ECS769P Advanced Object - Oriented Programming

Lab 1: OO Basic - Objects and Classes (in-lab assessment in week 7)

Exercise 1: Exam Report (basic exercise)

Complete the ExamReport programs introduced in the lecture. Try to write the code yourself without looking at the notes. Successfully compile and run the programs.

Exercise 2: Yearly Rainfall Statistics (core exercise)

Design and Implement a program that represents yearly rainfall statistics. The main input values of this program are twelve numbers, which are the monthly rainfall amount (mm) for a single year. For example: the UK monthly rainfall amount (mm) in 2019 are: 63.6, 72.8, 132.9, 51.4, 64.8, 111.5, 88.9, 136.5, 122.4, 138.8, 117.6, 138.9.

This program should have the following public functions:

- getMonthAmount – takes a single argument (1 for January, 2 for February, etc) and returns the rainfall amount for that month. Return –1 for an invalid month.
- setMonthAmount – takes an argument as the month and a second argument representing the rainfall amount to be set for that particular month. The function should set the rainfall figure for the given month to the given value. Display an error message for an invalid month or amount.
- getHighest – show the highest rainfall amount and its month.
- getLowest – show the lowest rainfall amount and its month.
- getMean – returns the mean monthly amount for the whole year.
- outputBarChart - display a chart of your data using *. (similar to the example in Lecture 01).
  amount<60, display one *.
  60<=amount<80, display two *.
  80<=amount<100, display three *.
  100<=amount<130, display four *.
  amount >= 130, display five *.

Notes:

- Separate header file, source code file and main program.
- Consider class encapsulation and information hiding.
- Use appropriate data types.

Exercise 3: Number Guessing Game (challenging exercise)

Design and Implement a program that represents a number guessing game. The game generates a random integer in the range 1000 to 9999 and prompts the user to guess the value. Invalid guesses (ie. ones that are outside the range, none integer etc) should receive an error message reminding the user of the range. Valid guesses that are above or below the random value should receive a message that indicates whether they are too high or too low. When the number has been guessed, print the number of valid guesses that the user took. It then should ask the user whether to play again. User may input “quit” to quit the game at any time.

Notes:

- Separate header file, source code file and main program.
- Consider class encapsulation and information hiding.
- Check the C++ reference for generating a random integer.