Lab 11: Building Classes

Objectives:

1. To practice building a new class:
   a. To define constructors
   b. To define methods

Preparation:

1. Go over the Lecture Notes Topic 12
2. Textbook reading (optional): Chapter 11, sections 11.1 to 11.6

In this lab we will look at the creation of a new class that represents information about employees in a university. We will define a class `Employee` which *models or represents* information about employees of a university (faculty, staff, researchers, etc.). The Employee class definition has been started below. Type it into the Definitions pane of DrJava and save it in a file *Employee.java*.

It starts with the field definitions: an Employee object has 6 attributes (also called instance variables) . **It is important to note that the attributes are declared outside the constructor and method definitions, so that the instance variable names can be used directly throughout the class definition (see Topic 13 slide 27).**

A main method has been included in the class, to use for testing your constructors and methods. (See the lecture notes Topic 13 Part 2 slides 2-3 on the use of a main method in a class definition for testing purposes.) Note that some of the main method code has been "commented out"; once you have added new methods to the class, you will have to remove the "//" from these lines.

```
/* Define a class for representing employees
 * of a University
 */
public class Employee {

  /* attributes (instance variables) */

  private String firstName;
  private String lastName;
  private String employeeNumber;
  private String jobTitle;
  private String department;
  private double salary;

  /* constructors */

      // PLACE YOUR CONSTRUCTORS HERE

  /* methods */

      // PLACE YOUR METHODS HERE
  /* for testing: main method for testing the class */

  public static void main (String[] args) {

    // create 2 sample Employee objects
    Employee emp1 = new Employee("Albert", "Einstein", "186000");
    Employee emp2 = new Employee("Grace", "Hopper", "19451945", "Faculty",
                                 "Computer Science", 86333.76);

    // show the employee information
    System.out.println(emp1.toString());
    System.out.println(emp2.toString());

    // try out the accessor and modifier methods
    // System.out.println(emp1.getFirstName() + " " + emp1.getLastName());
    // emp2.setSalary(emp2.getSalary() + 1000.00);
    // System.out.println(emp2.getSalary());

    System.out.println(emp1);
    System.out.println(emp2);

    /* add code to try out other method(s) here */



  }// end of main method

}// end of Employee class definition
```

Exercise 1: Defining the constructors for the class

In this exercise you will add two constructors to the Employee class in the indicated area of the class definition. Remember, constructors are just special methods that initialize attributes (fields) of the class. The attributes that are not initialized by the constructor will retain their default values. (Make sure you have reviewed Topic 13 of the Lecture Notes regarding constructors.)

a) The first constructor will have three parameters: the employee's first name, last name, and employee number. The constructor header will be
`public Employee(String empFirst, String empLast, String empNumber)`

b) For the second constructor you should decide what the header should be (i.e. what parameters it should have), by observing the usage of the second constructor in the main method above. (Hint: It's the constructor used in creating the Employee object referenced by the variable `emp2`.)

c) Compile the Employee class and run the main method (just use the Run button). What is printed by the `System.out.println` statements of the main method?

Exercise 2: Defining a `toString()` method for the class

It is conventional to have a `toString`() method defined in every class (see Topic 13 of the lecture notes.) Recall that if this method is not defined in a class, then Java uses one that is already defined elsewhere (in the Object class, which is the parent of all classes), as you saw in Exercise 1 step c). This default method is not very useful: it only showed the class name and an object identifier.

a) Add a `toString()` method to the Employee class that can be used to show the data about an `Employee` object. It should return a String containing all information about the employee.

b) Compile the Employee class with `toString()` added, and run its main method to see if your constructors and `toString()` worked correctly.
(Recall that a statement such as `System.out.println(emp1);` automatically invokes the `toString()` method of the Employee class.)

Exercise 3: Defining accessor and modifier methods

a) In most classes, you will need to define appropriate *accessor (getter)* and *modifier (setter, mutator)* methods. Define accessor methods for the attributes `firstName`, `lastName`, and `salary`, and a modifier method for the attribute `salary`. The names of these methods should be `getFirstName`, `getLastName`, `getSalary`, and `setSalary` respectively.

b) Now edit the main method so that the three statements using the accessor and modifier methods are no longer commented out. Compile the Employee class and run the main method to see if your accessor and modifier methods worked correctly. What is Grace Hopper's salary now?

Exercise 4: Defining another method

1. For programs using `Employee` objects, it may be useful to print out an employee's name in the format **lastname, firstname**
   Write a method called `lastNameFirst` and add it to the Employee class. This method will return a String that consists of the employee's name with the last name first, a comma, a space, and then the first name.

2. Now add statements to the main method to try out your `lastNameFirst` method. Compile the Employee class and run the main method to see if `lastNameFirst()` worked correctly.