Lesson Plan

# Lesson: Welcome to Fundamentals I

|  |  |  |  |
| --- | --- | --- | --- |
| **Lesson Overview** | | | |
| **Lesson Length** | 3 hours (180 minutes) | | |
| **Overview** | This lesson will provide students with an understanding of the Fundamentals of Problem Solving Using Python I course, its current structure, student responsibilities, and schedule. This lesson will also familiarize students with coding terminology and the Jupyter Notebook tool. In addition, the lesson will cover basic steps of Computational Thinking (CompT). | | |
| **Objectives**   * **Domain:** Cognitive * **Level:** Understanding | Using conditionals, loops, Python dictionaries, local data, the CSV Library, and the Glob Library, students will be able to:   * TLO 1: Examine the implications of using computation to solve a problem   + ELO 1.1: Discuss best practices for using computation to solve a problem   + ELO 1.2: Suggest types of problems that can be solved through computation   + ELO 1.3: Show how computation can solve a problem * TLO 2: Recognize key computer science concepts   + ELO 2.1: Identify data types used in Python scripting   + ELO 2.2: Identify data structures used in Python scripting   + ELO 2.3: Define variables and strings   + ELO 2.4: Recognize how queries operate * TLO 3: Demonstrate the ability to build basic scripts using Python scripting language   + ELO 3.1: Use various data types and structures in Python scripting   + ELO 3.2: Collect data using Python scripting   + ELO 3.3: Extract data using Python scripting   + ELO 3.4: Develop advanced data structures using Python scripting | | |
| **Instructional Methods** | Informal lecture, demonstration, facilitated discussion. | | |
| **Assessment Strategies** | Informal: Guided and practical exercises  Formal: N/A | | |
| **Materials and Equipment Needed** | Required:   * SBU * Jupyter Notebook * Python   Optional:   * N/A | | |
| **Background Resources** | Resource:   * NGA SME * Technical facilitators   Subject matter/content questions may be referred to:   * Jeremy DeBrow, Course Manager   [Jeremy.R.Debrow@nga.mil](mailto:Jeremy.R.Debrow@nga.mil)  [Jeremy.R.Debrow@coe.ic.gov](mailto:Jeremy.R.Debrow@coe.ic.gov)  National Geospatial-Intelligence College (NGC) HDNPE Branch  Unclassified: 571-557-7583 | | |
| **Comments** | N/A | | |
| **Lesson Sequence** | | | |
| **Lesson Topic** | **Instructional Method** | | **Time**  **(mins)** |
| Introduction | Informal lecture, guided discussion | | 5 |
| Lesson: Welcome to Fundamentals I (PowerPoint) | Informal lecture, guided discussion, demonstration, guided exercise | | 45 |
| Lesson: Welcome to Fundamentals I (Jupyter Notebook) | Informal lecture, guided discussion, demonstration | | 125 |
| Conclusion | Informal Lecture | | 5 |
| **Lesson Outline** | | | |
| **Introduction** | | | |
| Introduction | * **Attention** (to be personalized by instructor) * **Motivation** (to be personalized by instructor) * **(1.1 and 1.2) Overview** (to be personalized by instructor)   + Learning objectives   + Lesson topics/main points * **Rules of Engagement** (to be personalized by instructor) | | |
| **Body** | | | |
| ***Lesson Topic*** | ***Main Points/Notes*** | ***Personalization*** | |
| Lesson: Welcome to Fundamentals I (PowerPoint) | **Informal Lecture (45 minutes)**  **Slide 2: Lesson Overview**   * During this lesson, the following topics will be discussed:   + Course Admin   + Community of Practice (CoP)   + Course Materials   + Activate Jupyter Notebook   + Computational Thinking   + Python   + Using Jupyter Notebook   **Slide 3: Rules of Engagement**   * Students will participate in discussions and challenges, ask questions, and respect all other participants   **Slide 4: Introductions**   * Welcome to the Fundamentals of Computer Science course. * Introduce yourself:   + What is your name?   + What do you hope to take away from this training?   + Have you used Python or another coding language before?   **Slide 5: Course Objectives**   * Using conditionals, loops, Python dictionaries, local data, the CSV Library, and the Glob Library, students will be able to:   + Examine the implications of using computation to solve a problem.     - Discuss best practices for using computation to solve a problem.     - Suggest types of problems that can be solved through computation.     - Show how computation can solve a problem.   + Recognize key computer science concepts.     - Identify data types used in Python scripting.     - Identify data structures used in Python scripting.     - Define variables and strings.     - Recognize how queries operate.   + Demonstrate the ability to build basic scripts using Python scripting language.     - Use various data types and structures in Python scripting.     - Collect data using Python scripting.     - Extract data using Python scripting.     - Develop advanced data structures using Python scripting.   **Slide 8: Course Structure**   * In-Person Lessons   + Monday – Friday (1 week)     - 0800 – 1100     - 1200 – 1500   **Slides 9: Course Schedule**   * Class:   + Monday – Friday: 0800 – 1100 & 1200 – 1500 * Open Lab:   + Monday – Thursday: 1500 – 1600   *(Facilitator Note: Update the schedule prior to Day 1 if there have been any changes to the schedule.)*  **Slide 10: Course Credit**   * To receive credit for this course you must:   + Attend in-person lessons   + Participate in informal discussions   + Complete individual and group exercises   + Complete the final exercise   **Slide 11: Feedback**   * Data Literacy Community of Practice (CoP)   + Post your questions on the Data Literacy Blog/Parking Lot   *(Facilitator Note: A walkthrough on the Data Literacy CoP will be provided later in this lesson.)*  **Slide 12: Security Policy**   * The following policies are in effect:   + Badges are required at all times within the building.   + If you signed for a badge, please return it to the Visitor Control Center (VCC) at the end of the day.   + One-way pagers are allowed in the building but must be turned OFF.   + PDAs are not allowed in the building.   + No external media are allowed (i.e., floppy disks, zip disks, thumb drives, etc.).   + Mind the classification level of computers and classroom.   + Computers are used for training purposes only.   **Slide 13: Important Phone Numbers**   * Course Lead: (571) 557-7583 * Registrar NCE: (571) 558-2968 * Registrar NCW: (314) 676-0459 * NGA Info line: (888) 333-INFO (4636) * Inclement Weather NCE: (703) 805-3030 * Inclement Weather NCW: (866) 281-7001   **Slide 14: Collaboration Policy**   * Students are encouraged to collaborate   + Sharing .ipynb files with classmates is not considered a form of collaboration   + When asked by a technical facilitator, students should be able to explain how each answer was generated   + Annotate who was part of the collaboration   **Slide 15: Things to Try When You’re Stuck**   * 6 things to try when you’re stuck on a problem.   + Reread what the problem is asking for   + Write out pseudocode   + Identify the data type at each step   + Use Google   + Talk to your neighbor   + Take a five minute break   **Slide 17: Community of Practice (CoP)**   * Navigate to the CoP   + Step 1: http://home.nga.mil/   + Step 2: Click the NGA College icon (lower screen).   + Step 3: Click the Blackboard icon (middle screen).   + Step 4: Enter User Name and Password (students should have used it recently to complete the pre-course tasks). If unknown, students can contact NGCLearningTechnologies@nga.mil or call 571-557-2700.   + Step 5: Select Communities of Practice (upper right screen).   + Step 6: Under My Organizations, select Data Literacy. * CoP Walkthrough   + Fundamentals of Problem Solving Using Python I & II:     - Review CoP splash page   *(Facilitator Note: Point out CSCI2011 important numbers.)*   * + Python Resources   *(Facilitator Note: Briefly review the resources with the class.)*   * + Data Literacy Blog/Parking Lot   *(Facilitator Note: Review and discuss the purpose of the blog/parking lot space. Students can access the blog/parking space by acknowledging the rules and etiquette.)*   * + Testimonials – How we work!   *(Facilitator Note: Videos developed by NGC LT have been made available to the students.)*   * + Course Content     - Click on CSCI2011 – Fundamentals of Problem Solving Using Python I / Course Content   *(Facilitator Note: Students have been asked to download these files prior to getting to class. They should be located within the H: (network) drive and ready to be dragged, dropped and unzipped to the documents folder.)