```matlab
function [newGen] = GreenModGeneticAlgorithmLL(prevGen)
% GENETIC ALGORITHM TO PROCESS A GIVEN SET OF CHROMOSOMES DEFINING A
% POPULATION AND USE FITNESS SCORES TO PRODUCE A NEW GENERATION WITH
% CROSSOVER AND MUTATION ALTERATIONS.
%
%   NOTE: THIS IS USED IN THE LOWER LEVEL OPTIMIZATION!!!
%         USE GreenModGeneticAlgorithmUL.m FOR THE UPPER LEVEL CHROMOSOME
%         OPTIMIZATION
%         THE ONLY DIFFERENCE BETWEEN THIS SCRIPT AND
%         GreenModGeneticAlgorithmUL.m IS THAT IT DOES NOT TAKE INTO
%         ACCOUNT FITNESS SCORES FOR THE CROSSOVER. CROSSOVER SELECTION IS
%         THEREFORE DONE RANDOMLY
%
%   INPUTS:
%       [PxN DOUBLE] ARRAY OF DESIGN VECTORS DEFINING A POPULATION FOR A
%                    ITERATIVE GENERATION
%       [Px1 DOUBLE] SCORE OF THE INDIVIDUALS (CHROMOSOMES THAT ARE THE
%                    ROWS OF THE ARRAY) WHEN TESTED AGAINST A FITNESS
%                    FUNCITON
%   OUTPUTS:
%       [PxN DOUBLE] ARRAY OF ALTERED POPULATION THAT DEFINES THE NEXT
%                    GENERATION IN THE OPTIMIZATION
%
%
% ENGINEERS: JAMES S COLLINS
%            BEN DUSSALT
%            NAMKHA NORSANG
%            NISHANTH KATHIRVEL
%
% PROJECT: ME 6101 GREEN MODULAR DESIGN GROUP PROJECT
% DATE: NOVEMBER 2017
% LOCATION: GEORGIA INSTITUTE OF TECHNOLOGY. ATL, GA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% IDENTIFY THE NUMBER OF INDIVIDUALS IN THE POPULATION
nParents = size(prevGen,1);
% RANDOMLY DETERMINE THE SELECTION INDEX
mateIndex = randi(nParents,[nParents,1]);
% CALL THE CROSSOVER FUNCTION TO MATE DIFFERENT INDIVIDUALS IN ROWS OF THE
% POPULATION ARRAY
newGen = GreenModCrossoverScattered(prevGen,mateIndex);
% INTRODUCE MUTATIONS
newGen = GreenModMutation(newGen);
end
```

Published with MATLAB® R2017b