

COMP1102/8702 – Practical Class 5

Class Definitions, Logical Operators, Selection and Iteration Statements

Aims and Objectives

This laboratory has been designed to help you to

- design and modify class definitions,
- extend a class with additional methods and instance variables and
- make use of selection and iteration statements.

Getting Started

Start *IntelliJ* and open the project named “Practical05” (download if from FLO).

Task 1

The following class definitions are included.

```
public class FrogProgram {
    public static void main(String[] args) {
        Frog myFrog = new Frog(); // create a Frog object
        myFrog.name = "Fred";      //###
        myFrog.print();
    } // end of main
} // end of FrogProgram

public class Frog {
    String name = "Name_not_set";
    int id = -1 ; // -1 used to indicates it has not been set

    Frog() { // Constructor currently does nothing
        // no code in body of constructor
    }

    void print() {
        System.out.println("**** Start of print method ****");
        System.out.println("Frog's name is " + name);
        System.out.println("Id is " + id);
    }
}
```

Compile and run the application. It should produce the output:

```
**** Start of print method ****
Frog's name is Fred
Id is -1
```

Add an assignment statement to the `main` method which sets the instance variable `id`, contained in the object referred to by `myFrog`, to 1000. **Hint:** observe how the instance variable name is being assigned a value in the line ending with `//###`.

Compile and run the application. It should produce the output:

```
**** Start of print method ****
```

```
Frog's name is Fred
Id is 1000
```

Checkpoint 22

Have the program source code and output marked by a demonstrator

Task 2

Modify the program developed in Task 1 in the following ways.

1. Declare another instance variable in the class `Frog` to store a Frog's age in days. Its initial value should be 0.
2. Add code to the `main` method to assign the instance variable `age`, in the object referred to in my `myFrog`, to 22.
3. Define a method called `printAgeGroup` which has no formal parameters and which prints out the age group of the frog according to the following table:

Age	Age Group
0 to 20 days	young
21 to 50 days	juvenile
more than 50 days	adult

4. Add a (call) to the `printAgeGroup` method (in the object referred to by `myFrog`) to the end of the `main` method.
5. Compile and run the application from *IntelliJ*. The following output should be produced:

```
**** Start of print method ****
Frog's name is Fred
Id is 1000
Age group is juvenile
```

Checkpoint 23

Have the program source code and output marked by a demonstrator

Task 3

Modify the program developed in Task 2 in the following ways.

1. Prefix each of the instance variable declarations (**name**, **id** and **age**) with the reserved word **private**. Compile the program, note the compilation errors and be prepared to determine what caused them.
2. Remove the offending assignment statements from the **main** method.
3. Modify the constructor in the class **Frog** so that it has three formal parameters, **theName**, **theId** and **theAge**, and assigns the corresponding instance variables, **name**, **id** and **age** to the formal parameters.
4. Add a declaration and statements to the end of the **main** method to create another **Frog** with the name **Fran**, and **id** of 1001 and an **age** of 75. The variable **myFrog2** should be used to store a reference to the new frog. Include calls to **print** and **printAgeGroup** for this frog.
5. Compile and run the application from *IntelliJ*. The following output should be produced:

```
**** Start of print method ****
    Frog's name is Fred
    Id is 1000
    Age group is juvenile
**** Start of print method ****
    Frog's name is Fran
    Id is 1001
    Age group is adult
```

Checkpoint 24

Have the program source code and output marked by a demonstrator

Task 4

Modify the program developed in Task 3 in the following ways.

1. Add a definition of a method called **reversedName** to the class **Frog** which prints the name using the following translation table:

Letter	Letter printed in its place
a	i
n	i
r	o
F	o

2. Make use of a **for** loop and a **switch** statement to control the printing of letters.
3. At the end of the **main** method add calls to the **reverseName** method for both frogs.
4. Compile and run the application from *IntelliJ*. The following output should be produced:

```
**** Start of print method ****
    Frog's name is Fred
    Id is 1000
    Age group is juvenile
**** Start of print method ****
```

```
Frog's name is Fran
Id is 1001
Age group is adult
oood
ooii
```

Checkpoint 25

Have the program source code and output marked by a demonstrator

Task 5 (Extension Practice)

Modify the program developed in Task 4 in the following ways.

1. Add a definition of a method called `greaterAge` to the class `Frog` which takes another `Frog` as a formal parameter and returns the `frog` (object reference) with the greatest age. For example,

```
Frog f = myFrog.greaterAge(myFrog2);
f.print();
```

should cause the following lines to be printed:

```
**** Start of print method ****
Frog's name is Fran
Id is 1001
```

2. Add the above statements to the end of the `main` method.