SeaGlide_TEAM	Python Basics	Lesson # 1 (60 minutes)

Description:
Technology is rapidly evolving and is becoming more integrated into our everyday lives. It is used in many ways to work smarter, not harder. However, technology is helpful to a certain degree -- it is useful when it is utilized properly and optimally. Hence, the more we learn about technology, the more use we can get out of it. Our goal is to learn the fundamentals of Python programming to better understand the Arduino codes utilized by SeaGlide. Today, we are going to begin learning about the basics of errors, debugging and block coding.

Students will be able to:

- Distinguish between coding language and natural language
- Define key terminology associated with programming and Python
- Understand errors (syntax, runtime versus semantic)
- Debug block code

Students will understand:
Python is a high-level language that is utilized in a variety of careers. Through this lesson, the students will learn the introductory vocabulary and logic associated with programming in Python. By using a free online platform (Bee: Debugging Activity), the students will correct given block codes by applying the learned logic in a hands-on manner. This lesson leads to learning syntax, variables and loops in Python.

Key Definitions & Concepts: [1]

- See the attached “Exploring Python: Definitions” Answer Key

Standards: [Copied from: 2]
3B.AP.11: Evaluate algorithms in terms of their efficiency, correctness, and clarity

- This standard connects to this lesson because the students will correct erroneous block codes and will edit the code to be as efficient as possible.

	Background Information	

Prior Knowledge:

- Pattern recognition
- Logical (step-by-step) thinking

Scientific Practices: [Copied from: 3] ● Constructing explanations ● Engaging in argument from evidence	**Core Ideas: [Copied from: 4]** ● Developing possible solutions ● Optimizing the design solution	**Cross Cutting Concepts: [Copied from: 5]** ● Observe and make use of patterns ● Identify and utilize structure and function

Possible Preconceptions/Misconceptions:

Since this lesson is designed to be the first interaction that the students will engage in computer programming, expect the students to miss important steps, to have lines of code in an improper order, etc. This lesson is designed to teach students the introductory vocabulary and to ingrain the importance of specific, logical steps.

Lesson Plan - 5E(+) Model

Engage:

The instructor will hand out the PB&J activity half sheet titled, *How to Make a Sandwich*. Students will proceed to write how they would prepare a PB&J sandwich. The instructor will then pick one set of students directions and prepare the sandwich *EXACTLY* as the student specified (plastic gloves are strongly recommended). Expect students to forget necessary steps. This will serve as an example for the students to understand that one must be precise, meticulous, and logical to achieve the desired result when programming. This activity should take less than 10 minutes to complete.

Explore:

Part I: Introduction

Students will begin to build a little background knowledge on what Python is. In pairs, they will research using a laptop or tablet to find the answers to the questions on a provided worksheet titled *Exploring Python: Introduction*. The purpose of this worksheet is to have students actively learn through research and to have a beginning understanding of what Python is and other related information. The teacher should facilitate an open class discussion to correct students' responses and address any misconceptions as necessary. Allow the students to take 5 minutes to complete the worksheet in their pairs, and take 5 minutes to discuss their answers. Hence, this section should take up to 10 minutes to complete.

Part II: *Benchmark Lesson*: Learning Vocabulary [1]

Without preface, the instructor will allot 5 minutes for students to individually complete the vocabulary matching worksheet titled, *Exploring Python: Definitions*. For a following 5 minutes, students will work with a partner to compare and to defend their answers. If students desire, they may change their answers after reasoning with their partner. As a class, the instructor will take up to 5 minutes to discuss and to correct any misconceptions as necessary. The purpose for this jigsaw activity and class discussion is to help the students understand introductory Python topics and logic. This worksheet and the class discussion should take up to 15 minutes to complete.

The lesson continues by discussing the differences between formal and natural languages by referencing the engagement activity, then leads to defining syntax, runtime, and semantic errors. This part of the lesson will be completed by having the students complete the *Exploring Python: Errors* worksheet. It is recommended to provide the students with the definitions of formal and natural language, and of syntax, runtime and semantic errors. Once the students have these definitions, the instructor should facilitate an open class discussion to complete the rest of the questions on the worksheet. The purpose of this worksheet is to have students make connections between the engagement activity and the current worksheet, and to expand their understanding of Python programming. Completion of this worksheet and of the open class discussion should take up to 15 minutes.

