

CISC 3115 TY3
C20c: Exercises

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Exercises

- C20c-1: revise the sample code `SortingArrays` to sort an `ArrayList` instead of an array
- C20c-2: revise the sample code `SortShapeCollections`, and sort a list of `Circles` according their areas
- C20c-3: revise the sample code `SortShapeCollections`, and sort a list of `Shapes` according their areas
- (Optional) C20c-4: revise the sample code `AnimalKingdomEnhanced`, and to find the heaviest animal from a collection and an array of animals without using the sort method
- (Optional) C20c-5: revise the sample code `AnimalKingdomEnhanced`, and to sort animals in descending order according their weight.
- See next set of slides for more details

C20c-1: Sorting ArrayList

- Your task is to revise the sample code `SortingArrays` to sort an `ArrayList` instead of an array
 - Create a directory called `C20c-1`, and copy the [SortingArrays](#) sample code to the directory
 - Revise the code to store the objects in `ArrayList`'s instead of arrays and sort the `ArrayList`'s
 - Use git to make a submission

C20c-2: Sorting Circles

- Your task is to revise the sample code `SortShapeCollections`, and sort a list of `Circles` according their areas
 - Create a directory called `C20c-2`, and copy the [SortShapeCollections](#) sample code to the directory
 - Create and write a `Circle` class
 - Revise the code to sort a few circles in an `ArrayList`
 - Use `git` to make a submission

C20c-3: Sorting Shapes

- Your task is to revise the sample code `SortShapeCollections`, and sort a list of `Circles` according their areas
 - Create a directory called `C20c-3`, and copy the [SortShapeCollections](#) sample code to the directory
 - Create and write a `Circle` class
 - Revise the code to sort a few shapes including a few circles and a few rectangles in an `ArrayList` (Hint: what data type should the `ArrayList` store?)
 - Use git to make a submission

(Optional) C20c-4: Heaviest Animal

- Your task is to revise the sample code `AnimalKingdomEnhanced`, and to find the heaviest animal from a collection and an array of animals without using the `sort` method
 - Create a directory called `C20c-4`, and copy the [AnimalKingdomEnhanced](#) sample code to the directory
 - Write a generic method called `findTheHeaviest` that takes an `ArrayList` of animals, and return the heaviest animal
 - Revise the `AnimalApp` class, and create an `ArrayList` of animals, and displays the heaviest animal.
 - Use `git` to make a submission

(Optional) C20c-5: Sorting Animals in Descending Order

- Your task is to revise the sample code `AnimalKingdomEnhanced`, and to sort animals in descending order according their weight.
 - Create a directory called `C20c-5`, and copy the [AnimalKingdomEnhanced](#) sample code to the directory
 - Write a `AnimalComparator` class that implements the `Comparator` interface to be used to sort the animals according their weights in the descending order
 - Revise the `AnimalApp` class, and create an `ArrayList` of animals, sort the animals, and displays the sorted animals.
 - Use git to make a submission