Lesson 7

In-class Exercise

Write a class named Car that has the following member variables:

- year. An int that holds the car's model year.
- make. A string object that holds the make of the car.
- speed. An int that holds the car's current speed.

In addition, the class should have the following member functions.

- **Constructor. The constructor should accept the car's year and make as arguments and assign these values to the object's year and make member variables. The constructor should initialize the speed member variable to 0.**
- Accessors. Appropriate accessor functions should be created to allow values to be retrieved from an object's year, make, and speed member variables.
- Mutators. Appropriate mutator functions to assign values to the data members
- accelerate. The accelerate function should add 5 to the speed member variable each time it is called.
- brake. The brake function should subtract 5 from the speed member variable each time it is called.

Demonstrate the class in a program that creates a Car object and then calls the accelerate function five times. After each call to the accelerate function, get the current speed of the car and display it. Then, call the brake function five times. After each call to the brake function, get the current speed of the car and display it.

2. **Date**
Design a class called Date that has integer data members to store month, day, and year. The class should have a three-parameter default constructor that allows the date to be set at the time a new Date object is created. If the user creates a Date object without passing any arguments, or if any of the values passed are invalid, the default values of 1, 1, 2001

(i.e., January 1, 2001) should be used. The class should have member functions to print the date in the following formats:

3/15/20
March 15, 2020
15 March 2020

Demonstrate the class by writing a program that uses it. Be sure your program only accepts reasonable values for month and day. The month should be between 1 and 12. The day should be between 1 and the number of days in the selected month.

3. Report Heading

Design a class called Heading that has data members to hold the company name and the report name. A two-parameter default constructor should allow these to be specified at the time a new Heading object is created. If the user creates a Heading object without passing any arguments, “ABC Industries” should be used as a default value for the company name and “Report” should be used as a default for the report name. The class should have member functions to print a heading in either one-line format, as shown here:

Pet Pals Payroll Report

or in four-line “boxed” format, as shown here:

```
**************************************************
              Pet Pals
            Payroll Report
**************************************************
```

Try to figure out a way to center the headings on the screen, based on their lengths. Demonstrate the class by writing a simple program that uses it.

4. In a population, the birth rate and death rate are calculated as follows:

$$\text{Birth Rate} = \text{Number of Births} \div \text{Population}$$
$$\text{Death Rate} = \text{Number of Deaths} \div \text{Population}$$

For example, in a population of 100,000 that has 5,000 births and 2,000 deaths per year,

$$\text{Birth Rate} = 5{,}000 \div 100{,}000 = 0.05$$
$$\text{Death Rate} = 2{,}000 \div 100{,}000 = 0.02$$

Design a Population class that stores a current population, annual number of births, and annual number of deaths for some geographic area. The class should allow these three values to be set in either of two ways: by passing arguments to a three-parameter constructor when a new Population object is created or by calling the setPopulation, setBirths, and setDeaths class member functions. In either case, if a population figure less than 2 is passed to the class, use a default value of 2. If a birth or death figure less than 0 is passed in, use a default value of 0. The class should also have getBirthRate and getDeathRate functions that compute and return the birth and death rates. Write a short program that uses the Population class and illustrates its capabilities.

5.
Define the following classes:

Movie Data

MovieData to store the following information about a movie: title, director, release year, and running time.
Add appropriate constructors, mutators, accessors

Date
Month, day, year

Add appropriate constructors, mutators, accessors

Time
Hours, minutes, seconds
Add appropriate constructors, mutators, accessors

Ticket
Movie, Date, Time, Cost, seatNumber
Add appropriate constructors, mutators, accessors

WAP to display the Ticket.