# Bees - object-oriented development exercise

This exercise will not be the only factor in our recruitment process but provides an insight into your approach to development work. Please be prepared to talk about your design, testing, and decision making in your interview.

**You have 60 minutes**

Produce a compiled result in C#. Start with part 1 – the classes. If time permits move on to part 2 – the application. The focus is not on the interface but rather on the classes.

Once the time has ended please zip up your code/solution and provide a link to it through an online storage repository such as OneDrive, Google Drive or GitHub.

## Part 1 – the classes

Write classes to represent 3 different types of Bee - Worker, Queen and Drone.

Each Bee has a floating-point health property, which is not writable externally and upon creation is set to a value of 100.

Each Bee has a Damage() method that takes a single integer parameter that should be a value between 0 and 100.  When this method is called, the health of the bee is to be reduced by that percentage of their current health.

When a Worker has a health below 70% it cannot fly and therefore is pronounced Dead.  When a Queen has a health below 20%, or a Drone below 50%, it is pronounced dead.  This 'Dead' property should be readable from each Bee.  When a Bee is dead, no further health deductions should be recorded by the bee, although the Damage() method should still be invokable without error.

## Part 2 – the application

Your application should create a single list containing 10 instances of each type of Bee.  It must support methods to allow Damage() to be called for each Bee, and to return the health status of each bee, including whether it is alive or not.

The user interface should show the current status of each bee and contain a button.  When clicked, a different random value between 0 and 80 should be selected for each bee and applied with a call to Damage().  After each click the health status of the bees should be updated.