

## Lesson 6.07 — Programming Project

### Overview

#### Objectives — *Students will be able to...*

- **Write** complex code that uses polymorphism and inheritance.
- **Implement** the applied design stages as they create a program for an end-user.

#### Assessments — *Students will...*

- **Submit** multiple programs electronically
- **Take** two quizzes from the Barron's review book

#### Homework —

- A list of homework assignments is listed in the Pacing Guide

### Materials & Prep

- **Projector and computer**
- **Whiteboard and markers**
- **Classroom copies** of textbook (for Day 4)
- **A sample grocery receipt** (for Day 4)

#### Pacing Guide: Day 1

Section	Total Time
<b>Programming Project I</b>	Full class
<b>Grade student notebooks</b>	During class
<b>Homework:</b> <i>Outline Chapter 9</i>	TONIGHT

#### Pacing Guide: Day 2

Section	Total Time
<b>Programming Project I</b> ( <i>Project 9.1 from the textbook</i> )	Full class
<b>Check student outlines for completeness</b>	During class
<b>Homework:</b> <i>Read and outline Chapter 4 in Barron's review book</i>	TONIGHT

#### Pacing Guide: Day 3

Section	Total Time
<b>Programming Project II</b> ( <i>Project 9.3 from the textbook</i> )	Full class
<b>Homework:</b> <i>Take the Chapter 4 quiz. Grade your answers.</i>	TONIGHT

#### Pacing Guide: Day 4

Section	Total Time
<b>Programming Project III</b> ( <i>Exercise 9.3 from the textbook</i> )	Full class
<b>Homework:</b> <i>Read and outline Chapter 3 in Barron's review book.</i>	TONIGHT

## Pacing Guide: Day 5

Section	Total Time
<b>Take Chapter 3 quiz IN CLASS</b> <i>Students grade their own answers and submit review books for a grade.</i>	Full class
<b>Check Barron's review books for highlighting, note-taking, and quiz completion/grading</b>	During class
<b>Homework:</b> <i>Review Chapter 9 and submit 5 questions for review.</i>	TONIGHT

## Procedure

Students should complete the programming projects on their own. Read through the instructions with the class, and help students pace themselves by setting progress goals for each day.

Before letting students begin the projects, ask students what resources are available to them so they can help themselves before calling you over. (Notebooks, textbook, class handouts, student work and posters hanging around the room, online resources.

## About Barron's

- Barron's is an AP CS A review book that some schools provide students. If your school doesn't provide Barron's there are many alternative homework assignments that can be found at [codingbat.com/java](http://codingbat.com/java)
- Alternatively, you can save time spent on the lab by checking activities as homework.
- If you've chosen to introduce TextExcel earlier, you can assign portions of TextExcel as homework here to save time later in the course. TextExcel like FracCalc takes a lot of in and out of class time to complete, and time saved here saves time for AP test review.

## Programming Projects

On the overhead or shared electronically, provide students with the following programming prompts:

### Programming Project I Write an inheritance hierarchy of three-dimensional shapes:

1. Make a top-level shape class that has methods for getting information such as the volume and surface area of a three-dimensional shape.
2. Then make subclasses for various shapes such as cubes, rectangular prisms, spheres, triangular prisms, cones, and cylinders.
3. Place common behavior in superclasses whenever possible.
4. Add methods to the subclasses to represent the unique behavior of each three-dimensional shape, such as a method to get a sphere's radius.

**Programming Project II - Applied Design Stages and Creating for End-Users** In this project you will interview an end-user in order to determine their interests. You will then write an inheritance hierarchy that stores data related to the end-user's interests.

1. Select an end-user (this can be a classmate, friend, teacher, relative, etc).
2. Create a series of questions that will allow you to understand their interests. Have the user respond to the questions and record their responses.
3. With the end-user, design a common superclass that will store information related to their interests (ie: if the end-user is interested in sports then the superclass could include information common to any player regardless of sport, such as name, number, and salary).

4. With the end-user, design a subclasses for specific elements, characters or players that fall under the superclass structure (ie: players of your favorite sports, such as basketball, soccer or tennis could have specific attributes under the athlete superclass). Be sure to place common behavior in superclasses whenever possible. Place sport-specific information and behavior (such as kicking, vertical jump height, or speed) into subclasses whenever possible.
5. Add methods to the subclasses to represent the unique behavior of each athlete.
6. Program the superclass and subclasses. Your program should provide the user with a menu that allows them to display different data based on user's input.
7. Test the program with the end-user. Record their comments, suggestions and feedback and make any required changes.
8. Test the program with your classmates and friends. Record their comments, suggestions and feedback and make any required changes.
9. Submit to your teacher the interview questions and responses, the feedback from end-user testing, the feedback from classroom testing and the finished program.

## Grading Scheme/Rubric

Component	Value
Implementation	
Superclass is implemented effectively	4 points
Subclasses are implemented effectively	4 points
Superclass and Subclass are designed effectively considering user interests	2 points
User and class testing is effective	2 points
Menu and general flow of program is effective and user-friendly	2 points
TOTAL	14 points
Applied Design Steps	
User interview questions and responses are thorough and informative	1 point
Ideating components are complete and thorough	1 point
Ideating/Prototyping are components complete and thorough	1 point
Testing components with user are complete and thorough	1 point
Testing components with class are complete and thorough	1 point
Total	5 points
Grand Total	19 points

**Programming Project III** Have students submit answers to Exercise 9.3 in their textbooks. This assignment will be submitted on paper. As written, students are not given some code that implements GroceryBill, Employee, and Item. If you wish to make this an electronic assignment, you should consider providing students with some starter code.

## Accommodation and Differentiation

In some classrooms, you may want to provide the mathematical formulas for finding the volume of spheres, triangular prisms, cones, and cylinders so students can focus on coding rather than finding and translating the math. You can also introduce and reinforce useful vocabulary by providing physical examples of these shapes. If you are having trouble finding prisms, cones, and cylinders, you can make these items out of paper by printing out templates and following the instructions at this website: (<http://tinyurl.com/asz6e69>)

To help students process Programming Project III, you may want to read through exercise 9.3 in the textbook with the class, breaking down the problem into its parts as a whole group. In ELL classrooms, you should bring in physical samples of grocery receipts and pass them around or view them on the projector. Have students point out the different parts of the receipt as they relate to the assignment.

Remind students to always draw a structure diagram, sketch out their plan using pseudocode, and include comments to remind themselves (and you!) of what each section of code is for.

## **Forum discussion**

Lesson 6.07 Programming Project (TEALS Discourse account required)