Name:________________________________ **Partner:** __________________________

Python Activity 60: Java – Data Types & User Input

Java *is another programming language that shares some commonalities with Python. We explore Java to review our Python knowledge, and also have a greater appreciation for the Python we learned this semester!*

Learning Objectives

Students will be able to:

Content:

- Explain the difference between an ***interpreted*** and ***compiled*** programming language
- Summarize the process for running ***Java*** code
- Predict what Java code will do
- Compare and contrast Python & Java code and features

Process:

- Write Java code equivalents of Python code using data types and user input

Prior Knowledge

- Python concepts: classes, types, print, arithmetic

Conceptual Model: (Introduced by Instructor)

CM1. For the following statements, determine if they apply more to *Python*, *Java*, or *Both*:

Powerful language used by many programmers.	Python	Java
Features for making common programming tasks relatively simple.	Python	Java
Features for building large-scale systems design.	Python	Java
Must be compiled and run from Terminal.	Python	Java
Can run programs as scripts or interactively.	Python	Java
Dynamically typed: run-time error when variables used incorrectly.	Python	Java
Statically typed: compile-time error when variables used incorrectly.	Python	Java
Good fit for teaching programming to new computer scientists.	Python	Java
Good fit for large software projects, but steep learning curve.	Python	Java

Critical Thinking Questions:

1. The table below represents code that accomplishes the same task in Python on the left and in ***Java*** on the right:

Hello World: Python (hello.py) versus Java (Hello.java)

```python
# python

def main():
    print("Hello, World!")

 if __name__ == "__main__":
    main()
```

```java
// java

public class Hello {
    public static void main(String args[]) {
        System.out.println("Hello, World!");
    }
}
```

a. When we call `python3 hello.py` from the Terminal, what does the python interpreter do?

____

b. Complete the following table by writing an example of the programming language feature from the code above for both Python & Java:

Programming Language Feature	Python Example	Java Example
Comments	# python	
Function/Method declaration		
Determines what's run from Terminal		
Function parameters	( )	
Declaring parameter type	N/A	
Print to Terminal		
String syntax		
End of code block indicator	indentation	
End of code statement indicator	new line	
Returning value type		void
Access: public/private/protected		
Class method not tied to an instance	functions (sort of…)	static

Hint: Not everything in Python has a *<u>direct</u>* *Java equivalent, and vice versa!*

2. The table below is an interaction with the Terminal that runs our Python and Java code above:

Running Hello World from Terminal: Python (left) versus Java (right)

```
terminal% python3 hello.py
Hello, World!
```

```
terminal% javac Hello.java
terminal% java Hello
Hello, World!
```

a. How does running code in Python differ from running code in Java?

____

____

____

b. The table below explores running Java code from the Terminal further:

Running Java Code

```
terminal% ls # displays all files in this directory
Hello.java
terminal% javac Hello.java
terminal% ls
Hello.class    Hello.java
terminal% java Hello
Hello, World!
```

What happens when we call `javac Hello.java` from the Terminal?

____

What might happen when we call `java Hello` from the Terminal?
Hint: Which file might be being run?

____

FYI: ***Java*** is a compiled programming language. The Java ***Compiler*** converts our code into machine code that the processor can execute. Compiled languages require code to be manually compiled before execution.

c. What might be the benefit of using a compiled language over an interpreted language?

____

3. The tables below represent code that accomplishes the same task in Python and Java:

temp.py

```
def main():
  fahr = input("Enter the temperature in F: ")
  cel = (float(fahr) - 32) * 5.0/9.0
  print("The temperature in C is:" , cel)

 if __name__ == "__main__":
   main()
```

TempConv.java

```
import java.util.Scanner;

public class TempConv {
   public static void main (String args[]) {
       Double fahr;
       Double cel;
       Scanner in;

       in = new Scanner(System.in);
       System.out.print("Enter the temperature in F: ");
       fahr = in.nextDouble();

       cel = ( fahr - 32) * 5.0/9.0;
       System.out.println("The temperature in C is: " + cel);
   }
 }
```

a. What does this code do? ____

b. Complete the following table with examples of features from both Python & Java:

Programming Language Feature	Python Example	Java Example
Importing external libraries	from __ import * (not in this code)	
Variable declaration		Double fahr
Keyword for initializing a *new* object	N/A	
Prompt the user for input		
Read the input from keyboard		
Assigning a value to a variable		
Converts keyboard input to a decimal		
Arithmetic operations		
Printing multiple values		

Hint: Not everything in Python has a <u>*direct*</u> *Java equivalent, and vice versa!*

4. Match the Python data type on the left to its [closest] Java equivalent on the right:

```
bool
float
int
None
list
str
```

```
[]
null
String
Float
Double
```
(a decimal with *double* the capacity)
```
Integer
Boolean
Char
```

A few built-in Python data types not included here: `dict, range, set, tuple`

FYI: Java has two types of data types: ***primitive*** (non-objects) and ***Objects***. We'll mostly use the Object versions of the data types, and let the compiler handle conversions to primitives for us.

Application Questions.

1. Write a Java program that does the equivalent of the following Python program:

```
def main ():
  cups = input ("Enter the number of cups: " )
  ml = (float(cups)) * 236.5875
  ltrs = ml/1000
  print ("The mL is:" , ml, " or ", ltrs, " L")

if __name__ == "__main__":
    main()
```

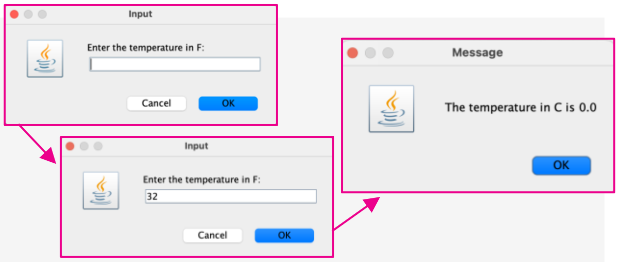
1. Observe the following code below, with its output pop-up windows shown to the right:

```
import javax.swing.*;

public class TempConvGUI {
    public static void main (String args[]) {
        Double fahr, cel;
        String fahrString;

        fahrString = JOptionPane.showInputDialog("Enter the temperature in F: " );
        fahr = Double.valueOf(fahrString);

        cel = (fahr - 32) * 5.0 / 9.0;
        JOptionPane.showMessageDialog(null, "The temperature in C is " + cel );
    }
}
```



Enter the temperature in F:

Enter the temperature in F: 32

The temperature in C is 0.0

a. Add in-line comments to the above code with what you believe each line is doing.
b. How does this code differ from the code in Question 3?

c. How might the *number of lines* of code differ for producing the Graphical User Interface in Java (above), versus in python?