

## OBJECT ORIENTED LAB

- Create a class called `BankAccount` that has the methods `withdraw` and `deposit` with no implementation.
- Create a class called `SavingsAccount` that **inherits** from `BankAccount`. `SavingsAccount` should have a constructor that only takes in a `self` argument. This constructor sets a property called `balance` to **500. (This should be the minimum balance at any given time)**.
  - In the `SavingsAccount` class, implement the `deposit` method that takes in cash deposit amounts, updates the balance accordingly and then returns the `balance`. For a negative deposit amounts, return `Invalid deposit amount`.
  - In the `SavingsAccount` class, implement the `withdraw` method that takes in the cash withdrawal amount, deducts this amount from the current balance and returns the balance. This method should **NEVER** allow the balance to get below 500. (Check for this condition and output `Cannot withdraw beyond the minimum account balance` if it happens). Also, output `Cannot withdraw beyond the current account balance` if withdrawal amount is greater than current balance. For a negative withdrawal amount, return `Invalid withdraw amount`.
- Create a class called `CurrentAccount` that **inherits** from `BankAccount`. `CurrentAccount` should have a constructor that only takes in the `self` argument and sets a property called `balance` to **0**.
  - In the `CurrentAccount` class, implement a `deposit` method that takes in cash deposit amounts, updates the balance accordingly and then returns the `balance`. For a negative deposit amount, return `Invalid deposit amount`.
  - In the `CurrentAccount` class, implement a `withdraw` method that takes in the cash withdrawal amount, deducts this amount from the current balance and returns the balance. For a negative withdrawal amount, return `Invalid withdraw amount`. Withdrawing more than the current balance should fail with message `Cannot withdraw beyond the current account balance`

Solution:

```
class BankAccount(object):
```

```
    def withdraw():
```

```
        pass
```

```
    def deposit():
```

```
pass
```

```
class SavingsAccount(BankAccount):
```

```
    def __init__(self):
```

```
        self.balance = 500
```

```
    def deposit(self, amount):
```

```
        amount = int(input("Please enter cash deposit amount: "))
```

```
        if amount < 0:
```

```
            return "Invalid deposit amount"
```

```
        else:
```

```
            self.balance += amount
```

```
        return self.balance
```

```
    def withdraw(self, amount):
```

```
        if amount < 0:
```

```
            return "Invalid withdraw amount"
```

```
        elif amount > self.balance:
```

```
            return "Cannot withdraw beyond the current account balance"
```

```
        elif (self.balance - amount) < 500:
```

```
            return "Cannot withdraw beyond the minimum account balance"
```

```
        else:
```

```
            self.balance -= amount
```

```
        return self.balance
```

```
class CurrentAccount(BankAccount):
```

```
    def __init__(self):
```

```
        self.balance = 0
```

```
def deposit(self, amount):
    amount = int(input("Please enter the deposit amount: "))
    if amount < 0:
        return "Invalid deposit amount"
    else:
        self.balance += amount
    return self.balance

def withdraw(self, amount):
    if amount < 0:
        return "Invalid withdraw amount"
    elif amount > self.balance:
        return "Cannot withdraw beyond the current account balance"
    elif (self.balance - amount) < 500:
        return "Cannot withdraw beyond the minimum account balance"
    else:
        self.balance -= amount
    return self.balance
```