Name:_______________________________________ Date:_______________________ Pd:_____

Unit 1: Python Basics

	My Grade	Date Completed
1.1 Hello World		
1.2 User inputs		
1.3 Simple Calculators		
1.4 Did I Fail? Calculator		
1.5 Letter Grade Calculator		
1.6 Unit Project		

Lesson 1: Hello World

Goal: Develop code that prints basic strings to a terminal

Key Terms to Know: String, terminal, IDE, Python, camelCase

Commands to Know:

1) print() - Prints whatever is in the ( ) to the terminal. Notice, print starts with a lower case p. To print a string, use " " inside the ( ). Example: print("this string")

Steps:

- ☐ Open VSCode
- ☐ Close any tabs that you currently have in your workspace
- ☐ On the left side file viewer create a new folder named "Unit1"
    - o No Spaces!
- ☐ Create a new file and call it "1HelloWorld.py"
    - o No Spaces!
    - o Done under "Save As" in the "File" menu
    - o Make sure you save it in the Unit 1 Folder!
- ☐ Using the print command outlined above, create code that prints "Hello World" to the bash terminal (bottom of the screen) when run
- ☐ Make sure the code works as intended
- ☐ Add two more print commands. Make one print a math operation (Ex: prints the answer of 2+2) and make the other print a string of your choice
- ☐ Confirm the code works
- ☐ Save the code
- ☐ Show project to teacher for grading

	Prints Hello World	Has one working print command showing the answer to a math problem	Has a third print command printing a string of choice that is appropriate	Code is efficient (contains minimal lines of code)
Possible Points	**5**	**2**	**2**	**1**
Earned Points				

_______/10 Points

Lesson 2: User Inputs

Goal: Develop code that responds to a user's input into the terminal and eventually reprints a message they enter in the terminal

Key Terms to Know: input, variable

Commands to Know:

1) input("String that prints to the terminal asking for input") – pauses code until a user input is put into the computer and the enter key is pressed. A message to the user can be printed if put inside the ( ) in " ".
2) = the equals sign is used to set variables equal to something. Example : x=5

Steps:

- ☐ Open VS Code
- ☐ Close any previous projects that you currently have in your workspace
- ☐ Create a new file and call it "2UserInputs.py"
  - No Spaces!
  - Make sure you save it in the Unit 1 Folder!
- ☐ Navigate into your User Input file editor tab
- ☐ Using the input command outlined above, create code that does the following:
  - 1) Prints "Hello!" to the terminal
  - 2) Asks the user to press enter on the next line in the terminal and pauses until the user does in fact press enter
  - 3) Prints to the terminal "Enter was pressed!" on the third line.
- ☐ Make sure the code works as intended
- ☐ Rewrite the code so the input is saved as a variable called "x" (remember, to set x, use "x =").
- ☐ Rewrite the comment you put in the Input function ( ) to state "Type a message and press Enter.".
- ☐ Have your code print the message the user entered to the terminal after they press enter
- ☐ Confirm the code works
- ☐ Save the code
- ☐ Show project to teacher for grading

	Input function pauses code and accepts an input from the user	A singular variable is correctly used	The user's message is reprinted to the terminal correctly	Code is efficient (contains minimal lines of code)
Possible Points	**5**	**2**	**2**	**1**
Earned Points				

________/10 Points

Lesson 3: Simple Calculators

Goal: Develop code that adds two user supplied numbers

Key Terms to Know: integer**,** concatenate

Commands to Know:

1) int() Takes an input and turns it into an integer number, which allows math operations to be performed to it.

Steps:

- ☐ Open VS Code
- ☐ Create a new file and call it "3AdditionCalculator"
  - o No Spaces!
  - o Make sure you save it in the Unit 1 Folder!
- ☐ Create code that prints a title when the code starts. For example, when my code starts, it prints "Welcome to the simple calculator".
- ☐ On the second line, make a code that prints instructions for the simple calculator for the user. For example, mine says "This Calculator adds two numbers supplied by the user."
- ☐ On the third line, make code that accepts the first number the user wants to add and stores it as a variable
- ☐ On the fourth line, make code that takes the second number the user wants to add and stores it as a different variable
- ☐ On the fifth line, make code that prints the result of the addition to the operator.
- ☐ Make sure the code works as intended
- ☐ You will have noticed that it did not add as you intended. By python rules, any input from a user is a string, meaning the numbers are seen as letters in the eyes of python. To do math on them, we need to convert them to integers (or whole numbers) using the command shown at the beginning of this sheet. Do that now!
- ☐ Run the code and see if it works now
- ☐ Make three more new files
  - o Name them 13SubtractionCalculator, 3MultiplicationCalculator, 3 Division calculator
- ☐ Make each file work as the name implies
- ☐ Confirm that each code works
- ☐ Save the code
- ☐ Show project to teacher for grading

