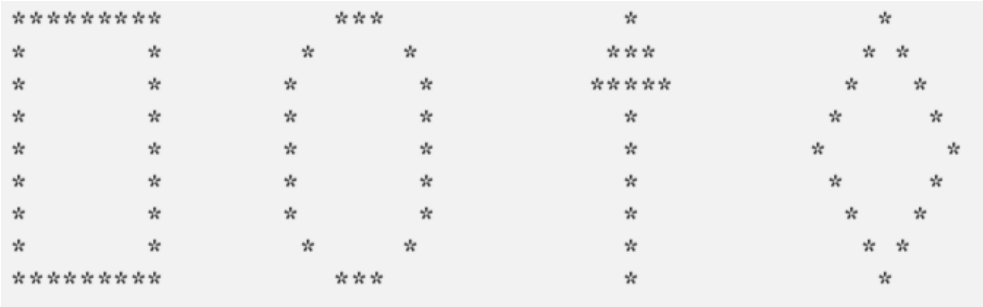
OOP With C++ (Assignment -2)
CSE, 3rd Semester
Deadline August 09, 2020

Prepared By: Deepak Uniyal (Assistant Professor CSE, GEU)

Note -

- Create your GitHub profile as taught in lectures and then push all your programs to folders named according to assignment. Example - when you push codes of this assignment, they should be inside Assignment2 folder.
- Please keep in mind that you don't commit all the codes together. Keep on committing codes module wise or question wise whatever seems available.

1. Write a program that prints a box, an oval, an arrow and a diamond as follows. Please keep in mind that you need to take number of rows and columns input from user.



(a). (b). (c). (d).

2. Create an **Account** class that a bank might use to represent customers' bank accounts. Include a data member of type int to represent the account balance. 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's greater than or equal to 0. If not, set the balance to 0 and display an error message indicating that the initial balance was invalid. 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 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**.

3. 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 con- structor 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 1–12; if it isn't, 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.

4. Create a class **Rectangle** with attributes **length** and **width**, each of which defaults to 1. Provide member functions that calculate the **perimeter** and the **area** of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.

5. Now create a more sophisticated Rectangle class than the one you created in problem 4. This class stores only the Cartesian coordinates of the four corners of the rectangle. The constructor calls a set function that accepts four sets of coordinates and verifies that each of these is in the first quadrant with no single x- or y-coordinate larger than 20.0. The set function also verifies that the supplied coordinates do, in fact, specify a rectangle. Provide member functions that calculate the **length**, **width**, **perimeter** and **area**. The length is the larger of the two dimensions. Include a predicate function square that determines whether the rectangle is a square.

6. Modify class **Rectangle** from problem 5 to include a draw function that displays the rectangle inside a 25-by-25 box enclosing the portion of the first quadrant in which the rectangle resides. Include a **setFillCharacter** function to specify the character out of which the body of the rectangle will be drawn. Include a **setPerimeterCharacter** function to specify the character that will be used to draw the border of the rectangle. If you feel ambitious, you might include functions to scale the size of the rectangle, rotate it, and move it around within the designated portion of the first quadrant.

Reference - C++ How To Program By Deitel and Deitel