**Software Quality and Testing CA 1**

* Implement testable code to determine the premium

public double CalcPremium(int age, string location)

{

double premium;

if (location == "urban")

{

if ((age >= 18) && (age <= 30))

{ premium = 5.0; }

else

if (age >= 31)

{ premium = 2.50; }

else

{ premium = 0.0; }

}

else

if (location == "rural")

{

if ((age >= 18) && (age <= 35))

{ premium = 60; }

else

if (age >= 36)

{ premium = 50; }

else

{ premium = 0.0; }

}

else

{ premium = 0.0; }

if (age >= 50)

{ premium = premium \* 0.15; }

return premium;

}

* Draw a flow graph for code

A close up of a map

Description automatically generated

* What is the cyclomatic complexity of the program, describe how this is calculated from the graph

Cyclomatic complexity tells how complex the code is. The higher the number is, the more complex the code will be. Its common practice to set cut off points for programs. 20 points is usually the cut off point. Separating code in methods will help lower the cyclomatic complexity.

E = Edges, N = Nodes

Cyclomatic Complexity = E – N + 2

8 = 23 – 17 + 2

* Devise a set of test cases to test achieve 100% branch coverage – Document these in a spreadsheet

A close up of text on a white background

Description automatically generated

* Using **NUni**t, automate your white box tests