

# **Insurance Policy Valuation Model Application**

**James Clinton Gray**

**September 2014**

**Dissertation submitted in partial fulfilment for the degree of  
Master of Science in Computing for Financial Markets**

**Computing Science and Mathematics  
University of Stirling**

## **Abstract**

Insurance companies such as Prudential have a constant need to generate the ongoing and future values of insurance policies that they cover for all their customers. Unfortunately, the way they do this changes at a rapid pace as more factors need to be inputted or removed from their valuation model. This generates a lot of problems in how these companies must do these calculations, as over time the code required to generate the valuations becomes more convoluted and therefore harder to understand and harder to update. This in turn makes updating their valuations incredibly expensive and time consuming to update and reduces the company's efficiency.

This project has been implemented with an aim of addressing these major problems in Insurance Policy Valuation. The general aim of this project being to design and create a basic application, in an object orientated and modular manor, which will serve as a starting basis for future valuations. The application is designed to be as easy to understand and extend as possible to allow for the large amount of changes that these valuations often undergo. With this in mind, the application should be able to solve the problems inherent within the current methods used to generate these valuations.

In order to make the application as easy to understand and modify as possible, a specific approach was required to the design and implementation of the program. First, it had to be able to do the calculations that were needed in order to actually value any given policy, and to allow these calculations to be modified and extended over time. This requires the use of modular development within the program for these calculations, so that it is easy for any programmer to see where the calculations start and what exactly they do. This also necessitated the use of an object-orientated approach, which allowed the complete separation of the User-Interface and the programming logic, making it much simpler to understand by reducing as much of the connections as possible.

This implementation approach allowed the fulfilment of the objectives of the application, in that it was easy to modify for current operations and to extend for future operations. As well as allowing the logic to be easy to understand due to its modular and object-orientated nature. While a basic application, it allows for easier future development in the area of Insurance Policy valuation.