

Assignment (Part-2)
Introduction to Classes, Objects, Methods, String

14. **(Modified Account Class)** Modify class `Account` (Textbook Fig. 3.8) to provide a method called `withdraw` that withdraws money from an `Account`. Ensure that the withdrawal amount does not exceed the `Account`'s balance. If it does, the balance should be left unchanged and the method should print a message indicating "Withdrawal amount exceeded account balance". Modify class `AccountTest` (Textbook Fig. 3.9) to test method `withdraw`.
15. **(Invoice Class)** Create a class called `Invoice` that a hardware store might use to represent an invoice for an item sold at the store. An `Invoice` should include four pieces of information as instance variables—a part number (type `String`), a part description (type `String`), a quantity of the item being purchased (type `int`) and a price per item (type `double`). Your class should have a constructor that initializes the four instance variables. Provide a `set` and a `get` method for each instance variable. In addition, provide a method named `getInvoiceAmount` that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a `double` value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test app named `InvoiceTest` that demonstrates class `Invoice`'s capabilities.
16. **(Employee Class)** Create a class called `Employee` that includes three instance variables—a first name (type `String`), a last name (type `String`) and a monthly salary (type `double`). Provide a constructor that initializes the three instance variables. Provide a `set` and a `get` method for each instance variable. If the monthly salary is not positive, do not set its value. Write a test app named `EmployeeTest` that demonstrates class `Employee`'s capabilities. Create two `Employee` objects and display each object's yearly salary. Then give each `Employee` a 10% raise and display each `Employee`'s yearly salary again.
17. **(Date Class)** Create a class called `Date` that includes three instance variables—a month (type `int`), a day (type `int`) and a year (type `int`). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a `set` and a `get` method for each instance variable. Provide a method `displayDate` that displays the month, day and year separated by forward slashes (/). Write a test app named `DateTest` that demonstrates class `Date`'s capabilities.
18. **(Target-Heart-Rate Calculator)** While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. Formula for calculating maximum heart rate is beats per minute is 220 minus your age in years. Your target heart rate is a range that's 50–85% of your maximum heart rate. Create a class called `HeartRates`. The class attributes should include the person's first name, last name and date of birth (consisting of separate attributes for the month, day and year of birth). Your class should have a constructor that receives this data as parameters. For each attribute provide `set` and `get` methods. The class also should include a method that calculates and returns the person's age (in years), a method that calculates and returns the person's maximum heart rate and a method that calculates and returns the person's target heart rate. Write a Java application that prompts for the person's information, instantiates an object of class `HeartRates` and prints the information from that object—including the person's first name, last name and date of birth—then calculates and prints the person's age in (years), maximum heart rate and target-heart-rate range.
19. **(Computerization of Health Records)** A health-care issue that has been in the news lately is the computerization of health records. This possibility is being approached cautiously because of

sensitive privacy and security concerns, among others. Computerizing health records could make it easier for patients to share their health profiles and histories among their various health-care professionals. This could improve the quality of health care, help avoid drug conflicts and erroneous drug prescriptions, reduce costs and, in emergencies, could save lives. You'll design a "starter" `HealthProfile` class for a person. The class attributes should include the person's first name, last name, gender, date of birth (consisting of separate attributes for the month, day and year of birth), height (in inches) and weight (in pounds). Your class should have a constructor that receives this data. For each attribute, provide *set* and *get* methods. The class also should include methods that calculate and return the user's age in years, maximum heart rate and target-heart-rate range (refer to Q18), and body mass index (BMI; refer to Assignment Part-1, Q12). Write a Java app that prompts for the person's information, instantiates an object of class `HealthProfile` for that person and prints the information from that object—including the person's first name, last name, gender, date of birth, height and weight—then calculates and prints the person's age in years, BMI, maximum heart rate and target-heart-rate range. It should also display the BMI values chart (refer to Assignment Part-1 Q12).