# *Programming III (420-B31-HR)*

# *Lab 2 – Generic Types*

Date assigned & due: Tuesday, August 30, 2016

**Objectives:**

1. Review:
   1. Object-oriented terminology
   2. Creating and using arrays
   3. Overloading the **toString()** and **equals()** methods
   4. Multiple inheritance using interfaces
2. Learn to:
   1. use generic types.

**To be handed in:**

1. Your ***username*\_B31\_L02\_Generic\_Types** folder should be zipped and uploaded to Moodle.

# Review

**Objectives**: Review inheritance, interfaces and method overriding.

Review the **toString()** and **equals()** methods.

**To Start:**

1. Copy the **B31\_L02\_Generic\_Types** folder from the Moodle page to your **420-B31\Labs** folder. Rename it to ***username*\_B31\_L02\_Generic\_Types**.
2. Start **Eclipse** and use your **420-B31\Labs** folder as your workspace.
3. Create a new Java Project called ***username*\_B31\_L02\_Generic\_Types.**

**To Do:**

## Refer to the following class diagram for Shapes. The **Cylinder** class does not exist in the code, and you will be adding it in this exercise.



## Add a new class called **Cylinder**. It should be a subclass of **Circle** and should implement the **ThreeD** interface. (Similar to the way **RectangularPrism** is a subclass of **Rectangle** and implements **ThreeD**).

## \*Sanity Saving Note: At the end of this section of the lab (step e), your code in the Cylinder class should have three constructors, toString(), equals(), getDepth(), setDepth(), getSurfaceArea(), getPerimeter() and getVolume() methods in it.

## Add three constructors with the following signatures. Refer to the **RectangularPrism** constructors for how to construct them.

**public Cylinder()**

**public Cylinder(double theRadius, double theDepth)**

**public Cylinder(Circle theCircle, double theDepth)**

## Right-click and select **Source🡪 Override/Implement Methods**. Select **getSurfaceArea()** and **getPerimeter()**.

## Code **all** the **Cylinder** methods. Refer to the **RectangularPrism** class for examples. The formulae for the calculations are:

**surface area = (2 \* CircleArea) + (2 \* PI \* radius \* depth)**

**volume = CircleArea \* depth**

**perimeter = CirclePerimeter**

## Remove the comments from **TestCylinder** and run it to test your changes. The output should be similar to:

**cylinder1:**

**Dimensions: Cylinder: radius = 1.0 depth = 1.0**

**Perimeter: 5.141592653589793**

**Surface Area: 12.566370614359172**

**Volume: 3.1415926535897932384**

**cylinder2:**

**Dimensions: Cylinder: radius = 5.5 depth = 25.0**

**Perimeter: 14.141592653589793**

**Surface Area: 1054.0043352793755**

**Volume: 2375.8294442772813**

**cylinder3:**

**Dimensions: Cylinder: radius = 5.5 depth = 25.0**

**Perimeter: 14.141592653589793**

**Surface Area: 1054.0043352793755**

**Volume: 2375.8294442772813**

**cylinder1 is not equal cylinder2**

**cylinder2 equals cylinder3**

# Java Generic Types

**Objectives**: Review how to create and use arrays.

Learn to use generic types.

**To Do:**

## Open the **Pair** class.

Look at the class header. Notice the **<T>**. What’s the data type for the **firstElement** instance variable? The **<T>** symbol indicates that the class is going to use a generic type that will be resolved when the class is instantiated. It is then used in the class everywhere that a generic class is desired.

## Open **GenericsTypeCheckingEx**. Uncomment the code. There are a number of type errors in this class. Refer to the notes on generic types and correct the errors. Run the program.

## Add code to **GenericsTypeCheckingEx** to:

### declare a variable **twoRectangles** that stores a pair of **Rectangles**. Each rectangle should be a different size.

### instantiate **twoRectangles** to store a pair of **Rectangles**.

### display **twoRectangles** using the **Pair toString()** method.

## Run **GenericsTypeCheckingEx** to see that your changes worked.

## Open the **NoahsArk** class.

## Pairs of animals have been defined. Now we want to create an equivalent array of pairs to store the animals in the ark.

## Create a **Pair** of **String** objects called **sexPair**. The first element should contain “Female” and the second element should contain “Male”.

## Create an array of 10 **Animal** **Pair** objects called **animalPair**. Use a while loop to populate the array with the objects from the **firstAnimal** and **secondAnimal** arrays. Loop while there is an instantiated **firstAnimal**. Store the number of pairs in the **numPairs** variable.

## Display the number of pairs of animals in the ark. Run the program to ensure that it is working correctly so far.

## Use the **toString** method to display all of the animals in the **animalPair** array. Run **NoahsArk**.

## Loop through the **animalPair** array and if the sex of the **firstElement** is not equal to the **firstElement** of the **sexPair** pair, swap the two elements of the **animalPair** array.

## Display all of the animals in the **animalPair** array again. Run the program to verify that your change worked.

## Add a method to **Pair<T>** that implements the following description.

/\*\*

\* Determine if either element of this <tt>Pair</tt> is equal to the argument.

\* @param e the object to compare to the elements of this <tt>Pair</tt>

\* @return the first element of this <tt>Pair</tt> found equal to <tt>e</tt>,

\* <tt>null</tt> if neither element is equal to <tt>e</tt>

\*/

public T elementEqualTo(T e)

## Use the **elementEqualTo()** method and loop through the **animalPair** array looking for a dmale lion. Display the index of the pair containing the male lion.

## Call the **speak()** method of each element of the **animalPairs** array to show the level of sound in the ark. Your final output should look similar to:

**There are 3 pairs of animals in the ark.**

**Pair[0] is < Female Dog called Unknown, Male Dog called Jasper >**

**Pair[1] is < Male Cat called Tripod, Female Cat called Tess >**

**Pair[2] is < Female Lion, Male Lion >**

**After swapping:**

**Pair[0] is < Female Dog called Unknown, Male Dog called Jasper >**

**Pair[1] is < Female Cat called Tess, Male Cat called Tripod >**

**Pair[2] is < Female Lion, Male Lion >**

**Index of pair with male lion is 2**

**The ark sounds like:**

**Woof Woof Meow Meow Roar Roar**

# Moodle Quiz

1. Log on to **Moodle**, go to the **Programming III** course page and complete the **Lab 2 Terminology Quiz by Friday September 2**.

**Marking Scheme**

|  |  |  |
| --- | --- | --- |
|  | **Mark** | **Out of** |
| Cylinder Class |  | 21 |
| GenericsTypeCheckingEx Class |  | 10 |
| NoahsArk Class |  | 44 |
| Pair Class |  | 7 |
| **Total** |  | **82** |