Listen

TidBIT

Identifying the right classes for a system is a skill that you will continue to develop over time. Remember that the classes you identify should support the object-oriented principles of abstraction, encapsulation, polymorphism, and reusability. It is common for beginning designers to identify too many classes. The system can still be developed, but it might be overly complex and difficult to maintain. Ask for feedback from instructors, mentors, and colleagues to see what improvements you can make to your designs. Reflect on your work periodically so that you can learn from your experiences and think about how to improve for the next time.

Reference

RoyClem. (2005, Mar 21). Identifying object-oriented classes. Retrieved from https://www.codeproject.com/Articles/9900/Identifying-Object-Oriented-Classes.

Required Resources

Reading: The Object Model (https://go.oreilly.com/SNHU/library/view/object-oriented-analysis-and/9780201895513/ch02.html)
This optional reading from the Shapiro Library provides an extended look into object models including the evolution, foundations, and elements of the model. The author provides an extensive look at the conceptual framework for object-oriented methods the object model encompasses including several attributes of a properly structured object model.

Textbook: *Modern Systems Analysis and Design*, Chapter 8
Chapter 8 provides you with the conceptual understanding you need to handle the data used in a system, and it explores the entity-relationship diagram, a commonly used diagram in system analysis and design.

NOTE: The last few pages of each chapter contain a list of key terms, review questions, and exercises. Review these pages to check how well you understand the reading.

Reading: How to Use a Gantt Chart (https://www.lucidchart.com/blog/how-to-use-a-gantt-chart)
This reading describes how to make and use a Gantt chart, a visual way of representing the different tasks for a project. You will learn the different steps that go into making a good schedule, including how to determine, organize, and assign tasks. You will also learn how to create a Gantt chart with Lucidchart, a commonly used computer-aided software engineering, or CASE, tool.

Reading: Lucidchart Tutorial (/d2l/lor/viewer/viewFile.d2lfile/1918255/24209,-1/)
This tutorial will show you how to create a Lucidchart account as well as how to create different types of documents and diagrams. Use this reading to help you create a Gantt chart that lays out a schedule of the different tasks needed for your project.

Reading: Object-Oriented Analysis and Design (OOAD) (https://www.geeksforgeeks.org/object-oriented-analysis-and-design/)
This article provides a high-level overview of object-oriented analysis and design. It covers the benefits and challenges, and it provides examples of real-world applications of OOAD.

Reading: UML Class Diagram Tutorial (https://www.lucidchart.com/pages/uml-class-diagram)
This tutorial covers how to read and construct a general UML diagram. Focus on how to read a UML and do not worry about all the nuances yet. This tutorial will also be in Module Six, when you will need to work with UML in more depth.

Reading: UML Use Case Diagram Tutorial (https://www.lucidchart.com/pages/uml-use-case-diagram)
This tutorial introduces the UML use case diagram, its purpose, and how to use it. Focus on how to read a UML and do not worry about all the nuances yet. This tutorial will also be in Module Six, when you will need to work with UML use case diagrams in more depth.

Additional Support (Optional)

Video: UML Use Case Diagram Tutorial | Definition, Symbols, and More (https://www.youtube.com/watch?v=uNtoyXJ1VDc) (8:54)
This video provides a more detailed explanation of UML use case diagrams, how to read them, and how to create them.