Part III: *Investigation Lesson*: Learning Logic and Introducing Loops [6]

The instructor will give each student access to a computer/tablet. The students will be engaged independently in an online block-coding activity titled "Bee: Debugging". This interactive website guides students through an 11 puzzle activity that helps them learn how to debug block code and primes them to condensing repetitive code through loops. Hence, each student will need access to a computer or tablet to complete this activity. Instructors should prompt the students to write down their codes on the final worksheet titled, *Exploring Python: Block Coding* (Note: It is appropriate to allot full credit for the students that clear through the levels and to award bonus points to those that take the initiative to condense the code via the available "repeat" function.) The remaining class time (up until the last 5 minutes reserved for the exit ticket) should be allotted for this activity. If students do not complete all 11 puzzles, it is recommended to assign the remaining puzzles for homework.

Explain:

Throughout the exploration, the students will engage in discussions that inquire their understanding and knowledge of the information at-hand. Thus, teachers will be informally asking students to explain all topics and relevant connections throughout the entirety of this lesson. The worksheets include "why" and "how" questions to provide the teacher with their individual explanations and to also give the students an opportunity to verbalize their understandings.

Elaborate:

The entirety of this lesson revolves around content that is used constantly -- logic and problem solving. Since the students would be primed to understanding the importance of concise and specific steps from the engagement activity and since they experienced block coding, then the students can be challenged to write pseudocode. The exit ticket is designed to take 5 minutes or less and prompts the students to write a pseudocode similar to the Bee activity. This leads to having students understand how to incorporate syntax and proper Python coding techniques.

Evaluate:

This lesson is designed with having both informal and formal evaluations throughout its entirety. The informal evaluations occur throughout the exploration because of the leading and open-ended questions and class discussions. This allows teachers to gauge surface-level student understanding. By surveying the students during completion of the worksheets and activities, teachers will be able to hear and to address any misconceptions or misunderstandings as necessary. The formal evaluations of this lesson are the Bee: Debugging activity and the exit ticket. This 5 minute, individual worksheet prompts the students to write a block code that converts degrees Fahrenheit to degrees Celsius. The purpose of the exit ticket is to evaluate whether or not the students grasped the lesson goals well enough to extend that knowledge into a mathematical equation.

Enrich:

This lesson could be extended by introducing proper Python syntax and instructing them to write Python programs. Since this lesson focuses on ensuring that the students understand the introductory vocabulary, error types and debugging, then they are successfully primed to learning how to write and debug original code. This lesson would also serve useful in an introductory science class by being a precursor to learning the scientific method. Since the scientific method requires that experimenters take specific, logical steps, and since it also may require debugging through re-experimentation, then this lesson serves as a fun and interactive way to introduce students to understanding the scientific method.

All associated documents are attached below
Reference *Annotated Bibliography* on the very last page of this packet

Name: ____ Date: ______________

How to make a Sandwich

Directions: In the left column of the table, list all of the steps necessary to make a PB&J Sandwich. Leave the right column of the table blank for future discussions.

Name: ____ Date: ______________

How to make a Sandwich

Directions: In the left column of the table, list all of the steps necessary to make a PB&J Sandwich. Leave the right column of the table blank for future discussions.

Name: ____ Date: ______________

Exploring Python: Introduction

Directions: Work with your partner to answer all of the following questions.

1. What is Python? ____
2. Who created Python? ____
3. When was Python released? ____
4. What does IDE stand for? ____
5. Why do we use an IDE? ____

   ____
6. What is IDLE? ____
7. Give four other examples of IDEs for Python. ________________________________

   ____
8. What are the advantages of coding in Python? _______________________________

   ____
9. What are the disadvantages of coding in Python? _____________________________

   ____
10. Why did the creator call it Python? ____________________________________

Name: ____ Date: ______________

Exploring Python: Definitions [1]

Directions: Pair each word with their proper definition.

