, but the general format should be very similar   **Instructor Notes:** |