CISC2000: Computer Science II
Polygon Class II

Objective: To design, develop and test a user defined class in the C++ programming language.

Implement the Polygon class and write a main driver program to allow for the instantiation of multiple polygon objects each of varying length and width. For full credit, the program should instantiate an array of pointers to objects of the Polygon class. The class data member MAX_POLYGONS should be used to declare the size of the array. The program should provide the user with the following options:

1. Build a new polygon object and assign it as the next entry in the array.

2. Draw the shape corresponding to any one of the polygons which have been built. The user can specify the desired polygon by using the objects numerical position in the array (i.e. 1 for the first polygon object, 3 for the third etc.)

3. Show a summary of information of all the polygons that have been built. The summary should include detail information about each polygon. This information should include the length, width, area and perimeter. The summary should also display class level statistics, including how many polygons were built as well as the average area and average perimeter of all the polygons. Note that these statistics should be maintained by the driver program.

   Write a local function whose purpose is to output a balance summary for each account created and **release the memory** for that account once it has been displayed.

   Note that this function should take as an argument the array of polygon objects. *The function should also release the memory occupied by the array of pointers.*

The definition of the Polygon class should be in an individual header (.h) file and the implementation of the class's methods should be in an individual source (.cpp) file. Write a main() function to test your code.

Challenge Add an option to remove from the array a previously built polygon. Take care as this is not as simple as it may appear. Deleting a polygon entails:

- deallocating the object from the heap.
- If the polygon being deleted is not the last polygon built, the elements of the array need to be shifted to eliminate the empty slot. In other words, if the array contains pointers to five polygon objects and the 2nd polygon object is removed, then the 3rd polygon object should shift up to the second slot, etc.

all class level statistics need to be adjusted to reflect the removal of one of the polygons.

Important

1. Follow an incremental approach in writing your program. In other words, write a few lines of code, make sure it compiles and executes as expected, then continue with more code.
2. If the program compiles but does not work as expected, consider using cout statements to display the value of variables in order to identify the errors.

For full credit be sure to:

- Include a descriptive Comment Block.
- Use correct indentation and allignment.
- Use descriptive identifiers for variable names as well as appropriate data types.
- Use blank lines to separate the code into appropriate blocks.
- Include comments to help others understand your program, and help yourself think!