CO2220 Laboratory 5
Graphical Object-Oriented and Internet Programming in Java

Corresponds to Vol 1, Chapter 8 (Inheritance) and 9 (Abstraction) of Study Guide

Learning Objectives:

To understand and apply the following concepts:
- How abstract casss work
- How the keyword "implements" is used

Task 1

We are on the Seabird Colony Island and we saw the characters Dale and Yale ahead. In this task we should write a `Character` superclass for these characters and a `greet` for them to interact.

a) Write a class named **`Character`**. Write a **`greet`**() method with a method signature shown below. The method should display the message "`I am a generic Character.`" concatenated with `str` as a message.

```
protected void greet (String str)
```

b) Write a class **`Dale`** as a subclass of `Character`. Include a **`greet`**() method with a method signature shown below. The method should display the message "`I am a Dale.`" concatenated with `str` as a message.

```
public void greet (String str)
```

c) Write a class **`Yale`** as a subclass of `Character`. Include the **`greet`**() method such that the message "`I am Yale. Good to see you!`" concatatenated with `str` will be displayed as a message.

d) Write a class `SeabirdColony` with `public static void main()`. Declare a variable `var` with the type `Character`. Using this variable, demonstrate polymorphism by creating objects with classes `Dale` and `Yale`.

e) Use the objects created in previous parts to display messages with their **`greet`**() method. Is there any problems raised by the compiler?

Declare class `Character` as an abstract class. What other changes should you make to your current program so that it can compile successfully? Discuss how the correct behaviour will be presented depending on the actual `Character` object.

HMJ Tay, SH Tian

Hint:

```
public abstract class Character
{
     abstract void greet (String str);
}
```

f) Write another **greet**() method for the class **Yale**. Let this method accept an argument of the type Object instead of String. Let this method show the message, "I am a Yale." concatenated with str.

Put this method before the greet method that you wrote in part c. Class Yale has two **greet**() methods. The first takes in an Object as argument. The second takes in a String as argument.

Can both **greet**() methods be selected when you execute the statement below?

```
aYale.greet("Greetings!");
```

// where aYale is a Yale object.

Run the program to find out which of the two **greet**() methods run. Discuss what you have learnt from this.

Hint:
Consider casting the string into an object, e.g., aYale.greet((Object) "My greetings to you!");

g) Write an interface GreetInterface with a **greet**() method with a single string argument. Let it display a simple message "Hello Hello." Does the compiler allow you to do this?

Ans: No, all interface method cannot have body. Make changes so that it is permissible by the compiler.

Hint: Remove the method body.

```
public interface GreetInterface
{
     public abstract void greet (String str);
}
```

h) Modify the class `Yale` such that it implements `GreetInterface`. What modifications are needed for this to be successful?

Tip:

```
public class Yale extends Character implements GreetInterface
{
     public void greet (Object str)
     {
          System.out.println ("Object … I am a Yale");
     }

     public void greet (String str)
     {
          System.out.println ("I am a Yale.");
     }
}
```

Task 2

Learning Objectives:

To understand the use of enhanced `for` loop.

Write a program that will do the following:

a) Declare and create an array of the type String with a size of 50.

b) Assign, the first few elements of the array, values such as "GERMANY", "RUSSIA", "JAPAN", "KOREA", "UNITEDSTATESOFAMERICA" and "AUSTRIA".

c) Use the enhanced for loop to examine each string in the array

d) Stop the looping if the string "AUSTRIA" is encountered in the array

e) Display all the strings that are longer than 5 characters in length

f) Test your program with and without the string "AUSTRIA" in the array. What happens if "AUSTRIA" is not in the array? Can you explain why?

Task 3

Learning Objectives:

To have a better understanding of how abstract classes work and how the keyword "implements" is used through a study of the java examples provided in the course guide (package abstractclassesandinterfaces).

Read through the java files in this package. They include the following:

- App.java
- Drawable.java
- Message.java
- Movable.java
- MovingPolygon.java
- Polygon.java
- Shape.java

(a) Run App
What is your observation?

(b) Shape implements **Drawable**. Can we supply an implementation of draw method inside the Shape class?

(c) Where should the draw method be implemented?

(d) Modify the program to have a moving polygon with 3 vertices.

(e) Modify the program such that the moving polygon has color changing effects continuously.