

תרגיל 4: Decorators, Error Handling, Class

חלק א': 🧪

צור מחלקה בשם BankAccount עם:

- מאפיינים:

- owner_name שם הבעלים של החשבון.
- balance יתרת החשבון (מספר), מתחיל ב-0 כברירת מחדל.

- מתודות:

1. deposit(amount) מקבל סכום ומוסיף אותו ליתרה.
2. withdraw(amount) מקבל סכום ומוריד אותו מהיתרה.
- אם אין מספיק כסף, יש להדפיס הודעה: "Not enough balance"
3. display_balance() מדפיס את שם הבעלים והיתרה הנוכחית.

חלק ב': 🧪

כתוב דקורטור בשם uppercase_decorator שמקבל פונקציה שמחזירה מחרוזת, ומחזיר את אותה מחרוזת באותיות גדולות. (upper())

השתמש בדקורטור הזה על פונקציה שמחזירה "hello world".

דוגמה להפעלה:

```
Edit 🗂 Copy 📄 python

@uppercase_decorator
def greet():
    return "hello world"

print(greet())
# Output: HELLO WORLD
```

חלק ג':

כתוב פונקציה בשם `safe_divide(a, b)` שמבצעת חילוק בין שני מספרים:

- אם `b` הוא אפס – תחזיר `"Cannot divide by zero"`.
- אם אחד מהערכים אינו מספר `int` או `float` – תחזיר `"Invalid input"`.
- אחרת – תחזיר את תוצאת החילוק.

רמז: שימו לב לסוגי הטעויות שנקראים `ZeroDivisionError` ו-`TypeError`

פתרונות

פתרון:

```
Edit  Copy python

class BankAccount:
    def __init__(self, owner_name, balance=0):
        self.owner_name = owner_name
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        if amount > self.balance:
            print("Not enough balance")
        else:
            self.balance -= amount

    def display_balance(self):
        print(f"Owner: {self.owner_name}, Balance: {self.balance}")

# דוגמה לשימוש:
account1 = BankAccount("Alice")
account1.deposit(500)
account1.withdraw(200)
account1.display_balance()
```

פתרון – Decorator:

```
Edit  Copy python

def uppercase_decorator(func):
    def wrapper():
        result = func()
        return result.upper()
    return wrapper

@uppercase_decorator
def greet():
    return "hello world"

print(greet()) # Output: HELLO WORLD
```

פתרון – Error Handling:

```
Edit  Edit Copy  python

def safe_divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Cannot divide by zero"
    except TypeError:
        return "Invalid input"

print(safe_divide(10, 2))
print(safe_divide(10, 0))
print(safe_divide(10, "a"))
```
