Exceptions: What They Are and How They Work

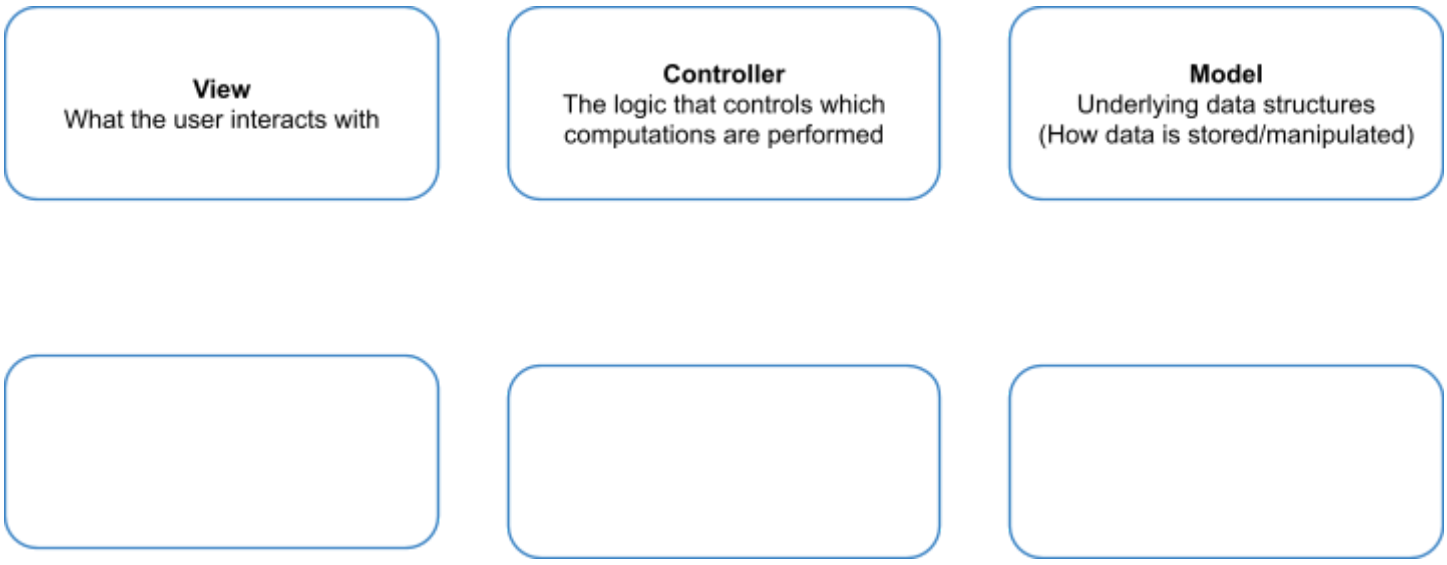
```
public Customer findCustomer(String custname) throws CustNotFoundExn {
   for (Customer cust:customers) {
       if (cust.nameMatches(custname)) {
           return cust;
       }
   }
   throw new CustNotFoundExn(custname); // instead of returning null
}

-------------------------

public class CustNotFoundExn extends Exception {
   public String custname;

   public CustNotFoundExn(String n) {
       this.custname = n;
   }
}
```

Model-View-Controller / Banking service recap



Thinking about how exceptions work

```
// BankingConsole (view)
public void loginScreen() {
  while(loggedIn) {
      // Read user and password
       try {
           loggedIn = c.login(user, pass);
           System.out.println("Logged in");
       } catch (CustNotFoundExn e) {
           System.out.println("No such user");
       } catch (LoginFailedExn e) {
           System.out.println("Wrong password");
       }
   }
}

// BankingService (controller)
public Boolean login(String custname, String withPwd)
       throws CustNotFoundExn, LoginFailedExn {
   Customer cust = customers.findCustomer(custname);
   if (cust.checkPwd(withPwd)) {
       return true;
   }
   // . . .
}

// CustomerList (model)
public Customer findCustomer(String custname)
   throws CustNotFoundExn {
   for (Customer cust : customers) {
       if (cust.nameMatches(custname)) {
           return cust;
       }
   }
   throw new CustNotFoundExn(custname);
}
```

```
public class CustomerList {
   private LinkedList<Customer> customers;

   public Customer findCustomer(String custname) throws CustNotFoundExn {
       for (Customer cust:customers) {
           if (cust.nameMatches(custname))
               return cust;
       }
       throw new CustNotFoundExn(custname); // instead of returning null
   }
}
-----------------------------

public class BankingService {
   CustomerList customers;

   public Boolean login(String custname, String withPwd)
           throws CustNotFoundExn, LoginFailedExn {
       Customer cust = customers.findCustomer(custname);
       if (cust.checkPwd(withPwd)) {
           return true;
       } else {
           throw new LoginFailedExn(custname);
       }
   }
}
-----------------------------

public class BankingConsole {
   BankingService controller;

   public void loginScreen() {
       boolean loggedIn = false;
       while (!loggedIn) {
           // Prompt for input
           try {
               loggedIn = controller.login(username, password);
               System.out.println("Thanks for logging in");
           } catch (CustNotFoundExn e) {
               System.out.println("No such user " + e.custname);
           } catch (LoginFailedExn e) {
               System.out.println("Password mismatch for " + e.custname);
           }
       }
   }
}
```

Speeding Up Access to Customers

```
public class AccountList {
   private LinkedList<Account> accounts = new LinkedList<Account>();

   public Account findAccount(int forAcctNum) {
       for (Account acct:accounts) {
           if (acct.numMatches(forAcctNum))
               return acct;
       }
       return null; // not yet converted to exceptions
   }
}
```