# Department of Computing

# CS 212: Object Oriented Programming

# Class: BSCS-8AB

# Lab 07: Polymorphism

# Date: March 25th, 2019

# Instructor: Ms. Hirra Anwar

**Learning Objectives**

The learning objective of this lab is to understand and practice the concept of polymorphism, a very powerful feature of OOP which helps in code extensibility.

**Warm-up.**

Consider there is a hair dresser, a surgeon and an actor. If you ask each of them to 'cut': the hair dresser starts cutting hair, the surgeon makes an incision, and the actor stops acting out the scene. This is polymorphism; same name but different behavior.

Now you should also think of a real life example of polymorphism.

**Task #1.**

Write an abstract class LivingThing.java followed by two concrete classes, Human.java and Monkey.java, extending the abstract class.

C:\Users\Its\Downloads\t1 (18).png

Once done defining classes use the following client class to test it.

package myabstractclassproject**;**

public class Main **{**

public static void main**(** String**[]** args**)** **{**

// Create Human object instance

// and assign it to Human type.

Human human1 **=** **new** Human**(** "Will Rodman"**);**

human1**.**walk**();**

// Create Human object instance

// and assign it to LivingThing type.

LivingThing livingthing1 **=** human1**;**

livingthing1**.**walk**();**

// Create a Monkey object instance

// and assign it to LivingThing type.

LivingThing livingthing2 **=** **new** Monkey**(** "Caesar"**);**

livingthing2**.**walk**();**

// Display data from human1 and livingthing1.

// Observe that they refer to the same object instance.

System**.**out**.**println**(** "human1.getName() = " **+** human1**.**getName**());**

System**.**out**.**println**(** "livingthing1.getName() = " **+** livingthing1**.**getName**());**

// Check of object instance that is referred by x and

// y is the same object instance.

boolean b1 **=** **(** human1 **==** livingthing1**);**

System**.**out**.**println**(** "Do human1 and livingthing1 point to the same object instance? " **+** b1**);**

**}**

**}**

Running the test should result in the following output.

Human Will Rodman walks...

Human Will Rodman walks...

Monkey Caesar also walks...

human1.getName() = Will Rodman

livingthing1.getName() = Will Rodman

Do human1 and livingthing1 point to the same object instance? true

**Bonus.** What happens when you create a LivingThing object in the Main class? For example using the statement,

LivingThing z = new LivingThing();

**Task #2.**

Your task is to write MyOnlineShop program by referring to the UML class diagram below.

**C:\Users\Its\Downloads\t1 (19).png**

Once done with the class definitions use the following tester class to confirm its working.

|  |
| --- |
| package myonlineshop**;**  public class Main **{**    public static void main**(**String**[]** args**)** **{**    // Declare and create Product array of size 5  Product**[]** pa **=** **new** Product**[**5**];**    // Create object instances and assign them to  // the type of Product.  pa**[**0**]** **=** **new** TV**(** 1000**,** "Samsung"**,** 30**);**  pa**[**1**]** **=** **new** TV**(** 2000**,** "Sony"**,** 50**);**  pa**[**2**]** **=** **new** MP3Player**(** 250**,** "Apple"**,** "blue"**);**  pa**[**3**]** **=** **new** Book**(** 34**,** "Sun press"**,** 1992**);**  pa**[**4**]** **=** **new** Book**(** 15**,** "Korea press"**,** 1986**);**    // Compute total regular price and total  // sale price.  double totalRegularPrice **=** 0**;**  double totalSalePrice **=** 0**;**    **for** **(**int i**=**0**;** i**<**pa**.**length**;** i**++){**    // Call a method of the super class to get  // the regular price.  totalRegularPrice **+=** pa**[**i**].**getRegularPrice**();**    // Since the sale price is computed differently  // depending on the product type, overriding (implementation)  // method of the object instance of the sub-class  // gets invoked. This is runtime polymorphic  // behavior.  totalSalePrice **+=** pa**[**i**].**computeSalePrice**();**    System**.**out**.**println**(**"Item number " **+** i **+**  ": Type = " **+** pa**[**i**].**getClass**().**getName**()** **+**  ", Regular price = " **+** pa**[**i**].**getRegularPrice**()** **+**  ", Sale price = " **+** pa**[**i**].**computeSalePrice**());**  **}**    System**.**out**.**println**(**"totalRegularPrice = " **+** totalRegularPrice**);**  System**.**out**.**println**(**"totalSalePrice = " **+** totalSalePrice**);**  **}**  **}** |

The test should result in the following output.

Item number 0: Type = myonlineshop.TV, Regular price = 1000.0, Sale price = 800.0

Item number 1: Type = myonlineshop.TV, Regular price = 2000.0, Sale price = 1600.0

Item number 2: Type = myonlineshop.MP3Player, Regular price = 250.0, Sale price = 225.0

Item number 3: Type = myonlineshop.Book, Regular price = 34.0, Sale price = 17.0

Item number 4: Type = myonlineshop.Book, Regular price = 15.0, Sale price = 7.5

totalRegularPrice = 3299.0

totalSalePrice = 2649.5

Also change the references and check which reference assignment is not allowed and how downcasting can work around.

**Hand in**

Compile a single word file and upload on LMS.