

Programming Assignment #8

Learning objective : You are to gain implementing an exception handler. You are to modify your assignment #1 (Diamond Class) to use exception handlers to cover the situations where values are entered that are not within the parameters of the program specifications.

Implementation Details:

- The size of a box cannot not go below 1 either during the initialization phase under the constructor, during the setSize() function nor during the shrink() member function.
- The size of a box cannot exceed 39 either during the initialization phase under the constructor, during the setSize() function nor during the shrink() member function.
- The Border character cannot be assigned to any character that is not between the '!' or '~'. If any attempt is made, the default character of a "*" is assigned. This exception is handled in the constructor and setBorder member function.
- The Fill character cannot be assigned to any character that is not between the '!' or '~'. If any attempt is made, the default character of a "#" is assigned. The exception is handled in the constructor and setFill member function .

Testing Your Class:

You will need to test your class, which means you will need to write one or more main programs that will call upon the functionality (i.e. the public member functions) of the class and exercise all of the different cases for each function. You can use the previous main program used in the original assignment #1 as a starting place. You should test the situation where the constructors and member functions are all exercised.

Submitting:

Submit the following files on the Assignment Blackboard portal:

diamond.h
diamond.cpp

General Requirements:

- all member data of the Date class must be **private**.
- The **const** qualifier should be used on any member functions where it is appropriate
- You only need to do error-checking that is specified in the descriptions above. If something is not specified (e.g. user entering a letter where a number is expected), you may assume that part of the input will be appropriate.
- user input and/or screen output should only be done where described (i.e. do not add in extraneous input/output).
- no global variables, other than constants
- The program must be properly documented

Grading Criteria:

- The program compiles. If the program does not compile no further grading can be accomplished. Programs that do not compile will receive a zero.
- (25 Points) The program executes without exception and produces output. The grading of the output cannot be accomplished unless the program executes.
- (25 Points) The program produces the correct output.
- (25 Points) The program specifications are followed.
- (10 Points) The program is documented (commented) properly.
- (5 Points) Use constants when values are not to be changed
- (5 Points) Use proper indentation
- (5 Points) Use good naming standards