# CS0455 – ALGORITHMS AND INFORMATION STRUCTURES

Assignment – Food Skewers

Suppose you have a food skewer with a handle and one pointed end. You can only add and remove food items at the pointed end – you cannot add/remove from the middle. Suppose you have already created a class called **FoodItem** to represent a food item chunk that can be added to a skewer and you want to write a class called **FoodSkewer** to represent the skewer described above. Part of the UML is shown:

| **FoodSkewer** |
|---|
| - skewer: FoodItem[]<br>- numberOfItems: int<br>- DEFAULT_CAPACITY = 10: int<br>- MAX_CAPACITY = 10000: int |
| + FoodSkewer()<br>+ FoodSkewer(initialCapacity: int)<br>+ getCurrentSize(): int<br>+ isEmpty(): boolean<br>+ add(newFoodItem: FoodItem): void  // calls **doubleCapacity** if necessary<br>+ getLastFoodItem(): FoodItem  // does not change the skewer<br>+ removeLastFoodItem(): FoodItem<br>- isArrayFull(): boolean<br>- doubleCapacity(): void |

Describe what each of the methods does by referring to the four fields/instance variables in the class. Make sure your answer is clear and also describes the return.

+ FoodSkewer()

+ FoodSkewer(initialCapacity: int)

+ getCurrentSize(): int

+ isEmpty(): boolean

+ add(newFoodItem: FoodItem): void

+ getLastFoodItem(): FoodItem

+ removeLastFoodItem(): FoodItem

- isArrayFull(): boolean

- doubleCapacity(): void