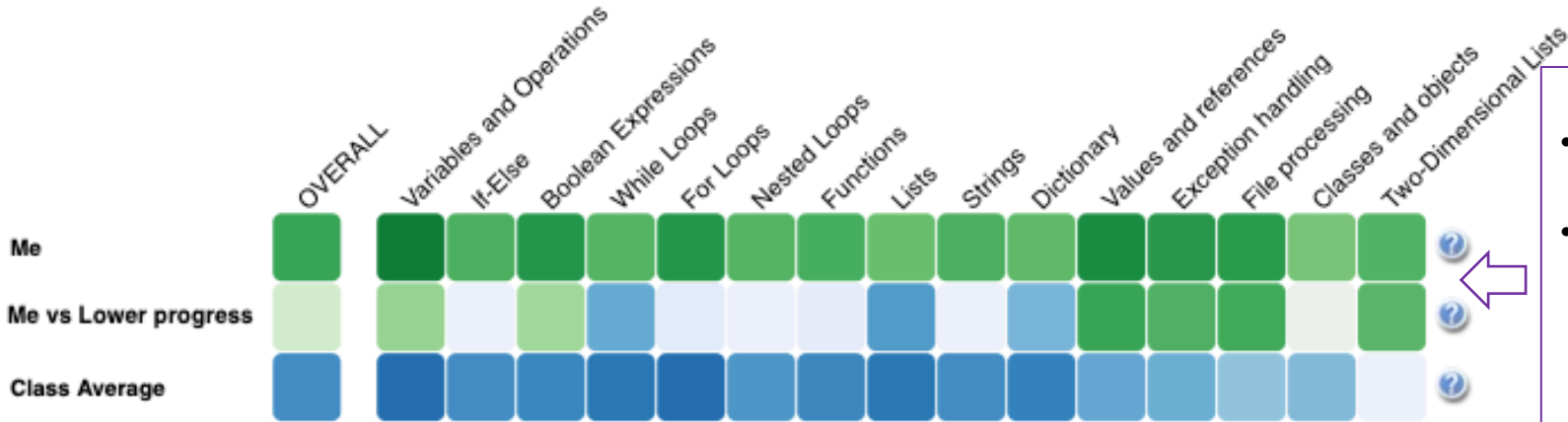


### Compared Group Explanation:

Your progress is compared to the average progress of students in the **lower half of the class** in terms of their **progress** in this system.



#### Me and group (Lower progress students)



### Progress Visualization

- First row (Me) shows **your progress** (Darker green means more progress on that topic)
- Second row (Me vs group) **compares your progress with only lower progress students** (Darker green means you have more progress than the group; darker blue means they have more progress than you; grey means equal progress).
- Third row (Group) shows the **average progress of 'lower progress students'** (Darker blue means more progress on that topic)

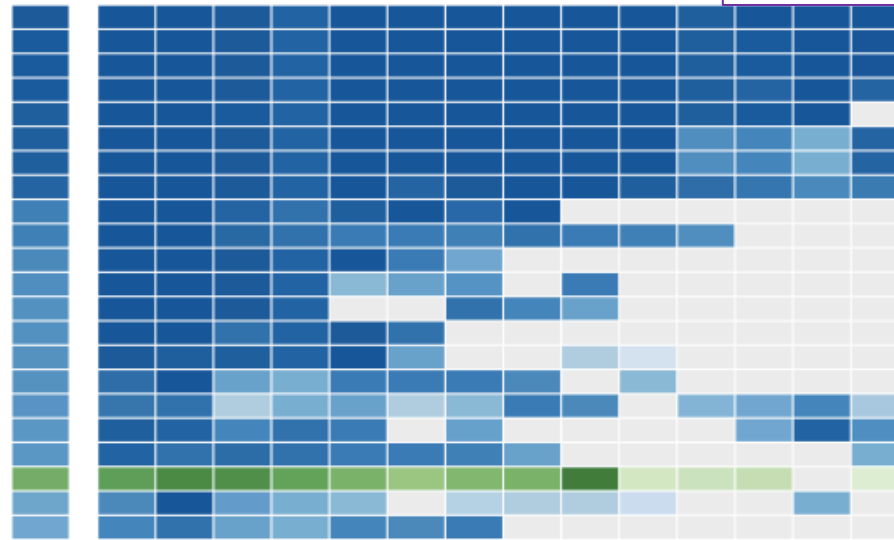
Load the rest of learners

Lower progress students (you are 20th out of 69)

Click the button above to load the list of other students (anonymized) and shows in which position you are in terms of progress.