*   * + - Solutions By Lesson   *(Facilitator Note: The Solutions By Lesson folder contains the solution files for all lessons 1 – 8.)*   * + Course Schedule     - Click on CSCI2011 – Fundamentals of Problem Solving Using Python I / Course Schedule   *(Facilitator Note: Schedule is the same as the one presented in Lesson 01 IP)*   * + Pre-Course Assessment:     - Click on CSCI2011 – Fundamentals of Problem Solving Using Python I / Pre-Course Assessment   *(Facilitator Note: Students should have completed the pre-course assessment prior to day 1 . Make students aware that the pre-course assessment is only used by NGC staff to analyze the transfer of knowledge between Lesson 1 and Lesson 8 of the Fundamentals of Problem Solving Using Python I course.)*  **Slide 18: Course Materials**   * CSCI2011\_CourseMaterials.zip * Go to your H: drive * Locate the CSCI2011\_CourseMaterials.zip file (downloaded prior to class) * Copy and paste CSCI2011\_CourseMaterials.zip to Libraries\Documents * Right Click on CSCI2011\_CourseMaterials.zip, select 7-Zip, select Extract Here * See below for properly extracted CSCI2011\_CourseMaterials.zip file to the Documents library (Image of Documents library with file extracted included in PPT)   *(Facilitator Note: CM and TFs should determine a second course of action for instances where students have not downloaded files to their H drive prior to class.)*  **Slide 19: Activate Jupyter Notebook**   * Go to the Start button (lower left). * Locate and select the Anaconda3 (64-bit) folder. * Select the Jupyter Notebook application. * The Jupyter Notebook prompt window will open – DO NOT CLOSE THIS WINDOW * Jupyter Notebook interface opens in your default browser * Open the Lesson 1 Jupyter Notebook file. (U\_CSCI2011\_L01\_Welcome\_to\_Fundamentals\_I\_SG\_V3.0)   **Slide 20: Activate Jupyter Notebook: Lesson 1**  *(Facilitator note: From this point forward, the lesson will be conducted from the Jupyter Notebook file.)* |  | |
| Lesson: Welcome to Fundamentals (Jupyter Notebook) | **Informal Lecture/Guided Discussion/Demonstration (125 minutes)**  *(Facilitator Notes:*   * *Have students load the U\_CSCI2011\_L01\_Welcome\_to\_Fundamentals\_I\_SG\_V3.0 file to begin the lesson.)* * *See Instructor’s Notebook for instructor guidance for discussion points and guided exercises.)*   **1.1. Objectives**  **1.2. Overview**  **1.3. Course Overview**   * This course is an instructor-led course, though its focus is different in two ways. First, it is designed to help shift how you think about solving problems. Second, its intent is to show you the potential application of coding as a tool to solve problems.   **1.4. Additional Assistance**   * Open Lab with Instructors (Monday – Thursday 1500 - 1600) * Community of Practice (CoP) – CSCI2011 Blog/Parking Lot   *(Facilitator Notes:*   * *This is a chance to explain to students the options they have for additional assistance during the course.* * *Discuss days and times for lab hours.)*   **1.5. Jupyter Notebook Basics**  *(Facilitator Note: Explain the concept of a cell and the benefits of having code broken into sections.)*  **1.6. Python**  *(Facilitator Note: This is a brief overview of the Python scripting language, including its origins, philosophy, and style guide.)*   * 1.6.1. Python 3.5   *(Facilitator Note: Explain that the version of Python that will be used throughout the course is v3.5 and that there are differences with other versions.)*   * 1.6.2. Some Vocabulary for Python   + Syntax   + Variable   + Comment   + Function or Method   + Arguments or Parameters   + Library * 1.6.3. Guided Exercise: Finding the Square Root   *(Facilitator Note: Walk the students through each step to demonstrate how to import libraries and how to call functions inside the library. Point out how these are always available to import once installed, but are only brought into your project on request.)*   * 1.6.4. Python Resources   *(Facilitator Note: The notebook provides common and popular resources that support Python scripting.)*  *(Facilitator Note: Open all of these [see notebook] and go over their contents.)