



CODE CHALLENGE
EGYPT

PYTHON QUESTIONS

CODE CHALLENGE EGYPT

JUNIOR CATEGORY

20
23

Mission 1: Exam Grade Calculator

Overview:

An English teacher named Mr. Adam made an exam of total 30 marks for his students in the school. He wants to facilitate the way he classifies his students into 4 different classes as well as calculating the percentage of each student in the exam. He requested you to make a Python program to help him in that matter.

Mission:

Mr. Adam told you that he wants to enter the student's grade, then the program calculates the percentage rounded to the nearest 2 decimal points of that student in the exam, and classifying him as the following:

- From 25 to 30 → the student should get **Excellent**.
- From 20 to 24 → the student should get **Very good**.
- From 15 to 19 → the student should get **Good**.
- Less than 15 → the student should get **Redo the test**.

The program should be working as long as Mr. Adam doesn't enter -1, which will exit the program.

As a bonus part, add a case that handles any grade entered greater than 30 or less than -1 and print "Invalid grade!".

Useful Hints:

- ✓ Use the **conditional statements** to classify students correctly.
- ✓ Make sure to set the **correct range** for each grade group.
- ✓ You can use this formula to calculate the percentage:
 $(\text{student's grade} / \text{total grade}) \times 100$

Expected Output:

```
Enter the grade (-1 to quit): 29
The percentage is 96.67% and your grade is: Excellent
Enter the grade (-1 to quit): 22
The percentage is 73.33% and your grade is: Very good
Enter the grade (-1 to quit): 16
The percentage is 53.33% and your grade is: Good
Enter the grade (-1 to quit): 11
The percentage is 36.67% and you should: Redo the test
Enter the grade (-1 to quit): 40
Invalid grade!
Enter the grade (-1 to quit): -1
Exiting the program.
```

Mission 2: Password Strength Checker

Overview:

A software startup company is in the process of building a website to showcase its services and attract customers. As part of the website development, there is a need to implement a sign-up and sign-in form for users. To ensure the security of user accounts, the company has tasked you with developing a Python function to check the strength of users' passwords.

Mission:

You will work on the first phase of creating that function which name is `check_password_strength` that should perform the following tasks:

- 1- Accept a password as a parameter which can be letters or digits or both.
- 2- If the password's length is less than 6 characters, the function should ask the user to input a longer password.
- 3- It will check each character in the password, and if a character is a digit, lowercase letter, or uppercase letter, it should be considered and printed as a passing character. Else, the function will return that the password must contain letters and/or digits only.
- 4- After checking all characters, it should classify the password's strength based on its length as the following:
 - From 6 to 9 characters → **Weak password.**
 - From 10 to 14 characters → **Medium password.**

- Greater than or equals to 15 characters → **Strong password.**

To check your work, call your function on those 5 passwords:

['Evol4', 'Evol4you', 'Evol4you&me', 'Evol4u2bgr8', 'Evol4you2bgr8rep']

As a bonus part, add an empty list at the beginning that will collect all invalid passwords and print them at the end of the program.

Useful Hints:

- ✓ Develop the function to accept the password parameter as a **type string**.
- ✓ Implement **conditional statements** within the function to handle the password strength criteria and length-based classifications.
- ✓ Return **appropriate messages** or strength labels from the function.

CODE CHALLENGE
EGYPT

Expected Output:

```
Sorry, you cannot enter a password less than 6 characters  
Enter a longer one: alternate
```

```
char 1 passed  
char 2 passed  
char 3 passed  
char 4 passed  
char 5 passed  
char 6 passed  
char 7 passed  
char 8 passed  
char 9 passed
```

```
Its strength is: weak
```



First password result

```
char 1 passed  
char 2 passed  
char 3 passed  
char 4 passed  
char 5 passed  
char 6 passed  
char 7 passed  
char 8 passed
```

```
Its strength is: weak
```



Second password result

EGYPT

char 1 passed
char 2 passed
char 3 passed
char 4 passed
char 5 passed
char 6 passed
char 7 passed
char 8 passed

