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 [ ]:

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

    def __str__(self):
        return "Account owner: Pavan \nAccount balance: 100"

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

    def widthraw(self, wd_amt):
        try:
            if  self.balance >= wd_amt:
                self.balance -= wd_amt
                print("Withdrwal accepted")
            else:
                print("Funds unavailable")
        except ValueError:
            print("valueerror for fund")
```

In [ ]:

```
# 1. Instantiate the class
```

In [ ]:

```
# 2. Print the object
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
# 5. Make a series of deposits and withdrawals
```

In [ ]:

```
# 6. Make a withdrawal that exceeds the available balance
```