

Homework Assignment 3

Late homework assignments will not be accepted, unless you have a valid written excuse (medical, etc.). You must do this assignment alone. No team work or "talking with your friends" will be accepted. No copying from the Internet. Cheating means zero.

Create a new Eclipse workspace named "**Assignment3_1234567890**" on the desktop of your computer (replace **1234567890** with your student ID number). For each question below, create a new project in that workspace. Call each project by its question number: "**Question1**", "**Question2**", etc. Answer all the questions below. At the end of the assignment, create a ZIP archive of the whole workspace folder. The resulting ZIP file must be called "**Assignment3_1234567890.zip**" (replace **1234567890** with your student ID number). Upload the ZIP file on iSpace.

Here are a few extra instructions:

- Do not forget to write tests for *all* the code of *all* the classes.
- Give meaningful names to your variables so we can easily know what each variable is used for in your program.
- Put comments in your code (in English!) to explain WHAT your code is doing and also to explain HOW your program is doing it.
- Make sure all your code is properly indented (formatted). Your code must be beautiful to read.

Failure to follow these instructions will result in you losing points.

Question 1

Create a class **Insect** with the following UML diagrams:

```
+-----+
|               Insect               |
+-----+
| - name: String                      |
+-----+
| + Insect(String name)              |
| + getName(): String                |
| + canFly(): boolean                |
| + testInsect(): void              |
+-----+
```

The **canFly** method of the **Insect** class returns a boolean indicating whether the insect can fly: some insects can fly and others cannot. By default, the **canFly** method just prints a message "**Cannot fly!**" and returns **false** (because the **canFly** method must return a boolean).

Add a class **Flee** to your program. A flee is an insect. The constructor for the **Flee** class takes no argument. All flees are named "**Fleet**" and flees cannot fly (the **canFly** method must not print any message though, it simply returns **false**).

Add a class **Dragonfly** to your program. A dragonfly is an insect. The **Dragonfly** class has a private instance variable **length** of type **double** that describes the length of the dragonfly and a private instance variable **age** of type **int**, with **getLength** and **getAge** methods. The constructor for the **Dragonfly** class takes the dragonfly's name, length, and age as arguments. A dragonfly can fly.

Add a class **EmperorDragonfly** to your program. Emperor dragonfly is a species of dragonfly. The **EmperorDragonfly** class has two constructors: the first constructor takes the Emperor dragonfly's name, and its length and age as arguments; the second constructor only takes the Emperor dragonfly's name as argument and always uses **2.0** as the Emperor dragonfly's length and **3** as its age. The second constructor must use the first constructor.

Add a class **MaleEmperorDragonfly** to your program. The constructor for the **MaleEmperorDragonfly** class takes no argument. Male emperor dragonfly is always named "**Beautiful**" and has a length of **3.5** and age of **4**.

Add a class **DragonflyNymph** to your program. Dragonfly nymph is a dragonfly in its baby form. The constructor for the **DragonflyNymph** class takes no argument. Dragonfly nymph is always named "**Baby Dragon**", has a length of **0.8**, age of **1**, and cannot fly.

Add a class **Start** to your program to test all your classes.

Question 2

Add a class **DisplayBox** to your program with the following UML diagram:

```
+-----+
|           DisplayBox           |
+-----+
| - dragonfly: Dragonfly         |
+-----+
| + DisplayBox(Dragonfly dragonfly) |
| + getDragonfly(): Dragonfly      |
| + testDisplayBox(): void         |
+-----+
```

In the **testDisplayBox** method, create male emperor dragonfly called **med1**, then create a display box with this male dragonfly in it. Then get the male emperor dragonfly from the display box and store it into a local variable called **med2** of type **MaleEmperorDragonfly**. Use the **==** operator to check that **med1** and **med2** are the same male emperor dragonfly.