

CSE 210

Polymorphism - Design Activity

Overview

Meet with your team and prepare a design for this week's programming assignment.

Regarding "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 don't start down a path that will lead to problems later.

If we were in a conference room together at a company, the mentor would 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 "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.

Important: There is more than one good way to design a project. We have been hesitant to include any guidance like this, because you shouldn't think this is the only way to solve the problem or the way you *must* do the project. If your design is following a similar approach, but uses slightly different words for the classes/variables/methods, that is great! You don't have to conform exactly to the suggestions here.

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.

1. What part of the learning activity was the hardest for you?

Step 4: Review the Program Specification

Refer to the [Eternal Quest 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

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 methods for the 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

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 variables should this class *store*?)
2. What are the data types of these member variables?
3. What constructors should each class have?

► Guidance from a Mentor

Step 8: Define Constructors

Now that you have defined the classes, including their methods and attributes, the next step is to think about the constructors you will need to create new instances of these classes. Remember that you can create multiple constructors with different parameters to make it easy to work with your classes.

Remember, that constructors help set up the initial state of the object, so you should consider what data is necessary for that initial state.

1. What constructors should each class have?

- In other words, what parameters should you pass in when creating an object of that type.

2. What other work needs to be done to set up these objects?

- For example, does the constructor need to run code to perform set up tasks, like creating lists, iterating through variables, etc.

► Guidance from a Mentor

Step 9: Conclude

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 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:

1. Make sure that each person in the meeting gets a copy of any resources you have developed as a team.
2. Determine who will be the lead student for the next meeting.

Submission

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

1. Complete a design based on the team discussion.

- Include a class diagram for each class you will need.
- Include some kind of description or chart showing how the program will run or how these methods relate to one another.
- You are welcome to use any drawings, diagrams, or other artifacts that you created as a team.
- You might also need to create a few more on your own, based on the discussion you had with your team.

2. Each person should individually submit their design to I-Learn.

Copyright © Brigham Young University-Idaho | All rights reserved