

Lab Exercise on Classes and Objects

Question 1

Create a class named `Billing` that includes three overloaded `computeBill()` methods for a photo book store.

- When `computeBill()` receives a single parameter, it represents the price of one photo book ordered. Add 8% tax, and return the total due.
- When `computeBill()` receives two parameters, they represent the price of a photo book and the quantity ordered. Multiply the two values, add 8% tax, and return the total due.
- When `computeBill()` receives three parameters, they represent the price of a photo book, the quantity ordered, and a coupon value. Multiply the quantity and price, reduce the result by the coupon value, and then add 8% tax and return the total due.

Write a `main()` method that tests all three overloaded methods. Save the application as `Billing.java`.

Question 2

Create a class named **Patient** that includes an ID number, age, and `BloodData`. Provide a default constructor that sets the ID number to 0, the age to 0, and the `BloodData` values to 0 and 1. Create an overloaded constructor that provides values for each field. Also provide get methods for each field. Save the file as `Patient.java`. Create an application that demonstrates that each method works correctly, and save it as `TestPatient.java`.

Question 3

People who deal with historical dates use a number called the *Julian day* to calculate the number of days between two events. The Julian day is the number of days that have elapsed since January 1, 4713 B.C. For example, the Julian day for October 16, 1956, is 2435763. There are formulas for computing the Julian day from a given date, and vice versa. One very simple formula computes the day of the week from a given Julian day:

Day of the week = (Julian day + 1) % 7

where % is the Java modulus operator. This formula gives a result of 0 for Sunday, 1 for Monday, and so on, up to 6 for Saturday. For Julian day 2435763, the result is 2 (Tuesday). Your job is to write a Java application that requests and inputs a Julian day, computes the day of the week using the formula, and then displays the name of the day that corresponds to that number. Your output might look like this:

Enter a Julian day number and press Enter.

2451545

Julian day number 2451545 is a Saturday.

Enter a Julian day number and press Enter.

2451547

Julian day number 2451547 is a Monday.

Question 4

Create a class named Shirt with data fields for collar size and sleeve length. Include a constructor that takes arguments for each field. Also include a String class variable named material and initialize it to "cotton". Write a program named TestShirt to instantiate three Shirt objects with different collar sizes and sleeve lengths and then display all the data, including material, for each shirt.

Question 5

Create a class named TaxPayer. Data fields for TaxPayer include Social Security Number (SSN- use an int for the type), and yearly gross income. Methods include a constructor that requires values for both data fields, and two get methods that return each of the data field. The SSN starts from 100000 and increases by 1 for the next TaxPayer. Use static block static variable wherever is applicable.

Write a program named UseTaxPayer that declares an array of 10 TaxPayer objects. Set SSN and gross income. Display the 10 TaxPayer objects.