

CISC 3115 TY2

# Exercises for Java Interfaces

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

# Exercises

- EX1. implementing the improved Animal class hierarchy with interfaces
- Ex2: revise the SortingArrays in the lecture to sort an ArrayList instead of an array
- Ex3. revise the SortShapeCollections in the lecture, and sort a list of Circles according their areas
- Ex4. revise the SortShapeCollections, and sort a list of Shapes according their areas
- Ex5. revise the sample code AnimalKingdomEnhanced, and to find the heaviest animal from a collection and an array of animals without using the sort method
- Ex6: revise the sample code AnimalKingdomEnhanced, and to sort animals in descending order according their weight in both descending and ascending orders
- See next set of slides for more details

# Required and Optional Exercises

- Ex1 and Ex3 are required
- The rest are optional.

# Ex1. Animal Class Hierarchy with Interfaces

- Your task is to implement the improved Animal class hierarchy with interfaces
  - Create a directory in your journal
  - Based on the lecture nodes, implement the class hierarchy and interfaces
    - Including FlyingCat class, implementing the generic methods discussed
  - Use git to make a submission

# Ex2: Sorting ArrayList

- Your task is to revise `SortingArrays` to sort an `ArrayList` instead of an array
  - Create a directory in your journal
  - Download the `SortingArrays` on Blackboard 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

# Ex3: Sorting Circles

- Your task is to revise SortShapeCollections, and sort a list of Circles according their areas
  - Create a directory in your journal
  - Download the SortShapeCollections on Blackboard 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

# Ex4: Sorting Shapes

- Your task is to revise SortShapeCollections, and sort a list of Circles according their areas
  - Create a directory in your journal
  - Download the SortShapeCollections on Blackboard 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

# Ex5. Heaviest Animal

- Your task is to revise the code you written for ex1, and to find the heaviest animal from a collection and an array of animals without using the sort method
  - Create a directory in your journal
  - Copy your solution in ex1 to the directory
  - Write a generic method called findTheHeavist 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



# Ex6. Sorting Animals in Descending Order

- Your task is to revise the code you written for ex1, and to sort animals in descending order according their weight.
  - Create a directory in your journal
  - Copy your solution in ex1 to the directory
  - Write a AnimalComparatorDescend class that implements the Comparator interface to be used to sort the animals according their weights in the descending order
  - Write a AnimalComparatorAscend class that implements the Comparator interface to be used to sort the animals according their weights in the ascending order
  - Revise the AnimalApp class, and create an ArrayList of animals, sort the animals, and displays the sorted animals in both descending and ascending orders
  - Use git to make a submission