Object Oriented Programming Challenge - Solution

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 [1]:

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

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

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

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

In [2]:

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

In [3]:

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

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

In [4]:

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

Out[4]:

```
'Jose'
```

In [5]:

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

Out[5]:

```
100
```

In [6]:

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

```
Deposit Accepted
```

In [7]:

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

```
Withdrawal Accepted
```

In [8]:

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

```
Funds Unavailable!
```

Good job!