	Addition calculator works as intended	Subtraction calculator works as intended	Multiplication calculator works as intended	Division calculator works as intended
Possible Points	**5**	**5**	**5**	**5**
Earned Points				

________/20 Points

Lesson 4: Did I Fail? Calculator

Goal: Develop code that creates a calculator that tells the user if they failed an assignment and tells them their grade as a percent

Key Terms to Know: NONE

Commands to Know:

<	Less than
>	Greater
<=	Less than or equal to
>=	Greater than or equal to
==	Is equal to
=	Gets set equal to
!=	Not equal to

if something:
    Do this
else:
    Do this instead
	Sees if something is happening, if it is it will perform the (1 time) indented block of code underneath it. Can have multiple lines under the If, but they all need to be indented the same amount. If the statement is not true, it will perform the else action: any code indented under the “Else:”

int( )	turns the argument in the ( ) into an integer
str( )	turns the argument in the ( ) into a string

Steps:

- ☐ Open VS Code
- ☐ Close any other projects that you currently have in your workspace
- ☐ Create a new file and call it “4FailCalculator”
    - o No Spaces!
    - o Done under “Save As” in the “File” menu
    - o Make sure you save it in the Unit 1 Folder!
- ☐ Navigate into your 1.4FailCalculator file editor tab
- ☐ Make code that does the following:
    - o Prints to the terminal stating the name of this app
    - o Prints to the terminal a description of the app so the user knows what it does
    - o Asks the user for how many points the grade was out of
    - o Asks the user for how many points they scored
    - o Calculates their grade as a percent and prints it to the terminal with clear description of what the number is
    - o Calculates whether the user passed and prints the answer to the terminal
- ☐ Make sure the code works as intended
- ☐ Save the code
- ☐ Show project to teacher for grading

	Calculates and correctly prints grade as a %	Correctly tells user if they passed or failed	All descriptions printed for user are clear and necessary	Code is efficient (contains minimal lines of code)
Possible Points	**10**	**5**	**3**	**2**
Earned Points				

________/20 Points

Lesson 5: Letter Grade Calculator

Goal: To modify the code from 1.4 to tell user their letter grade for an entered assignment

Key Terms to Know:

Commands to Know:

if something:
        Do this
elif something else:
        Do this
else:
        Do this
	Sees if something is happening, if it is it will perform the (1 time) indented block of code underneath it. Can have multiple lines under the If, but they all need to be indented the same amount. If the statement is not true, it will go to the next else if statement ("elif"). If none of the statements are true, it executes the else statement.

Steps:

- ☐ VS Code
- ☐ Create a new file and call it "5LetterGradeCalculator"
  - o No Spaces!
  - o Make sure you save it in the Unit 1 Folder!
- ☐ Open 4FailCalculator as well
- ☐ Navigate into your 4FailCalculator file editor tab
  - o Copy and paste all the working code from 1.4 into your empty 1.5 file
- ☐ Make changes to the code such that:
  - o instead of telling them if they passed or failed, it tells them their letter grade
  - o Make sure it still tells them their grade as a percent as well
- ☐ Make sure the code works as intended
- ☐ Save the code
- ☐ Show project to teacher for grading

	Calculates and correctly prints grade as a letter	Correctly tells user if they passed or failed	All descriptions printed for user are clear and necessary	Code is efficient (contains minimal lines of code)
Possible Points	**10**	**5**	**3**	**2**
Earned Points				

________/20 Points

Lesson 6: Solo Project

Goal: To create a program that shows what you have learned

Key Terms to Know:

Commands to Know:

Steps:

- ☐ Log into VS Code
- ☐ Create a new file and call it "6*NameOfYourProgram"*
    - o Replace *NameOfYourProgram* with a name that fits what your program does
    - o No Spaces!
    - o Done under "Save As" in the "File" menu
    - o Make sure you save it in the Unit 1 Folder!
- ☐ Navigate into your new file editor tab
- ☐ Create code that includes the following:
    - o at least one user input
    - o at least one integer math operation (+,-,*,/)
    - o at least one string
    - o at least one integer number
    - o at least one if…else…. or if…elif…else… statement
- ☐ Make sure the code works as intended
- ☐ Save the code
- ☐ Show project to teacher for grading

	At least one working user input	At least one working integer math operation	At least one working string	At least one working if else statement	Project works as a whole
Possible Points	**5**	**5**	**5**	**5**	**5**
Earned Points					

________/25 Points