

Class exercises

Canvas: [exercises-classes.zip](#)

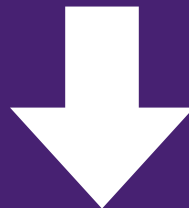
You can also see a description of the same
exercises below



1.

Create a class Counter1, with the following methods:

- constructor(initial)
- reset()
- increment()
- decrement()
- getValue()



1.

Example use:

```
var counter = new Counter1(10);  
counter.increment();  
counter.getValue(); // => 11  
counter.decrement();  
counter.decrement();  
counter.getValue(); // => 9  
counter.reset();  
counter.getValue(); // => 10
```

2.

Create a class Counter2, with the following methods:

- constructor(initial)
- reset(toValue)
- increment()
- decrement()
- getValue()



2.

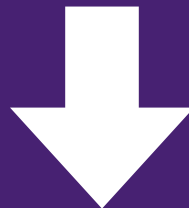
Example use:

```
var counter = new Counter2(10);  
counter.increment();  
counter.getValue(); // => 11  
counter.decrement();  
counter.decrement();  
counter.getValue(); // => 9  
counter.reset(20);  
counter.getValue(); // => 20
```

3.

Create a class Counter3, with the following methods:

- constructor(initial, step)
- reset(toValue)
- increment()
- decrement()
- getValue()



3.

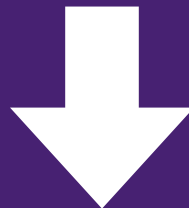
Example use:

```
var counter = new Counter3(0, 5);  
counter.increment();  
counter.getValue();    // => 5  
counter.decrement();  
counter.decrement();  
counter.getValue();    // => -5  
counter.reset(20);  
counter.getValue();    // => 20
```

4.

Create a class Account1, with the following methods:

- deposit(funds)
- withdraw(funds)
- getBalance()



4.

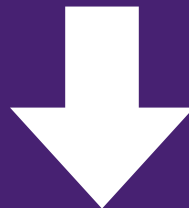
Example use:

```
var account = new Account1();  
  
account.deposit(100);  
account.getBalance(); // => 100  
account.withdraw(50); // => 50  
account.withdraw(100); // => "Insufficient funds!"  
account.getBalance(); // => 50
```

5.

Create a class Account2, with the following methods:

- constructor(pin)
- deposit(funds, pin)
- withdraw(funds, pin)
- getBalance(pin)



5.

Example use:

```
var account = new Account2("4332");

account.deposit(100, "0000"); // => "Wrong pin code!"
account.deposit(100, "4332");
account.getBalance("4332");   // => 100
account.withdraw(60, "0000"); // => "Wrong pin code!"
account.withdraw(60, "4332"); // => 60
account.getBalance("4332");   // => 40
```

6. (bonus)

Create a class Accounts, with the following methods:

- constructor(pin)
- addAccount(name, pin)
- getAccounts(pin)
- getBalances(pin)
- getBalance(account, pin)
- deposit(account, funds, pin)
- withdraw(account, funds, pin)



6. (bonus)

Example use:

```
var accounts = new Accounts("4332");

accounts.addAccount("savings", "4332");
accounts.addAccount("checking", "4332");
accounts.getAccounts("4332") // => ["savings", "checkings"]
accounts.deposit("savings", 100, "4332");
accounts.withdraw("savings", 20, "4332"); // => 20
accounts.getBalances("4332"); // => { savings: 80, checking: 0 }
accounts.getBalance("savings", "4332") // => 80
```

7. (extra bonus)

Create a console application
"bank" using the class from
Exercise 6

(using prompt/confirm/alert)