



Tutorial # 8 Polymorphism and Polymorphic Sorting

SOFE 2710U Object Oriented Programming and Design

Objective

In this tutorial, you will learn more about polymorphism and polymorphic sorting.

Part 1: Polymorphism

In part one of this tutorial we will talk about polymorphism, Polymorphism is a concept in which we can perform a single task in different ways, it can be done by method overloading and overriding. The main use of polymorphism occurs when a parent class reference is used to refer to a child class object. It is important to know that the only possible way to access an object is through a reference variable. A reference variable can be of only one type. Once declared, the type of a reference variable cannot be changed.

The reference variable can be reassigned to other objects provided that it is not declared final. The type of the reference variable would determine the methods that it can invoke on the object.

A reference variable can refer to any object of its declared type or any subtype of its declared type. A reference variable can be declared as a class or interface type.

Problems

1. Draw and annotate a class hierarchy that represents various types of animals in a zoo. Show what characteristics would be represented in the various classes of the hierarchy. Explain how polymorphism could play a role in guiding the feeding of the animals.

Part 2: Polymorphic Sorting

In part two, we will look into sorting, sorting is a process of arranging a list of items in a particular order, we will be analyzing selection and insertion sorting.

For selection sort, we will select a value and place it at its final position in the list and we will repeat the same for all the other values.

For insertion sort, we will pick any element and insert it into its final place in an already sorted sublist and we will repeat the same for all the items.

Problems

1. The file Numbers.java reads in an array of integers, invokes the selection sort algorithm to sort them and print the sorted array.
2. Write a program Strings.java, similar to Numbers.java, that reads in an array of String objects and sorts them.
3. Modify the insertion Sort algorithm so that it sorts in descending order rather than ascending order. Change Numbers.java and Strings.java to call insertion Sort rather than selection Sort. Run both to make sure the sorting is correct.