*   * + - Resources Directory     - Glossary     - Student Handout     - “Parking Lot” * 1.6.5. Reserved Keywords and Built-In Functions   + 1.6.5.1. Keywords   + 1.6.5.2. Built-In Functions * 1.6.6. Python’s help() Function   *(Facilitator Note: Have students type "math" at the “input()” prompt for the first choice. After that, participants should make their own choices. As it says, when done, "just type "quit".")*  **1.7. Jupyter Notebooks Tips and Tricks**  *(Facilitator Note: Lesson 01 – Jupyter Notebook is meant to familiarize students with the Jupyter Notebook interface and its capabilities.)*   * 1.7.1. Keyboard Shortcuts   *(Facilitator Note: Poll the students for any other functionality they are interested in, or demonstrate some you feel is useful. They have not had much experience at this point. So, they may not know what to ask. Point out that this can be used as a reference so they can refer to it later, since they will continue to have access to this notebook.)*   * 1.7.2. The Kernel Menu   *(Facilitator Note: This exercise walks students through the workings of Jupyter and the concept of the kernel. It may not seem relevant to them at this point, but remind them of this portion once they get to while loops.)*   * 1.7.3. Output vs print()   *(Facilitator Note: this is an important clarification so that they understand that print is not synonymous with return once they get to functions. Explain that both the print and the output are just displaying the data that is stored in some variable.)*  **1.8. Computational Thinking**  *(Facilitator Notes:*  *Solving problems using Python or any language at NGA requires a clear understanding of computational thinking and how each component can be leveraged to further understand and answer problems. You will be expected to refer back to this section of Lesson 1 and address the four steps of computational thinking to solve problems as they relate to lessons, exercises, examples, student questions/comments, etc., throughout the course.*  *Use the embedded computational thinking graphic taken from CMPT1000. Throughout CSCI2011, refer back to this graphic when discussing the process of thinking computationally.)*   * The basic steps for approaching a computational problem are:   + Decomposition: Breaking down data, processes, or problems into smaller, manageable parts   + Pattern Recognition: Observing patterns, trends, and regularities in data   + Abstraction: Identifying the general principles that generate these patterns   + Algorithm Design: Developing the step-by-step instructions for solving this and similar problems * 1.8.1. Guided Exercise: Apply Computational Thinking   + Given the paragraph in the notebook, identify which sentence has the most words in it.   *(Facilitator Note: Work this out with the students. Highlight that the result should be to have a process that someone who does not understand the goal of what they are doing could follow and consistently get the right result.)*   * + Decomposition     - Make sure we understand the problem.       * What are the inputs?       * What are the outputs?     - If it’s a large-scale problem, identify possible smaller components.   + Pattern Recognition     - Inspect your data.       * Make sure the inputs and outputs you’ve identified make sense.       * Look for trends or possible fringe cases.   + Abstraction     - Work out an example by hand.   + Algorithm Design     - Write pseudocode.     - Identify tools available to you to replicate your solution.     - Implement!   **1.9. Practice Python**   * 1.9.1. Hard-Coding vs. Programmatic Coding * 1.9.2. Storing Values in Variables   **1.9. Appendix** |  | |
| Administrative Notes | N/A |  | |
| **Assessment** | | | |
| ***Assessment Type*** | ***Instructions/Prompts/Notes*** | | |
| Guided Exercise | See the facilitator notes located above for additional guidance. All exercises will be conducted inside the Jupyter Notebook lesson file. | | |
| **Conclusion** | | | |
| Conclusion | * **Final Summary** (to be personalized by instructor)   + Review learning objectives   + Review lesson topics/main points * **Remotivation** (to be personalized by instructor) * **Closure** (to be personalized by instructor) * **Next Lesson Introduction** (to be personalized by instructor) | | |