GMIT

BEng(H) Software & Electronic Engineering

C++ Programming

Respiratory Rate Monitor

Lecturer: Michelle Lynch

Assignment

Design, code, and test C++ software to represent a **Respiratory Rate Monitor (RRMonitor)**.

Your code is required as part of a larger software engineering project that requires a Respiratory Rate Monitor.

Requirements

- The RRMonitor is required to store n samples of respiratory rate data; n can be any value but limit n to 10 for testing. The respiratory rate data has units of Breaths Per Minute (BPM), where a normal range for an adult is approximately 12 to 25 BPM. An example set of data samples is: data = [18, 20, 21, 19, 20, …].
- Each RRMonitor should also have a unique identifier or name.
- Develop your own code. You may re-use/adapt code you developed during the C++ Programming module if desired. Include reference to any online third-party code/examples referenced in the comment header at the top of your relevant source file(s).
- Use object-oriented programming for the RRMonitor design code. Use procedural programming for test code.
- Use modern C++.
- Provide a basic interface for the user of your design to enable them to perform basic common tasks, such as create new RRMonitor(s), read and/or write to an RRMonitor.
- Include a copy assignment operator in your design.
- Create & run unit tests for your code.
- Add functionality to your RRMonitor design to enable the RRMonitor's data to be multiplied by a scalar.
- Add functionality to your RRMonitor design to calculate the difference in BPM between adjacent samples.
- Add functionality to your RRMonitor to add a new data sample to the BPM data, while keeping the total number of samples stored, n, the same.
- Add any additional design and/or test feature(s) to your code, of your own choosing, that you think would be useful and demonstrate your C++ skills.
- Make any assumptions or coding decisions necessary to deliver what you consider to be your best quality design and test code.
- Create 'clean code': well designed, efficient, demonstrating software engineering principles with good coding style.