**Note:** If you are not among the lower progress students (lower half of the class), your ranking will not be shown here. Your average progress in the system is higher than the lower half of the class.

20. Me ->



# How to Increase your Progress?

To have more greener cells on **Me** row, you need to interact with the learning activities inside each topic.

Click on a topic cell as shown below and access the contents. Viewing animation steps, clicking on example lines or solving challenges, questions and parsons problems to increase your progress.

Me and group (Lower progress students)



Load the rest of the grid

Animated Examples

Examples-Challenges

Questions

Parsons Problems

close

# Animated Examples

Play animation steps to how the program executed

```
1 account1 = 2540
2 account2 = 13250
3 price = 3400
4
5 can_afford = account1 >= price or account2 >= price
6 can_afford = account2 >= price or account1 >= price
7
8 account2 = account2 - price
9
10 money_left = account1 > 0 and account2 > 0
11 limit_exceeded = account1 < 0 and account2 < 0
```

Stack

Stack frame

account1  
2540

account2  
13250

Literals

>= or - > and

<

Text console

Fetching value 13250 - ready.

⏮ ⏪ ⏩ ⏭

# Questions

Predict the output of the program. It is either the console output or the value of **result** variable.

Tester.py

```
account1 = 186
account2 = 186 + 50
price = 250
can_afford = account1 >= price or account2 >= price
account2 = account2 - price
money_left = account1+account2 > 0
result = money_left
```

What is the final value of **result**?

**CORRECT!**

Your Answer is:  
True

Correct Answer is:  
True

Try Again

# Examples-Challenges

Check how a program is constructed line by line in examples and challenge yourself with challenges and complete the missing lines.

Example: Determining When to Buy a New Phone (Case 1)

Challenge Me!

Construct a program that determines whether it is time to buy a new phone based on the inputs that it receives from the user. A new phone should be bought if the phone breaks or the phone is at least 3 years old.

1 #Step 1: Read the user inputs
2 text = input("Enter the phone age in years:")
3 phone\_age = int(text)
4 text = input("Enter 1 if the phone is broken, otherwise enter 0:")
5 input\_num = int(text)
6 #Step 2: Determine whether the phone is broken or not
7 if input\_num == 1:
8 is\_broken = True
9 else:
10 is\_broken = False
11 #Step 3: Write the boolean expression to determine whether the student
12 need\_phone = is\_broken or phone\_age >= 3
13 #Step 4: Print the result
14 if need\_phone == True:
15 print("Yes! It is time to buy a new phone.")
16 else:
17 print("No! It is not yet the time to buy a new phone.")

Challenge: Determining When a Student Fails a Course (Case 2)

Back

Construct a program that determines whether a student fails the course based on the inputs that it receives from the instructor. The student fails the course if the exam score is less than 55 or when the student has cheated.

Drag a tile to each missing field to construct this program.

1 #Step 1: Read the instructor inputs
2 text = input("Enter the exam score:")
3 exam\_score = int(text)
4 text = input("Enter 1 if the student has cheated, otherwise enter 0:")
5 input\_num = int(text)
6 #Step 2: Determine whether the student has cheated
7 if input\_num == 1:
8 has\_cheated = True
9 else:
10 has\_cheated = False
11 #Step 3: Write the boolean expression to determine whether the student
12 is\_failing = not ( exam\_score >= 55 ) or has\_cheated
13 #Step 4: Print the result
14 if is\_failing == True:
15 print("Yes! The student fails the course.")

Drag a tile from here

CHECK

is\_failing = exam\_score < 55 or not has\_cheated

is\_failing = exam\_score < 55 or has\_cheated

is\_failing = exam\_score < 55 and not has\_cheated

# Parsons Problem

Reorder the program lines to solve the given task at the bottom of the screen. Pay attention to indentation.

Drag from here

Construct your solution here

else:

elif input\_a == 0 ?? input\_b == 0:

print(??)

print(??)

if input\_a == 1 and input\_b == 1:

print(??)

New instance Get feedback

Construct a program that mimics a XOR gate (exclusive or). When input\_a and input\_b are the same, it should print out 0 and in other cases print out 1.