Third password result

Sorry, the password must contain letters and/or digits only
Its strength is: medium

char 1 passed
char 2 passed
char 3 passed
char 4 passed
char 5 passed
char 6 passed
char 7 passed
char 8 passed
char 9 passed
char 10 passed
char 11 passed
char 12 passed
char 13 passed

Fourth password result

Its strength is: medium

char 1 passed
char 2 passed
char 3 passed
char 4 passed
char 5 passed
char 6 passed
char 7 passed
char 8 passed
char 9 passed
char 10 passed
char 11 passed
char 12 passed
char 13 passed
char 14 passed
char 15 passed
char 16 passed

Fifth password result

Bonus part

Its strength is: Strong

The invalid passwords detected: ['Evol4', 'Evol4you&me']

Mission 3: Monthly Budget Tracker

Overview:

Your friend Mahmoud is having trouble managing his monthly budget and wants your help to create a Python program that will assist him. He needs a tool to track her income, expenses, and calculate his remaining budget for the month.

Mission:

Create a Python program that allows Mahmoud to input his monthly income and then let him to choose between 3 options:

- 1- Add an expense
- 2- View remaining budget
- 3- Exit

If he chooses the first option, he should input the name of the expense and its cost. Then, the app will tell him that the expense is added successfully. He can add his expenses one by one until he decides to stop. The program should deduct the cost of each expense added from his monthly income and keep a running total to calculate the remaining budget. So, he can add the expense if and only if its cost is less than or equals to his remaining budget.

At any point, Mahmoud can choose the second option to view his remaining budget and the expenses that he added. While the third option is only when he needs to exit the program.

As a bonus part, add a case if Mahmoud chose the second option to check the remaining budget before adding any expenses and print to him “No expenses added yet”.

Useful Hints:

- ✓ Use a **while loop** to continuously add expenses until Mahmoud decides to stop.
- ✓ Use **dictionaries or lists of tuples** to keep track of expenses and show them to Mahmoud.
- ✓ Handle **invalid inputs** gracefully (e.g., non-numeric values, negative values).

Expected Output:

The bonus part example

```
Enter your monthly income: $ 2000

Options:
1- Add an expense
2-View remaining budget
3- Exit
Enter your choice (1/2/3): 2
No expenses added yet.

Options:
1- Add an expense
2-View remaining budget
3- Exit
Enter your choice (1/2/3): 3
Thanks for using the Monthly Budget Tracker app!
```



Example of adding and showing expenses

Enter your monthly income: \$ 2000

1

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: Home rent

Enter the cost of the expense: \$ 700

Home rent added to your expenses.

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: Food

Enter the cost of the expense: \$ 500

Food added to your expenses.

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 2

Your monthly expenses: {'Home rent': 700.0, 'Food': 500.0}

Remaining budget for the month: \$800.00

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 3

Thanks for using the Monthly Budget Tracker app!



Example of negative inputs and how to deal with them

Enter your monthly income: \$ -2000

Monthly income cannot be negative.

Enter your monthly income (must be positive): \$ 2000

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: Home rent

Enter the cost of the expense: \$ -700

Expense cost cannot be negative.

Enter the cost of the expense (must be positive): \$ 700

Home rent added to your expenses.

Example of adding expenses exceeds Mahmoud's income

Enter your monthly income: \$ 2000

1

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: Home rent

Enter the cost of the expense: \$ 1200

Home rent added to your expenses.

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: Food

Enter the cost of the expense: \$ 700

Food added to your expenses.

Options:

2

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 1

Enter the name of the expense: School materials

Enter the cost of the expense: \$ 200

Sorry, this expense exceeds your current budget.

Options:

- 1- Add an expense
- 2-View remaining budget
- 3- Exit

Enter your choice (1/2/3): 3

Thanks for using the Monthly Budget Tracker app!

CODE CHALLENGE

EGYPT

Visit our
Facebook page



SCAN ME

For further questions:
Technical Support: 01022758877

Good Luck