

W02 Team Activity: Journal Design

Overview

This activity will walk you through the process of creating a design for your developer project. It is very important that you understand the decisions and tradeoffs that are made along the way so that later on, you can produce a design on your own with less guidance.

Team Activity Instructions

This activity is designed to be completed as a team, working together synchronously on a video call.

For this and future team activities, you will choose someone to be the lead student. The lead student should help guide the discussion and ask the questions. Then, you will choose a different person to be the lead student for the next meeting.

This activity contains two kinds of questions. The first are questions that help you build up the design. To make sure that you have a solid design to use for your program, "Guidance from a Mentor" will be provided to help.

Guidance from a Mentor

This activity is designed to help you walk through the design process as a team. Because decisions in the first steps determine many steps that follow, we want to make sure you do not start down a path that will lead to problems later. Similarly, when you start writing the code for your project, it is important that you have a solid design to work from, and one that will reinforce the concepts of the course.

If we were in a conference room together at a company, a mentor could quietly observe the discussion and only interject if the conversation started to lead in a

bad direction.

In the case of this activity, because the instructor cannot be in each team meeting, we have placed expandable "Guidance from a Mentor" sections throughout the design process. You should **not** expand and look at these sections until you have done your best as a team to answer the questions. Then, **after** you have answered the questions, you should check these Guidance sections to make sure you are headed in a good direction.

Keep in mind that there are often many good designs for a given problem, but for this project, you should use the design created here to make sure that you have a good approach that will also reinforce the principles of the lesson.

Make sure to expand and read each Guidance from a Mentor section as you move through the activity.

In addition to the design questions which contain Guidance from a Mentor, you will also see questions to help you evaluate the design, as show below:

Evaluate the Design

- This is an example of how questions will appear.

These questions will give you a chance to examine the reason for making certain decisions, and to compare and contrast various design possibilities. You should discuss these questions as a group, just as with the other questions, but they will not contain solutions in a "Guidance from a Mentor" box. Instead, at the end of the activity (or during the activity if you would like) you will submit a quiz with your answers to the "Evaluate the Design" questions. Keep in mind that you can submit this quiz multiple times if you would like.

Agenda

Use the following as an agenda for your team meeting. Whoever is assigned to be the lead student for this gathering should help guide the group through these steps and ask the questions listed here.

Step 1: Before the meeting: Verify the time, location, and lead student

This could be as simple as posting a message to your MS Teams channel that says something like, *"Hi guys, are we still planning to meet tomorrow at 7pm Mountain Time? Let's use the MS Teams video feature again."* Or, if someone else has already posted a message like this, it could be as simple as "liking" their message.

Make sure to identify who will be the lead student for this week. For example, *"Emily, are you still good to be the lead student for this week?"*

Step 2: Begin with Prayer

Step 3: Discuss the Preparation Learning Activity

Take a minute to talk about the learning activity from this week. Talk through any difficulties that people had understanding the material or completing the activity.

- What part of the learning activity was the hardest for you?

Step 4: Review the Program Specification

Refer to the [Journal program specification](#). As a team, review the program requirements and how it is supposed to work.

1. What does the program do?
2. What user inputs does it have?
3. What output does it produce?
4. How does the program end?

► Guidance from a Mentor

Step 5: Determine the classes

The first step in designing a program like this is to think about the classes you will need. When thinking about classes, it is often helpful to consider the strong nouns in the program description.

1. What are good candidates for classes in this program?
2. What are the primary responsibilities of each class?

► Guidance from a Mentor

Evaluate the Design

- If you followed the design from the "Guidance from a Mentor" section, and then in the future, you changed your program so that the prompts were retrieved directly from a Web database, how many classes would have to be updated?

Step 6: Define class behaviors

Now that you have decided on the classes you will need and their responsibilities, the next step is to define the behaviors of these classes. These will become the methods of each class.

Go through each of your classes and ask:

1. What are the behaviors this class will have in order to fulfill its responsibilities? (In other words, what things should this class *do*?)

► Guidance from a Mentor

Evaluate the Design

- What are the potential benefits of having a **Display** method in the **Entry** class rather than allowing the **Journal**'s display method to display an entry's date and text directly?

Step 7: Define class attributes

Now that you have defined the classes, their responsibilities, and their behaviors, the next step is to determine what attributes the class should have, or what variables it needs to store.

Go through each of your classes and ask:

1. What attributes does this class need to fulfill its behaviors? (In other words, what member variables should this class *store*?)
2. What are the data types of these member variables?

► Guidance from a Mentor

Step 8: Review the Design

Take a minute to review your final design.

1. Are there any classes, methods, or variables, that you do not understand?

► Guidance from a Mentor

Evaluate the Design

- Using this design, when you want to add a new entry to the journal, you will use code such as `theJournal.AddEntry(anEntry)` ; instead of using the `_entries` variable and its add method like this `theJournal._entries.Add(anEntry)` ;. What is a benefit of our design approach (the `AddEntry` method), instead of accessing the variable directly?

Step 9: Conclude the Meeting

At this point, you have the design of the classes you will need for this project. If your design is not "perfect," or it needs to change a little as you begin working on the project, that is just fine! As you learn more details, you will naturally need to adjust your planning. This is why the principles of programming with class are so valuable, because they allow your program to easily change.

At the end of your meeting:

- Determine who will be the lead student for the next meeting.

Step 10: After the Meeting: Start the code

Now that you have a design for your classes in mind. The next step is to start the code of the program.

You begin programs with classes by creating "stubs" for everything in your design, or in other words, an empty skeleton that contains all of the classes from your design with all of the member variables and methods. At this point, the methods can be (mostly) empty. You will fill them in later as you begin the program.

Avoid Build Errors

One important factor as you "stub out" your program is that you want to make sure that it can build (we often say "compiled") without errors. This is why some of your methods cannot be completely empty.

If the function has a **void** return type, meaning it does not return anything, it can be left completely empty.

However, if the function has a return type, you will need to return *something*, or else you will have errors when you try to run it. For example, if the return type is **string** then you might include **return ""**; as a single line of the function so that it will not have errors.

After the team activity, each person needs to individually do the the following:

1. Open the project in VS Code. Create new files that contain the "stubs" or empty code for all the classes, member variables, and functions in your design.
 - At this point the body of the methods can be empty, except for the necessary return statements.
 - Each class should be in its own file and the name of the file should match the class name (for example, **Journal.cs**).
2. Make sure that your program can build without errors.
3. Commit and push your code to your GitHub repository.

► Need help getting started?

Submission

After completing this activity, return to Canvas to submit two quizzes associated with this activity:

1. **W02 Team Activity: Journal Design:** In this quiz, you will respond to the "Evaluate the Design" questions from the activity above. You may take this quiz 3 times.
2. **W02 Team Activity: Participation Report:** In this quiz, you will report on your participation with your team. The lowest score from this category will be dropped at the end of the course. So if you need to miss one meeting for any reason, it will not be a problem, but if it becomes a pattern, you will not earn full points for your teamwork.