| **(a)** | simulateOneDay |  |  |
| --- | --- | --- | --- |
|  | **Scoring Criteria** | **Decision Rules** |  |
| **1** | Generates a random value | Responses **can** still earn the point even if they  • fail to save or use the generated random value  Responses **will not** earn the point if they  • fail to call Math.random, or possibly the equivalent, at least one time  • make any incorrect call to  Math.random | **1 point** |
| **2** | Identifies two cases based on a comparison of a randomly generated value and some constant that implements a 5% probability | Responses **can** still earn the point even if they  • reverse the 5/95 probability  • compare double values using <= or  >= instead of < or >  • incorrectly cast the 5/95 random value, as long as a suitable range is generated and the comparison divides that range appropriately  • call Math.random incorrectly | **1 point** |
| **3** | Generates a random integer that is uniform in the range [10, 50] | Responses **can** still earn the point even if they  • call Math.random incorrectly | **1 point** |
| **4** | Scales for numBirds and subtracts appropriate amount from currentFood in all cases (*algorithm*) | Responses **can** still earn the point even if they  • compare double values using <= or  >= instead of < or >  Responses **will not** earn the point if they  • reverse the 5/95 probability  • incorrectly process the 5/95 range/comparison (e.g., by incorrect casting)  • exit the method while currentFood  has a negative value | **1 point** |
|  |  | **Total for part (a)** | **4 points** |

| **(b)** | simulateManyDays |  |  |
| --- | --- | --- | --- |
|  | **Scoring Criteria** | **Decision Rules** |  |
| **5** | Calls simulateOneDay with value of  numBirds | Responses **will not** earn the point if they  • fail to make at least one correct call to  simulateOneDay  • make any call to simulateOneDay on the class or on an object other than this (use of this is optional) | **1 point** |
| **6** | Loops over the simulation method call and guards that it runs at most numDays times | Responses **can** still earn the point even if they  • call the method that simulates one day incorrectly  • count the simulated days incorrectly, as long as the loop guard would work if the count were corrected  • fail to guard against calls to the simulation method when currentFood <= 0  Responses **will not** earn the point if they  • fail to count the simulated days at all | **1 point** |
| **7** | Counts the number of times that the method that simulates one day is called with food available in the feeder (*algorithm*) | Responses **can** still earn the point even if they  • simulate extra days when currentFood is 0, as long as the count is still correct  Responses **will not** earn the point if they  • fail to initialize the counter appropriately  • compute the correct count but return something else  • fail to test if food is available | **1 point** |
| **8** | Compares currentFood and 0 | Responses **can** still earn the point even if they  • fail to make the comparison in the context of a loop  • use the result of the comparison incorrectly | **1 point** |
| **9** | Returns any int value, in all cases | Responses **can** still earn the point even if they  • return a constant | **1 point** |
|  |  | **Total for part (b)** | **5 points** |
|  |  | **Total for question 1** | **9 points** |