Program	____	A: Executes a program in a high-level language by translating it one line at a time.
High Level Language	____	B: A way of using the Python interpreter by typing commands and expressions at the prompt.
Low Level Language	____	C: Data that is displayed on the screen or sent into a file or other device.
Portable	____	D: A programming language that is designed to be easy for a computer to execute.
Interpreter	____	E: The output of the compiler after it translates the program.
Compiler	____	F: A sequence of instructions that specifies how to perform a computation.
Source Code	____	G: Information gathered from the keyboard, a file or some other device.
Object Code	____	H: A property of a program that can run on more than one kind of computer.
Interactive Mode	____	I: A way of using the Python interpreter to read and execute statements in a script.
Script Mode	____	J: A program in a high-level language before being compiled.
Input	____	K: A programming language that is designed to be easy for humans to read and write.
Output	____	L: Translates a program from a high-level language to a low-level language all at once, in preparation for execution later on.

Name: ____ Date: ________________

Exploring Python: Errors [1]

Directions: Fill out the questions below based on the class discussions and through research.

1. Define formal language, and define natural language.

   What are some differences?

   Give an example of a formal language versus a natural language.

2. Define the following errors:

   Syntax:

   Runtime:

   Semantic:

   What are some differences?

   Give an example of the following:

   Syntax error:

   Runtime error:

   Semantic error:

3. What error did we experience during the PB&J activity? Why?

Name: ____ Date: ______________

Exploring Python: Block Coding [6]

Directions: Use the tablet/computer provided to complete the “Bee: Debugging” activity. Write all of your codes in the spaces below.

1:	2:
3:	4:
5:	6:

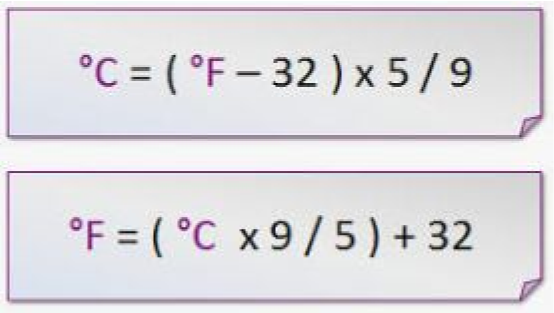
7:	8:
9:	10:

11:

Name: ____ Date: _________________

Exit Ticket

Prompt: Chemistry teaches us that temperature affects (nearly) everything in our universe. Notice that, depending where you live, temperature is measured on different scales. Given the following relationships, write Pseudocode, Block Code and/or Python Code (your choice) that will successfully convert a temperature in Fahrenheit to a temperature in Celsius.



°C = (°F – 32) x 5 / 9

°F = (°C x 9 / 5) + 32

Name: ______________________ANSWER_KEY___________________ Date: ________________

How to make a Sandwich

Directions: In the left column of the table, list all of the steps necessary to make a PB&J Sandwich. Leave the right column of the table blank for future discussions.

- Acquire and place on the table all ingredients, a knife and a plate needed to make sandwich
- Open bread
- Take two slices of bread out
- Open peanut butter
- Scoop peanut butter with the knife
- Spread peanut butter on one side of one slice of bread
- Put bread (peanut butter side up) onto the plate
- Wipe knife off
- Close peanut butter
- Open jelly
- Scoop jelly with the knife
- Spread jelly on the opposing side of the other slice of bread
- Close sandwich (jelly side onto the peanut butter side)
- Put knife in sink
- Close jelly
- Put peanut butter, jelly, and bread back where you found them
- Enjoy sandwich

Name: ______________________ANSWER_KEY____________________ Date: _________________

Exploring Python: Introduction

Directions: Work with your partner to answer all of the following questions.

