Homework 03, CPSC-4175

Chapter 12, Object-Oriented and Classical Software Engineering

August 31, 2017

1. What two kinds of things must a *software requirements specification* contain?
2. Give an example of an ambiguous requirement in English (that is, using a natural language). Explain the ambiguity.
3. Consider the domain of processing student grades. Draw a simple *data flow diagram* of the process. Create the drawing as a PDF and upload the file to your Github repository. Explain the drawing in English as your answer to this question. **Use Gane and Sarsen notation in your answer.**
4. Consider the domain of an online shopping cart. Draw a simple *finite state machine* of the process. Create the drawing as a PDF and upload the file to your Github repository. Explain the drawing in English as your answer to this question.
5. Consider the domain of a database with customer, products, and orders tables. Draw a simple *entity-relation diagram* of the database. Create the drawing as a PDF and upload the file to your Github repository. Give an account of the data structure in English as your answer to this question.
6. Consider the project you have chosen as a team. Draw a simple *SADT* (structured analysis and design technique) diagram of the project. Create the drawing as a PDF and upload the file to your Github repository. Explain your project on a high level in English as your answer to this question.
7. This is not in the book. Consider the project you have chosen as a team. Draw a simple *SDL* (Specification and Description Language) diagram of the project. Create the drawing as a PDF and upload the file to your Github repository. Explain your project on a high level in English as your answer to this question.
8. Find an image or document online of a *simple* function using Z. Convert the image or document to a PDF and upload it to your Github repository. Explain the specification in English as your answer to this question. Note that the Z language is difficult and takes years to learn. This course is not a course in Formal Methods, where you might spend an entire semester learning the Z language. You don't have to understand the Z specification. All you have to do is have some faint idea about its concept and operation.