Today's topic:

- Designing classes using CRC Cards
CRC Cards were first proposed by Kent Beck and Ward Cunningham [link](http://c2.com/doc/oopsla89/paper.html).

“The most difficult problem in teaching object-oriented programming is getting the learner to give up the global knowledge of control that is possible with procedural programs, and rely on the local knowledge of objects to accomplish their tasks. Novice designs are littered with regressions to global thinking: gratuitous global variables, unnecessary pointers, and inappropriate reliance on the implementation of other objects.”

CRC refers to:
- **Class** or (Candidate Class) – the name of the class.
- **Responsibilities** – what the class is responsible for doing.
- **Collaborations** – what other classes the class collaborates with.

What are CRC Cards?

A short description of the class.

A list of responsibilities of the class.

A list of the classes it collaborates with.
Why use cards?

- Cards can be *moved* around, *grouped* in different ways or thrown away.
- They can aid in thinking in terms of *objects*.
- They are *limited in size*, and can help keep your classes *cohesive* without adding too much.
- They allow for *group* participation (programmers and users) in developing and understanding the design.

CRC card example

- Suppose we need to create a time card system:
  Employees should be able to enter their total hours working, vacation time and sick leave for each day of the month.
**CRC card process**

- **Step 1: Brainstorm Candidate Classes**
  - Identify as many *potential classes* that may be in the problem domain and write them in a list.
  - For the initial design only focus on the classes required to simulate the work. The interface design will come later.
  - Focus on the *nouns* or “things” in the problem domain.
  - Remember that all ideas are *potentially* good, not all of them will be included in the final design.

**CRC card example**

- **Step 1: Brainstorm Candidate Classes**
  - Employee
  - Time Card
  - Time at Work
  - Vacation
  - Sick Leave
  - Month
  - Week
  - Day
  - Hours Worked
  - Vacation Hours
  - Sick Hours
**CRC card process**

- Step 2: Filter and Refine the List

- Review the list you created in step 1 and come up with a refined list.
  - Are some candidates *attributes* of others?
  - Are some *instances* of others?
  - Does the candidate *know* things (data)?
  - Does the candidate *do* things (responsibility)?

**CRC card example**

- Step 2: Filter and Refine the List

- Employee
- Time Card
  - Time at Work
  - Vacation
  - Sick Leave
- Month
- Week
- Day
  - Hours Worked
  - Vacation Hours
  - Sick Hours
  - Duplicate of Hours Worked.
  - Duplicate of Vacation Hours.
  - Attribute of a day.
Step 3: Create cards

- From the revised list, create CRC cards.
  - Write down name at top of both sides.
  - On one side right a brief description of the class.
  - On the other side make space for responsibilities and roles.

CRC cards example

Step 3: Create cards

- Month
- Week
- Day
- TimeCard
- Employee
- The person submitting a time card.
Step 4: Identify responsibilities and collaborations using scenarios.

- Develop scenarios that will occur and play out the scenario to identify responsibilities and collaborations.
  - Play out the scenario with the cards. What object has the responsibility for each task?
  - What other objects does the class collaborate with to complete a scenario?
  - Write these responsibilities and scenarios on the cards.

---

**CRC card example**

- Step 4: Identify responsibilities and collaborations using scenarios.
  - Scenario 1: Bob records 9 hours worked for February 1st.

<table>
<thead>
<tr>
<th>Employee</th>
<th>TimeCard</th>
<th>Day</th>
<th>TimeCard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters hours worked</td>
<td></td>
<td>Knows hours worked</td>
<td></td>
</tr>
<tr>
<td>TimeCard</td>
<td></td>
<td>Day</td>
<td></td>
</tr>
<tr>
<td>Records hours worked</td>
<td>Day</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 4: Identify responsibilities and collaborations using scenarios.
- Scenario 2: Bob records 8 hours sick leave for February 4th.

Employee
- Knows name
- Enters hours worked
- Enters sick leave hours

Day
- Knows hours worked
- Knows sick leave hours

TimeCard
- Records hours worked
- Records sick leave hours
- Day

Step 5: Finalize classes.
- Throw out any cards that do not have any responsibilities and collaborations.
- Make sure you cannot think of any other scenarios.
- You may end up with some cards that do not have any responsibilities or collaborations.
Step 5: Finalize classes.

Get rid of the Month and Week cards, because they did not have any responsibilities defined.

CRC Cards recap

Step 1: Brainstorm Candidate Classes
Step 2: Filter and Refine the List
Step 3: Create cards
Step 4: Identify responsibilities and collaborations using scenarios.
Step 5: Finalize classes.