

ECET11 – Object Oriented Programming LAB activity 1

Objective: Intro to Classes

Source: Deitel & Deitel Edition 5 – Chapter 3 Exercises

1. (Account Class) Create a class called Account that a bank might use to represent customers' bank accounts. Your class should include one data member of type int to represent the account balance. [Note: In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75) called floating-point values to represent dollar amounts.] Your class should provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure that it is greater than or equal to 0. If not, the balance should be set to 0 and the constructor should display an error message, indicating that the initial balance was invalid. The class should provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and should ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debit amount exceeded account balance." Member function getBalance should return the current balance. Create a program that creates two Account objects and tests the member functions of class Account.
2. (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 data members 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 int). [Note: In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75) called floating-point values to represent dollar amounts.] Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int 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. Write a test program that demonstrates class Invoice's capabilities.

3. (Employee Class) Create a class called Employee that includes three pieces of information as data members a first name (type string), a last name (type string) and a monthly salary (type int). [Note: In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75) called floating-point values to represent dollar amounts.] Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.
4. (Date Class) Create a class called Date that includes three pieces of information as data members a month (type int), a day (type int) and a year (type int). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1 to 12; if it is not, set the month to 1. Provide a set and a get function for each data member. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class Date's capabilities.