Exercises 2
If, While, Pseudocode & Errors

TO SOLVE THESE YOU CAN **GOOGLE, COLLABORATE AND ASK** THE TEACHERS AS MUCH AS YOU WANT!!!!!

1. Which of these fragments are **valid** and **invalid** first lines of **if** statements? Explain why:
    - `if (x > 4)`
    - `if x == 2`
    - `if (y =< 4)`
    - `if (y = 5)`
    - `if (3 <= a)`
    - `if (1 - 1)`
    - `if ((1 - 1) <= 0)`
    - `if (name == "James")`

2. What is the **output** of the following code fragment? Explain why.

```
x = 2
if x > 3:
    print("This number")
print("is greater")
print("than 3.")
```

3. How can we **simplify** these code fragments?

```
if bool(a) == True:
    print("a is true")

if x > 50:
    b += 1
    a = 5
else:
    b -= 1
    a = 5
```

4. The following **Python program** is not **indented** correctly. Re-write it so that it is correctly indented:

```
def happy_day(day):
if day == "monday":
return ":("
if day != "monday":
return ":D"

print(happy_day("sunday"))
print(happy_day("monday"))
```

5. For what values of **input** will this program print **"True"**?

```
if not input > 5:
    print("True")
```

6. For what values of **absentee_rate** and **overall_mark** will this program print **"You have passed the course."**?

```
if absentee_rate <= 5 and overall_mark >= 50:
    print("You have passed the course.")
```

7. For what values of **x** will this program print **"True"**?

```
if x > 1 or x <= 8:
    print("True")
```

8. Eliminate **not** from each of these **boolean** expressions:

```
not total <= 2
not count > 40
not (value > 20.0 and total != 100.0)
not (angle > 180 and width == 5)
not (count == 5 and not (value != 10) or count > 50)
not (value > 200 or value < 0 and not total == 0)
```

9. **Rewrite** the following fragment as an if-ladder (using **elif** statements):

```
if temperature < 0:
    print("Below freezing")
else:
    if temperature < 10:
        print("Very cold")
    else:
        if temperature < 20:
            print(Chilly)
```

```
else:
    if temperature < 30:
        print("Warm")
    else:
        if temperature < 40:
            print("Hot")
        else:
            print("Too hot")
```

10. Write a **program** which uses a **while** loop to sum the squares of integers (starting from **1**) until the total exceeds **200**. Print the **final** total and the **last** number to be squared and added.

11. Write the **pseudocode** for a **program** which keeps prompting the user to guess a word. The user is allowed up to ten guesses – write your code in such a way that the secret word and the number of allowed guesses are easy to change. Print messages to give the user feedback.

12. Write now the **program** which you wrote the **pseudocode** for on the previous exercise.

13. Write a **program** which repeatedly prompts the user for an **integer**. If the integer is **even**, print the integer. If the integer is **odd**, don't print anything. Exit the program if the user enters the integer **99**.

14. Implement a simple **calculator** with a menu. Display the following options to the user, prompt for a selection, and carry out the requested action (e.g. prompt for two numbers and add them). After each operation, return the user to the menu. Exit the program when the user selects **0**. If the user enters a number which is not in the menu, ignore the input and redisplay the menu. You can assume that the user will enter a valid integer:

    -- Calculator Menu --
    0. Quit
    1. Add two numbers
    2. Subtract two numbers
    3. Multiply two numbers
    4. Divide two numbers

15. Find potential sources of runtime errors:

```
dividend = float(input("Please enter the dividend: "))
divisor = float(input("Please enter the divisor: "))
quotient = dividend / divisor
quotient_rounded = math.round(quotient)
```

16. Find the syntax errors in the code below and explain why they are errors.

```
myfunction(x, y):
    return x + y

else:
    print("Hello!")

if mark >= 50
    print("You passed!")

if arriving:
    print("Hi!")
esle:
    print("Bye!")

if flag:
print("Flag is set!")
```

17. Find potential sources of logical errors:

```
product = 0
for i in range(10):
    product *= i

sum_squares = 0
for i in range(10):
    i_sq = i**2
sum_squares += i_sq

nums = 0
for num in range(10):
    num += num
```

Last 10 minutes round the table what was hardest/most fun