

Object Oriented Programming Challenge

For this challenge, create a bank account class that has two attributes:

- owner
- balance

and two methods:

- deposit
- withdraw

As an added requirement, withdrawals may not exceed the available balance.

Instantiate your class, make several deposits and withdrawals, and test to make sure the account can't be overdrawn.

In [12]:

```
class Account:
    def __init__(self,owner,balance=0):
        self.owner = owner
        self.balance = balance

    def deposit(self,amount):
        self.balance = self.balance + amount
        print('Deposit Accepted')

    def withdraw(self,amount):
        if (self.balance - amount) >= 0:
            self.balance = self.balance - amount
            print('Withdrawal Accepted')
        else:
            print('Funds Unavailable!')

    def __str__(self):
        return f"Account owner: {self.owner}\nAccount balance: {self.balance}"
```

In [13]:

```
# 1. Instantiate the class
acct1 = Account('Jose',100)
```

In [14]:

```
# 2. Print the object
print(acct1)
```

```
Account owner: Jose
Account balance: 100
```

In [15]:

```
# 3. Show the account owner attribute  
acct1.owner
```

Out[15]:

'Jose'

In [16]:

```
# 4. Show the account balance attribute  
acct1.balance
```

Out[16]:

100

In [17]:

```
# 5. Make a series of deposits and withdrawals  
acct1.deposit(50)
```

Deposit Accepted!

In [18]:

```
acct1.withdraw(75)
```

Withdrawal Accepted

In [19]:

```
# 6. Make a withdrawal that exceeds the available balance  
acct1.withdraw(500)
```

Funds Unavailable!

Good job!