Listen

TidBIT

The software design template is the first step in your journey towards learning to communicate technical information to nontechnical audiences. Business and technical information need to be clear and concise. Software design templates are not English essays or academic theses - learn to state facts in a tight vernacular that gets the point across. One trick is to try removing unnecessary words. Avoid using 20 words when 10 will do the job.

In a software design template, the headings are a guide to follow. The executive summary should reveal what the system will do and what type of system (client-server, web-based, mobile, and so on) will be built. The design constraints section is a place to say what is "in" and what is "out." For example, if the company uses all Windows-based desktops, laptops, and servers, you would likely not propose using Mac OS and Swift to develop a new application. If you are proposing a cloud-based design and the client is a government agency, then you would most likely be using Amazon's GovCloud, which is certified for government use. These are two examples of reading, analyzing, and inferring information from the requirements given that will impact the technical design.

Required Resources

Reading: *Hands-On Design Patterns with Java*, Chapter 3 , Chapter 4, and Chapter 5
As you read, consider the following:

- What are some common challenges that can be solved using behavioral and creational design patterns?

- What are some other examples of problems or challenges that you use design patterns to solve?

Reading: Design Constraints PDF (Course%20Documents/CS%20230%20Design%20Constraints.pdf?ou=1798772)
When working with a client, you will likely be provided with a set of requirements that they will be looking to meet. It is an important skill to translate these requirements into design constraints. As you review this resource, consider how various technical and business requirements translate into constraints that influence how something can be designed and developed.

Reading: How to Write a Good Software Design Doc (https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/)
Software designs and design documents can be as varied as the software itself. There are common types of information to include in a design document and this article explores why we write a design document, what to include, and the processes that you should go through with it. This document provides a framework and best practices for using software design templates. How are system goals defined before the design process begins?

Reading: Painless Functional Specifications - Part 1: Why Bother? (https://www.joelonsoftware.com/2000/10/02/painless-functional-specifications-part-1-why-bother/)
Outspoken industry leader Joel Spolsky is to the software business what Elon Musk is to the automotive business - a disruptor. Joel has built teams and software and written about his experiences along the way. His vision of an online Q&A site became Stack Overflow, the number one site for technology, with over 50 million visitors each month. In this reading, Joel walks you through an imaginary process of developing software with and without specifications.

Reading: Java Packages PDF (Course%20Documents/CS%20230%20Java%20Packages.pdf?ou=1798772)
You probably have subfolders under your Documents folder, either on your own computer or an online storage system like Google Drive, for each of the courses you have taken. Packages are an organizational tool used primarily to aid developers in organizing their code. Java packages allow developers to group related classes, interfaces, and other files together by giving them a name. Review this document to learn more about packages.

Additional Support (Optional)

Reading: Systems Analysis and Design with UML, Chapter 3 (https://go.oreilly.com/SNHU/library/view/systems-analysis-and/9781118037423/08_chapter003.html#ch003-sec002)
Review sections 1. Introduction and 2. Requirements Determination to gain a better understanding of system and business requirements that are used to identify and meet the needs of a client. Requirements can be used to inform the software design document you prepare for a client and a team of developers.

Reading: Design Patterns for Beginners - With Java Examples (https://www.springboottutorial.com/design-patterns-for-beginners-with-java-examples)
This article provides in-depth details and examples of the Gang of Four design patterns as well as embedded tutorial videos you can watch. This is not required reading, however it is valuable information that will help you to move from theory to practical use of design patterns.

Reading: *Hands-On Design Patterns with Java*, Chapters 3–5
To learn more about software design patterns, read the chapters on behavioral, creational, and structural design patterns. This reading provides a high-level overview of several design patterns in addition to use cases for each.