Here are some general comments that recurred on the feedback of the Phase I of the checkbook for you to consider in your final revisions:

1) Your first task is to identify all the core concepts in the problem either by defining classes that represent the essence of the concept or by defining relationships between objects of classes demonstrating shared responsibility or cooperation.

2) The best way to test your model is to read the syntax of your symbols to yourself “OUT LOUD” and listen to what the symbols “say!” If the reading of the symbols syntax (class, attributes, services, relationships) matches up with the problem scenario and the actions needed to satisfy the requirements then you’re on the right track. Usually reading them “out loud” to yourself will help you catch problems with cardinality (“How many of these belong to how many of these?”) as well as “ownership” (“If one of these goes away do these related to them go away also?”) and finally inheritance (gen/spec) (“Object of this child class IS A object of the parent class?”)

3) Pat’s categories may seem arbitrary (food, rent, entertainment, . . .) but when we introduce the probability that the checkbook assistant will support tax preparation, those categories take on a more focused purpose. If Pat’s bright enough to align the categories with aspects of the tax form’s input requirements, then “keeping track of expenses” can easily be expanded to track expenses that relate to taxes as well as income that is taxable.

4) The whole purpose of defining classes is to identify a framework, a template, of structure and behavior that allows many individual instances of a concept to be described only once in the class (e.g. many specific students in the student class, many specific courses in the course class, etc.). The key is to find a template that lets you plug in any specific attribute values to depict a specific instance of the class. So, the skill is “generalization” (how are all these individuals alike in structure [data attribute variables, and services]). *(Did you notice the reference to the OO Ontology! That’s where all these model concepts come from! Maybe you should take 15 minutes and review that again!)*

5) You have several model examples from the class to use as frameworks for your model and documentation. The most compact is the last Zoo handout that’s posted on the web. It eliminates redundant terms and highlights the basic elements that your next submission must contain: Class diagram with descriptions for each element (class, attribute, service, relationship), Use Case’s describing the interaction of Pat with the checkbook assistant, and Sequence Diagrams to depict the sequence of messages exchanged between objects in your class diagram demonstrating what ACTIONS occur in what order to complete a task that satisfies Pat’s needs. There are also the UML Guidelines and the Association and Class example handouts not to mention the Fowler text for inspiration!!! *Model On!*
 Courtesy of one of your classmates!