Programming Projects

PP 9.1 Write a class called MonetaryCoin that is derived from the Coin class presented in Chapter 5. Store an integer in the MonetaryCoin that represents its value and add a method that returns its value. Create a main driver class to instantiate and compute the sum of several MonetaryCoin objects. Demonstrate that a monetary coin inherits its parent's ability to be flipped.

PP 9.2 Design and implement a set of classes that define the employees of a hospital: doctor, nurse, administrator, surgeon, receptionist, janitor, and so on. Include methods in each class that are named according to the services provided by that person and that print an appropriate message. Create a main driver class to instantiate and exercise several of the classes.

PP 9.3 Design and implement a set of classes that define various types of reading material: books, novels, magazines, technical journals, textbooks, and so on. Include data values that describe various attributes of the material, such as the number of pages and the names of the primary characters. Include methods that are named appropriately for each class and that print an appropriate message. Create a main driver class to instantiate and exercise several of the classes.

PP 9.4 Design and implement a set of classes that keeps track of various sports statistics. Have each low-level class represent a specific sport. Tailor the services of the classes to the sport in question, and move common attributes to the higher-level classes as appropriate. Create a main driver class to instantiate and exercise several of the classes.

PP 9.5 Design and implement a set of classes that keeps track of demographic information about a set of people, such as age, nationality, occupation, income, and so on. Design each class to focus on a particular aspect of data collection. Create a main driver class to instantiate and exercise several of the classes.

PP 9.6 Design and implement a set of classes that define a series of three-dimensional geometric shapes. For each, store fundamental data about their size and provide methods to access and modify this data. In addition, provide appropriate methods to compute each shape's circumference, area, and volume. In your design, consider how shapes are related and thus where inheritance can
be implemented. Create a main driver class to instantiate several shapes of differing types and exercise the behavior you provided.

**PP 9.7** Design and implement a set of classes that define various types of electronics equipment (computers, cell phones, pagers, digital cameras, etc.). Include data values that describe various attributes of the electronics, such as the weight, cost, power usage, and the names of the manufacturers. Include methods that are named appropriately for each class and that print an appropriate message. Create a main driver class to instantiate and exercise several of the classes.

**PP 9.8** Design and implement a set of classes that define various courses in your curriculum. Include information about each course such as the title, number, description, and department that teaches the course. Consider the categories of classes that constitutes your curriculum when designing your inheritance structure. Create a main driver class to instantiate and exercise several of the classes.

**PP 9.9** Modify the Rebound program from this chapter such that when the mouse button is clicked the animation stops, and when it is clicked again the animation resumes.

**PP 9.10** Write a program that displays an animation of a car (side view) moving across the screen from left to right. Create a Car class that represents the car (or use one that was created for a programming project in Chapter 8).

**PP 9.11** Write a program that displays an animation of a horizontal line segment moving across the screen, eventually passing across a vertical line. As the vertical line is passed, the horizontal line should change color. The change of color should occur while the horizontal line crosses the vertical one; therefore, while crossing, the horizontal line will be two different colors.

**PP 9.12** Write a program that plays a game called Catch-the-Creature. Use an image to represent the creature. Have the creature appear at a random location for a random duration, then disappear and reappear somewhere else. The goal is to “catch” the creature by pressing the mouse button while the mouse pointer is on the creature image. Create a separate class to represent the creature, and include in it a method that determines if the location of the mouse click corresponds to the current location of the creature. Display a count of the number of times the creature is caught as well as the number of misses.