BMI_1_Main_only (Differences between py and java)

1. Basic Syntax Rules
    a. Java code wrapped in a class.
    b. Squiggly brackets { in Java. Python block of code identified through indentation. Java block of code identified by using squiggly brackets.
    c. Some Java statements end in a semicolon;
    d. Python None versus Java null.
    e. Java uses data types when declaring a variable.
    f. Python line comment uses #. Java line comment is two slashes //.
    g. Java requires each program to have a public static void main(String[] args) method. Python does not!
2. Types of statements
    a. Import statement in Java.
    b. Python uses print function, Java uses System.out.print.
    c. Python uses print function, Java uses System.out.println.
    d. Java condition in if statement must be inside parentheses.
    e. No elif, instead its else if.
    f. Python if statement ends with colon while Java ends with a closing parentheses.
    g. Python else ends with a colon while Java else does not.
    h. Java condition in while statement must be inside parentheses.
    i. Python while statement ends with colon while Java ends with a closing parentheses.
3. Types of operators
    a. Add an int to a string in Java.
    b. Pyhton uses or, Java uses ||
4. Java API
    a. Java using Math.pow method versus Python ** operator.
    b. Python input function does two things: display prompt to user and obtain data entered. In Java, this is two separate statements.
    c. Java using Scanner to obtain user input.
5. More code in Java to do same thing that Python code does.

BMI_2_Functions (Differences between py and java)

1. Basic Syntax Rules
    a. Python Booleans are True and False; Java uses true and false
    b. Line comments in Python start with #, Java starts with two //
    c. Python uses None, Java uses null
    d. Python variables have no data type; Java variable declaration must include a data type.
    e. Java using public and private as access modifiers, Python has nothing like this!!!!!!!!!!!!!!!!
    f. Some Java statements end with a semicolon.
    g. Python using float builtin function versus Java declaring variables as type double.
    h. Java is using void. Is this a Java data type? No, void is not a data type; It acts as a placeholder instead of specifying a data type.
    i. Java requires each program has one method defined as public static void main(String[] args)
2. Types of statements
    a. Java import on two different types of exceptions
    b. Python try except, java try catch
    c. Python except clause is different from the Java catch clause. Python identifies type of exception; Java identifies type of exception and specifies a variable name.
    d. Python print function versus Java System.out.print and System.out.println
    e. Python print function allows for many arguments/parameters; Java System.out.print* allows you to display a single value (i.e., one argument/parameter)
    f. Python if and while statements end with colon; Java if and while statements end with a right parentheses.
    g. Python elif is a language keyword; Java it's else if (two keywords)
    h. Python else ends with colon; Java else does not.
    i. Java functions defined using static; Python functions do not mention static.
3. Types of operators
    a. Python uses and, Java uses &&
    b. Python uses not, Java uses !
    c. Python has ** operator, Java using Math.pow method
4. Java API
    a. Python uses input function; Java uses Scanner (in Java library aka Java API)
5. Code formatting
    a. Location of main function is different; Java it's at top, Python it's at bottom
    b. Java code is quite a bit longer than Python code.
    c. Python block of code denoted via indentation; Java block of code inside squiggly braces { }
    d. Java methods must be defined inside a class definition.

BMI_2_Functions versus BMI_2_Functions_Alt1

- Original has one validation method; Alt1 has two validation methods:
- Original: has three parameters; includes valid range.
    - `isInputValid(weight, 0, 1000)`
    - `isInputValid(height, 0, 100)`
- Alt1: has single parameter; valid range specified separately in each method.
    - `isWeightValid(weight)`
    - `isHeightValid(height)`
- Design for Reuse is better in the original when compared to Alt1.
- Separation of Concerns is about the same in both versions.

BMI_2_Functions versus BMI_2_Functions_Alt2

- Original has one validation method; Alt2 has one validation method.
- Original: has 3 parameters.
    - `isInputValid(weight, 0, 1000)`
    - `isInputValid(height, 0, 100)`
- Alt2: has 4 parameters, includes error message to display.
    - `isInputValid(weight, 0, 1000, "Weight must be a number in range (0,1000]")`
    - `isInputValid(height, 0, 100, "Height must be a number in range (0, 100]")`
- Design for reuse are about the same.
- Separation of Concerns is better in original when compared with Alt2.

BMI_2_Functions versus BMI_3_Methods_Obj

- Import Scanner; no longer importing two exceptions.
- We have BMI_3_Methods_Obj method that uses Scanner.
- More comments for each method.
- Ver 2 using static in each method definition; version 3 does not use static (except for main)
- Roughly twice as many methods in ver 3.
- Ver 3 line 52 double num = -1; This is not in ver 2.
- Ver 2 getNumber has a loop; ver 3 getNumber does not.
- Ver 2 main is “large”; ver 3 main only has two statements.
- Ver 3 has go method that looks a lot like ver 2 main.

BMI_3_Methods_Obj versus BMI_4_main & BMI_4_methods

- main() method in a different class.
- BMI_3: has main() as public; all other methods private. BMI_4_methods: two public methods (contructor, go); all other methods private.

BMI_4_main & BMI_4_methods versus BMI_5_main, BMI_5_methods1 & BMI_5_bmi

- BMI_5 calculations are now in a separate class.
- Use of the Java keyword (aka reserved word) **this**.
- BMI_5_methods1 class contains methods that appear to be more reusable.
- BMI_5_methods1 class imports two Exceptions.
- BMI_5_bmi has public getBMIcategory while BMI_4_methods has this as private.
- BMI_5_methods1 creates a BMI_5_bmi object; BMI_4 does not.
- BMI_5_methods1 is calling methods in the BMI_5_bmi class.
- BMI_5_methods1 class uses private for the Scanner instance variable.

BMI_5_methods1 versus BMI_5_methods2 versus BMI_5_methods3

- Methods3 has local variable `numericInput`.
- Methods3 using getValidInput which calls isInputValid.
- Methods2 using getValidWeight and getValidHeight each calling isInputValid.
- Methods1 using getNumber and isInputValid called within go().
- Methods2 and Methods3 has better separation of concerns when compared with methods1.
- Methods3 has better design for reuse when compared to methods1 and methods2.