

Object Oriented Programming

Assignment No. 1

QUESTION # 01

Write a program that uses a class named **MovieData** to store the following information about a movie:

Title
Director
Year Released
Running time (in minutes)

Include a function that allows all four of these member data values to be specified. The program should create two **MovieData** variables and pass each one in turn to a function that displays the information about the movie in a clearly formatted manner.

Modify the Movie Data program written in Question#01 to include two additional members that hold the movie's production costs and first-year revenues. The function should be modified so that all six member values can be specified when a **MovieData** variable is created. Modify the function that displays the movie data to display the title, director, release year, running time, and first year's profit or loss.

QUESTION # 02

Write a program that uses a class named **CorpData** to store the following information on a company division:

Division name (such as East, West, North, or South)
First quarter sales
Second quarter sales
Third quarter sales
Fourth quarter sales
Total annual sales
Average quarterly sales.

Include a constructor that allows the division name and four quarterly sales amounts to be specified at the time a **CorpData** variable is created.

The program should create four variables of this class, each representing one of the following corporate divisions: East, West, North, and South. Each variable should be passed in turn to a function that calculates and stores the total sales and average quarterly sales for that division. Once this has been done for each division, each variable should be passed in turn to a function that displays the division name, total sales, and quarterly average.

QUESTION # 03

A student has established the following monthly budget:

Housing	500.00
Utilities	150.00
Household expenses	65.00
Transportation	50.00
Food	250.00
Medical	30.00
Insurance	100.00
Entertainment	150.00
Clothing	75.00
Miscellaneous	50.00

Write a program that declares a **MonthlyBudget** class with member variables to hold each of these expense categories. The program should create two **MonthlyBudget** class variables. The first will hold the budget figures

given above. The second will be passed to a function that has the user enter the amounts actually spent in each budget category during the past month. The program should then pass both class variables to a function that displays a report indicating the amount over or under budget the student spent in each category, as well as the amount over or under for the entire monthly budget.

QUESTION # 04

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:

```
3/15/10
March 15, 2010
15 March 2010
```

Demonstrate the class by writing a program that uses it.

Input Validation: Only accept values between 1 and 12 for the month, between 1 and 31 for the day, and between 1950 and 2020 for the year.

QUESTION # 05

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

- **year.** An `int` that holds the car's model year.
- **make.** A `string` 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.
- **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.

QUESTION # 06

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

Birth Rate = Number of Births ÷ Population

Death Rate = Number of Deaths ÷ Population

For example, in a population of 100,000 that has 8,000 births and 6,000 deaths per year,

Birth Rate = 8,000 ÷ 100,000 = 0.08

Death Rate = 6,000 ÷ 100,000 = 0.06

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. 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.

QUESTION # 07

Assuming that a year has 365 days, write a class named **DayOfYear** that takes an integer representing a day of the year and translates it to a string consisting of the month followed by day of the month. For example,

Day 2 would be January 2

Day 32 would be February 1

Day 365 would be December 31.

The constructor for the class should take as parameter an integer representing the day of the year, and the class should have a member function **print()** that prints the day in the month-day format. The class should have an integer member variable to represent the day, and should have member variables of type string to assist in the translation from the integer format to the month-day format.

Test your class by inputting various integers representing days and printing out their representation in the month-day format.