1. What is Python? Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
2. Who created Python? Guido van Rossum
3. When was Python released? December 1989
4. What does IDE stand for? An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
5. Why do we use an IDE? Having an IDE has the following advantages: Compiling is usually "on the fly" which means no more switching to the command line to compile. Debugging is integrated, and having that in an IDE means that the step debugger actually uses your in-place editor to visually show you which code is executed.
6. What is IDLE? It is an IDE for Python specifically.
7. Give four other examples of IDEs for Python. Spyder, PyCharm, Thonny, Visual Studio Code, etc.
8. What are the advantages of coding in Python? Extensive support libraries, integration feature, improved productivity, etc.
9. What are the disadvantages of coding in Python? Difficulties in using other languages, weak in mobile computing, gets slow in speed, runtime errors, etc.
10. Why did the creator call it Python? Named after Monty Python.

Name: ______________________ANSWER_KEY____________________ Date: _________________

Exploring Python: Definitions [1]

Directions: Pair each word with their proper definition.

Program	__F__	A: Executes a program in a high-level language by translating it one line at a time.
High Level Language	__K__	B: A way of using the Python interpreter by typing commands and expressions at the prompt.
Low Level Language	__D__	C: Data that is displayed on the screen or sent into a file or other device.
Portable	__H__	D: A programming language that is designed to be easy for a computer to execute.
Interpreter	__A__	E: The output of the compiler after it translates the program.
Compiler	__L__	F: A sequence of instructions that specifies how to perform a computation.
Source Code	__J__	G: Information gathered from the keyboard, a file or some other device.
Object Code	__E__	H: A property of a program that can run on more than one kind of computer.
Interactive Mode	__B__	I: A way of using the Python interpreter to read and execute statements in a script.
Script Mode	__I__	J: A program in a high-level language before being compiled.
Input	__G__	K: A programming language that is designed to be easy for humans to read and write.
Output	__C__	L: Translates a program from a high-level language to a low-level language all at once, in preparation for execution later on.

Name: ______________________ANSWER_KEY_____________________ Date: _________________

Exploring Python: Errors [1]

Directions: Fill out the questions below based on the class discussions and through research.

1. Define formal language, and define natural language.

   Formal: languages that are designed by people for specific applications.

   Natural: languages that people speak.

   **Programming languages are formal languages that have been designed to express computations and operations.

   What are some differences?

   -Natural languages can be ambiguous and have multiple meanings when referring to the same word or concept.

   -Formal (programming) languages are unambiguous and literal. They can be understood entirely by analysis of the tokens and structure.

   Give an example of a formal language versus a natural language.

   Natural: find x + 3

   Formal: x+=3

   **Answers will vary here.

2. Define the following errors:

   Syntax:

   Error resulting from improper structure and/or token usage.

   Runtime:

   Error resulting from an exception occurring. This error occurs after the program begins running and is infrequently an encountered error.

   Semantic:

   Error resulting from improper logic. These programs run, execute and produce an output. However, the output is not desirable (doesn't do what you want).

   What are some differences?

   Obvious in the definitions.

   Give an example of the following:

   Syntax error:

   print((1 + 2)

   Runtime error:

   X / 0

   Semantic error:

   X += 3
   X = 2

3. What error did we experience during the PB&J activity? Why?

   **Likely Semantic, but the answers will vary here

Name: ________________________ANSWER_KEY____________________ Date: _________________

Exploring Python: Block Coding [6]

Directions: Use the tablet/computer provided to complete the “Bee: Debugging” activity. Write all of your codes in the spaces below.

1:

```
When run
Repeat (4) times
        Do move forward
Get nectar
```

2:

```
When run
Repeat (2) times
        Do move forward
        Get nectar
```

3:

```
When run
Move forward
Get nectar
Turn right
Move forward
Turn left
Move forward
Get nectar
```

4:

```
When run
Move forward
Move forward
Turn left
Move forward
Get nectar
Get nectar
Turn right
Move forward
Get nectar
```

5:

```
When run
Repeat (4) times
        Do move forward
Get nectar
```

6:

```
When run
Repeat (3) times
        Do move forward
Turn right
Repeat (5) times
        Do move forward
Get nectar
```

7:

```
When run
Move forward
Move forward
Repeat (5) times
        Do get nectar
Move forward
Repeat (5) times
        Do make honey
```

8:

```
When run
Repeat (3) times
        Do move forward
        Get nectar
```

9:

```
When run
Repeat (3) times
        Do move forward
        Get nectar
Turn left
Repeat (3) times
        Do move forward
        Make honey
```

