

4. In pseudocode, write the first line of the definition for a `Poodle` class. The class should extend the `Dog` class.

5. Look at the following pseudocode class definitions:

```
Class Plant
  Public Module message()
    Display "I'm a plant."
  End Module
End Class
```

```
Class Tree Extends Plant
  Public Module message()
    Display "I'm a tree."
  End Module
End Class
```

Given these class definitions, what will the following pseudocode display?

```
Declare Plant p
Set p = New Tree()
Call p.message()
```

Programming Exercises

1. Pet Class

Design a class named `Pet`, which should have the following fields:

- **name:** The name field holds the name of a pet.
- **type:** The type field holds the type of animal that a pet is. Example values are "Dog", "Cat", and "Bird".
- **age:** The age field holds the pet's age.

The `Pet` class should also have the following methods:

- **setName:** The `setName` method stores a value in the name field.
- **setType:** The `setType` method stores a value in the type field.
- **setAge:** The `setAge` method stores a value in the age field.
- **getName:** The `getName` method returns the value of the name field.
- **getType:** The `getType` method returns the value of the type field.
- **getAge:** The `getAge` method returns the value of the age field.

Once you have designed the class, design a program that creates an object of the class and prompts the user to enter the name, type, and age of his or her pet. This data should be stored in the object. Use the object's accessor methods to retrieve the pet's name, type, and age and display this data on the screen.

2. Car Class

Design a class named `Car` that has the following fields:

- **yearModel:** The `yearModel` field is an `Integer` that holds the car's year model.
- **make:** The `make` field references a `String` that holds the make of the car.
- **speed:** The `speed` field is an `Integer` that holds the car's current speed.

In addition, the class should have the following constructor and other methods:

- **Constructor:** The constructor should accept the car's year model and make as arguments. These values should be assigned to the object's `yearModel` and `make` fields. The constructor should also assign 0 to the `speed` field.



- **Accessors:** Design appropriate accessor methods to get the values stored in an object's `yearModel`, `make`, and `speed` fields.
- **accelerate:** The `accelerate` method should add 5 to the `speed` field each time it is called.
- **brake:** The `brake` method should subtract 5 from the `speed` field each time it is called.

Next, design a program that creates a `Car` object, and then calls the `accelerate` method five times. After each call to the `accelerate` method, get the current speed of the car and display it. Then call the `brake` method five times. After each call to the `brake` method, get the current speed of the car and display it.

3. Personal Information Class

Design a class that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator methods. Also, design a program that creates three instances of the class. One instance should hold your information, and the other two should hold your friends' or family members' information.

4. Employee and ProductionWorker Classes

Design an `Employee` class that has fields for the following pieces of information:

- Employee name
- Employee number

Next, design a class named `ProductionWorker` that extends the `Employee` class. The `ProductionWorker` class should have fields to hold the following information:

- Shift number (an integer, such as 1, 2, or 3)
- Hourly pay rate

The workday is divided into two shifts: day and night. The shift field will hold an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Design the appropriate accessor and mutator methods for each class.

Once you have designed the classes, design a program that creates an object of the `ProductionWorker` class and prompts the user to enter data for each of the object's fields. Store the data in the object and then use the object's accessor methods to retrieve it and display it on the screen.

5. Essay Class

Design an `Essay` class that extends the `GradedActivity` class presented in this chapter. The `Essay` class should determine the grade a student receives for an essay. The student's essay score can be up to 100 and is determined in the following manner:

- Grammar: up to 30 points
- Spelling: up to 20 points
- Correct length: up to 20 points
- Content: up to 30 points

... prompts the user to enter the ... and con-