10:

```
When run
Repeat (3) times
        Do move forward
        Get nectar
Turn right
Repeat (4) times
        Do move forward
        Get nectar
Turn right
Repeat (5) times
        Do move forward
        Make honey
```

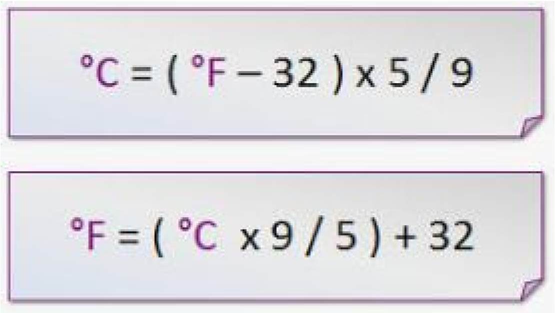
11:

```
When run
Repeat (4) times
        Do move forward
Turn left
Move forward
Repeat (5) times
        Do get nectar
```

Name: ______________________ANSWER_KEY___________________ Date: ________________

Exit Ticket

Prompt: Chemistry teaches us that temperature affects (nearly) everything in our universe. Notice that, depending where you live, temperature is measured on different scales. Given the following relationships, write Pseudocode, Block Code and/or Python Code (your choice) that will successfully convert and display 15 degrees Fahrenheit to a temperature in Celsius.



°C = (°F – 32) x 5 / 9

°F = (°C x 9 / 5) + 32

Pseudocode:

```
Define the given temperature.
Define the temperature conversion formula from Fahrenheit to Celsius.
Calculate and output the final answer.
```

Block Code:

```
When run
Define x = 15
Define y = (x - 32) * (5 / 9)
Print(y)
```

Python:

```
F = 15
C = (F - 32) * (5 / 9)
round(C, 2)    **Note: this line is optional because it rounds the value of C to two decimal places.**
print(C)
```

Annotated Bibliography

[1] Downey, A. (2012). *Think Python: How to Think Like a Computer Scientist* (Vol. 2.0.17). Needham, MA: Green Tea Press. From http://www.greenteapress.com/thinkpython/thinkpython.pdf

This online textbook was used for excerption within the Python Basics lesson plan as part of the Computer Science module. This reference aided in the completion of providing definitions for the key concepts and definitions sections and for the associated worksheets. This book was useful because of its layout and completeness. The lesson expands upon the material used form this book as it uses the material in the creation of worksheets and activities that are not provided in the textbook.

[2] Standards Aligned System. (n.d.). Retrieved from https://www.pdesas.org/

This website was used in each lesson in the Computer Science module to select proper Pennsylvania State standards, which are based in Common Core, that each lesson is centered around.

[3] Nsta. (n.d.). Science and Engineering Practices. Retrieved January 18, 2019, from https://ngss.nsta.org/PracticesFull.aspx

This website used in every lesson in the Computer Science module to find Standards for Scientific Practices that are applicable in each lesson.

[4] Nsta. (n.d.). Disciplinary Core Ideas. Retrieved from https://ngss.nsta.org/DisciplinaryCoreIdeasTop.aspx

This website was used in each lesson in the Computer Science module to select appropriate disciplinary core ideas set forth by the NSTA that are at the center of each lesson.

[5] Nsta. (n.d.). Crosscutting Concepts. Retrieved from https://ngss.nsta.org/CrosscuttingConceptsFull.aspx

This website was used in each lesson in the Computer Science module to selecting appropriate crosscutting concepts set forth by the NSTA that apply to each Python lesson.

[6] Course 2: Bee: Debugging #1. (n.d.). Retrieved from https://studio.code.org/s/course2/stage/10/puzzle/1

This website is an online tool designed to help students and beginning learners understand the logical thinking behind computer programming. Studio Code was used as the investigation of the exploration in the Python Basics lesson because it utilizes concepts learned through this lesson, but it also introduces loops. While loops are not covered through this lesson, the Course 2: Bee: Debugging guides students through an 11 puzzle activity that helps them learn how to debug block code and primes them to condensing repetitive code through loops.